Tesis Doctoral
La cadena media y su aplicación en Reconocimiento de Formas[1]

Presentada por D. Carlos David Martínez Hinarejos
Dirigida por el Dr. Francisco Casacuberta Nolla
y el Dr. Alfons Juan Ciscar

[1]Trabajo parcialmente enmarcado en los proyectos SisHiTra (TIC2000-1599-C02-01) y TAR (TIC2000-1703-C03-01)

Agradecimientos

Con este trabajo se culmina un periodo importante en mi carrera académica y profesional, periodo que no hubiera acabado de no ser por la colaboración de ciertas personas a los que pretendo enviar mi más humilde agradecimiento en estas líneas.

En primer lugar, por su directísima implicación en esta tesis, a mis directores Francisco Casacuberta y Alfons Juan, sin cuya intervención ni sin los dolores de cabeza que me han provocado hubiera conseguido este trabajo (poco podíamos sospechar que diera tanto de sí aquella primera reunión en otoño del 99). También a Andrés Marzal y Guillermo Peris, por su imprescindible colaboración en el tema de la distancia de edición normalizada para las cadenas cíclicas. A Ramón Mollineda por haberme facilitado mucho material para la parte de cadenas cíclicas y por haberme presentado el trabajo en el SSPR'2002. A José Molina, con cuya librería de generación de números aleatorios pude generar mi corpus artificial. Y también a Enrique Vidal, por sus ideas sobre métodos alternativos, y a Hermann Ney, por sus sugerencias sobre la utilización de modelos de Markov.

En el plano institucional, agradecer el apoyo de la Comisión Interministerial de Ciencia y Tecnología (CICYT), que a través de los proyectos SisHiTra (TIC2000-1599-C02-01) y TAR (TIC2000-1703-C03-01) ha apoyado en buena parte el trabajo realizado en esta tesis.

En un término también profesional, aunque ya más alejado de este trabajo, a todo el personal técnico y administrativo del ITI y del DSIC, así como a mis profesores de doctorado. A mis compañeros del PRHLT, en especial a toda la gente de la quinta planta del ITI (Fran, Jorge, Moisés, Alberto, Ismael, David) donde tantos buenos ratos pasamos, y a toda la gente de ambas instituciones que, de una manera u otra, me han ayudado a progresar. También a mis compañeros de docencia y a mis alumn@s y proyectand@s, que nunca me robaron más tiempo del debido.

Desde el lado personal, agradecer profundamente a mi familia el seguir ahí siempre. A l@s foráne@s, con los que tantos ratos buenos y menos buenos he compartido. A la gente que conocí en Spanish City, en especial a los valencianos (Paco, Nacho, Julio, David, Lucas, Moncho y Jorge) y al gran Min. A mis compañeros y profesores en el programa de doctorado del DISA, donde satisfice otra de mis pasiones. A mis amigos de toda la vida de Mislata (Jorge, JR, Toni y Miguel). A mis amig@s de Campillos Paravientos. A los buenos amigos y amigas que he hecho en mis viajes y congresos (especialmente a la gente que conocí en Odense, César, Enrique, Javi, Marta, Iraide y Elvira). A la gente de inglés (en especial a Adrián y Alicia). A Isabel, Nuria y Míriam, por haberme mantenido en contacto con el universo femenino en diversas etapas.

Espero no haber olvidado a nadie. Mi más sincera disculpa a quien se sienta omitid@.

Gracias a tod@s.

Resumen

En el campo del Reconocimiento de Formas, las técnicas de clasificación basadas en distancia (y más específicamente el clasificador k-NN) necesitan de la obtención de prototipos adecuados para cada clase. Una de las posibilidades es usar la media de la clase (o el conjunto formado por la media de las diversas subclases que componen la clase) como prototipo de la misma. Cuando se habla de espacios euclídeos (representación vectorial), hallar la media es un problema sencillo, pero no así si usamos la representación por cadenas. En dicho caso, el problema de hallar la cadena media es NP-Duro.

Así, se pasan a definir aproximaciones sobre la cadena media para dichos usos. La aproximación clásica es la cadena mediana. Nuevas aproximaciones se proponen siguiendo diversos esquemas; en primer lugar, una aproximación voraz, que no resulta competitiva respecto a la cadena mediana. Posteriormente, se presentan dos aproximaciones basadas en perturbación iterativa que sí resultan competitivas a nivel de clasificación con respecto a la cadena mediana, a costa de un mayor coste computacional.

Posteriormente, se tratan diversos aspectos interesantes sobre este tema. Se da una definición alternativa de cadena media (que no otorga diferencias significativas con respecto a la definición clásica). Se introducen técnicas específicas de reducción de coste computacional en los algoritmos de perturbación iterativa (a costa de una cierta degradación en la calidad de los prototipos). Se realiza también el cálculo exacto de la cadena media mediante Ramificación y Poda, revelando los resultados la buena calidad de las aproximaciones propuestas respecto a la solución exacta.

Seguidamente, se aplican las aproximaciones a la realización de agrupamientos en las clases, mostrando un mejor comportamiento de las nuevas propuestas respecto al método usado habitualmente (k-medianas). Se hace una aplicación de las aproximaciones propuestas también para cadenas cíclicas, revelando resultados competitivos respecto al uso de la cadena mediana también en esta circunstancia. Por último, se realiza una comparativa entre los métodos paramétricos y los basados en distancias para este problema de clasificación, revelando un mejor comportamiento de los métodos no paramétricos a medida que aumenta el número de agrupamientos usado en cada clase.

Resum

Al camp del Reconeixement de Formes, les técniques de classificació basades en distància (i més específicament el classificador k-NN) necessiten de l'obtenció de prototips adequats per cada classe. Una de les possibilitats és fer server la mitjana de la classe (o el conjunt format per les mitjanes de diverses subclasses que composen la classe) com prototip d'eixa classe. Quan parlem d'espais euclidis (representació vectorial), trobar la mitjana és un problema senzill, però no ho és quan fem servir la representació per cadenes. En tal cas, el problema de trobar la cadena mitjana és NP-Dur.

Així, es passa a definir aproximacions sobre la cadena mitjana en eixos usos. L'aproximació clásica és la cadena mediana. Noves aproximacions es proposen usant diversos esquemes; en primer lloc, una aproximació golafre, que no es mostra competitiva respecte a la cadena mediana. Posteriorment, es presenten dues aproximacions basades en pertorbació iterativa que sí són competitives respecte a la cadena mediana, amb el problema de presentar major cost computacional.

Posteriorment, es tracten diversos aspectes interessants sobre este tema. Es proposa una definició alternativa (que no provoca diferències significatives respecte a la definició clàsica). S'introdueixen tècniques específiques de reducció del cost computacional als algorismes de pertorbació iterativa (a canvi d'una certa degradació en la qualitat dels prototips). Es realitza també el càlcul exacte de la cadena mitjana usan Ramificació i Poda, revelant els resultats la bona qualitat de les aproximacions propostes respecte a la solució exacta.

A continuació, s'apliquen les aproximacions a la realització d'agrupaments dins les classes, donant millor comportament les noves propostes que el mètode emprat habitualment (k-medianas). Es fa una aplicació de les aproximacions proposades també per a cadenes cícliques, mostrant resultats competitius respecte a l'ús de la cadena mediana també en estes circumpstàncies. Per últim, es fa una comparativa entre els mètodes paramètrics y els basats en distàncies per este problema de classificació, revelant un millor comportament els mètodes no paramètrics a mesura que augmenta el nombre d'agrupaments usats en cada classe.

Summary

In the Pattern Recognition field, good prototypes are needed for each class when using distance-based classification techniques (and more specifically, k-NN classifiers). One possibility is to use the mean of the class as prototype of the class (or, when a partition in subclasses of the class is available, the set of the means from each subclass). In euclidean spaces (vectorial representation), to compute the mean is a simple problem. However, when data is represented by strings, the problem of computing the so-called median string is NP-Hard.

Therefore, approximations to the median string are defined for classification purposes. The classic approximation is the set median string. New approximations are proposed following several methods. The first proposed method is a greedy method whose results do not improve set median results. Afterwards, two approximated methods based on iterative perturbation are proposed. These new approximations achieve better classification results than set median, although their computational cost is higher.

Afterwards, some interesting topics are studied. An alternative definition of median string is proposed (this new definition does not show practical differences with the classical definition). Some specific cost reduction techniques are introduced for the iterative perturbation methods (but producing worse prototypes). The exact calculation of the median string is also performed using Branch and Bound. The exact results show that the proposed approximations are very accurate.

Then, the proposed approximations are use for clustering methods (to obtain several clusters in each class). This new proposal presents a better behaviour than the classical method (k-medians). The proposed approximations are also used in cyclic strings. The results reveal that the approximations to the median string outperform the set median in classification in cyclic strings. Finally, parametric methods and distance-based methods are compared for classification in string sets. This comparation reveals that non-parametric methods perform better as the number of clusters of each class increases.

Prólogo

Dentro de la disciplina del Reconocimiento de Formas (RF), la tarea tradicionalmente más importante es la tarea de clasificación: dado un objeto desconocido, indicar a qué clase (de un conjunto determinado previamente) pertenece. Para llevar a cabo esta tarea mediante métodos automáticos, es necesario establecer un proceso que transforme los objetos reales en representaciones tratables automáticamente.

Una de las opciones es representar los objetos mediante vectores, donde cada una de las componentes del vector indica una característica del objeto. Otra posible opción es la codificación de dicho objeto como una cadena en un determinado alfabeto.

Cuando los objetos adquieren representación vectorial se establece una relación entre ellos debida a la distancia euclídea, de manera que los objetos más cercanos (a menor distancia) se parecen más entre sí. De la misma manera, para la representación por cadenas es posible adoptar un criterio que mida lo distintas que son dos cadenas, pudiendo considerarse como una medida de distancia entre ellas.

Así, a la hora de modelar un conjunto de objetos representados mediante vectores es posible escoger el vector media de dicho conjunto (cuya definición se basa en la distancia euclídea) como representante. El conjunto de prototipos obtenidos de esta manera se aplica posteriormente en las tareas de clasificación. De igual manera, dada una definición de distancia entre cadenas, es posible definir el concepto equivalente para conjuntos de cadenas: la cadena media.

La obtención de la cadena media es un problema NP-Duro. Esto hace que no pueda ser computable en un tiempo razonable para cualquier conjunto de cadenas y, por tanto, no sea recomendable como modelo de un conjunto de cadenas. Frente a esta posibilidad, habitualmente se ha optado por usar la cadena mediana (cadena más centrada de un conjunto de cadenas dado) como representante, ya que su obtención tiene complejidad polinomial.

El trabajo desarrollado en esta tesis investiga las posibilidades de usar la cadena media como prototipo en tareas de clasificación para objetos codificados mediante cadenas, específicamente para clasificadores basados en distancias (en particular, clasificadores por k vecinos más cercanos). Como el problema de la cadena media no es computacionalmente abordable, se busca la obtención de aproximaciones razonablemente buenas en un tiempo asequible, y comparar la calidad de dichas aproximaciones en tareas de clasificación con respecto al prototipo tradicional (la cadena mediana). También se busca la aplicación de dichas aproximaciones a la tarea de obtención de agrupamientos de conjuntos de datos.

Así, en el **Capítulo 1** se introducen los conceptos preliminares con los que se va a trabajar a lo largo de la tesis: clasificadores basados en distancias, técnicas de agrupamiento, medidas de distancia entre cadenas y definición del problema de la cadena media.

En el **Capítulo 2** se presenta una primera aproximación a la cadena media basada en un proceso voraz y se realizan experimentos que verifican la calidad de

dicha aproximación. Es en el **Capítulo 3** donde se realiza la primera aportación importante de esta tesis: se presentan dos métodos aproximados basados en perturbaciones iterativas sobre una cadena y se realiza una experimentación exhaustiva para verificar su calidad.

Los capítulos posteriores trabajan sobre derivaciones y aplicaciones de estos algoritmos propuestos y pueden estudiarse de manera independiente entre sí. En el **Capítulo 4** se propone una definición alternativa de cadena media, puesto que la definición clásica presenta ciertos problemas en algunos casos. Dicha definición se incorpora en los algoritmos aproximados para verificar su posible influencia.

De igual manera, en el **Capítulo 5** se realizan ciertas variaciones sobre los algoritmos propuestos a fin de obtener aproximaciones con un menor coste temporal. La calidad en tareas de clasificación de estas nuevas aproximaciones se comprueba también experimentalmente.

En el **Capítulo 6** se aborda el problema de la obtención exacta de la cadena media mediante el uso de la técnica de Ramificación y Poda, haciendo el estudio teórico de las cotas necesarias para aplicar dicho esquema. Igualmente, se comparan estas soluciones exactas obtenidas con las aproximaciones propuestas (incluyendo las soluciones obtenidas mediante la definición alternativa del Capítulo 4).

El **Capítulo 7** hace una aplicación de los algoritmos aproximados propuestos a tareas de obtención de agrupamientos en conjuntos de cadenas. La calidad de los diversos agrupamientos aplicando las distintas técnicas propuestas es medida también en términos de error de clasificación, usando los agrupamientos como nueva fuente de extracción de prototipos y usándolos en clasificación.

El **Capítulo 8** hace una aplicación de los algoritmos propuestos sobre cadenas medias cíclicas, y compara de nuevo a nivel de clasificación su calidad como prototipos.

Finalmente, en el **Capítulo 9** se hace una comparación entre los métodos de clasificación basados en distancias, usando como prototipos las cadenas obtenidas mediante los algoritmos propuestos, y los métodos de clasificación probabilísticos basados en modelos ocultos de Markov.

Esta tesis se completa con un capítulo de conclusiones y trabajos futuros, donde se resumen los principales logros y conclusiones alcanzados y las posibles vías de continuidad de este trabajo. Además, se proporcionan varios apéndices con información adicional que permite profundizar en algunas de las conclusiones obtenidas.

Índice general

Capítulo 1

Introducción

Este capítulo va destinado a introducir los conceptos básicos con los cuales se va a tratar a lo largo de la tesis. Se introducen ciertas generalidades sobre el Reconocimiento de Formas, y se tratan temas que resultan básicos para el desarrollo del resto de la tesis: clasificadores basados en distancias, técnicas de extracción de prototipos, técnicas de agrupamiento y medidas de distancia entre cadenas. Finalmente, se introduce el problema principal con el que se trata: la cadena media.

1.1. Generalidades sobre el Reconocimiento de Formas

El Reconocimiento de Formas (RF) se define como la tarea de adivinar o predecir usando un elemento automático (computador) la naturaleza desconocida de una observación [14]. Así pues, en el RF se parte de un objeto real y se trata de averiguar mediante un computador "qué es". Según el concepto de "qué es" que sea necesario aplicar, las tareas de RF se subdividen en tareas de **clasificación** y en tareas de **interpretación** [15].

En las tareas de clasificación se trata de obtener a qué clase pertenece el objeto. Es decir, se tiene el universo de los objetos subdividido en varios conjuntos, y dado un objeto desconocido se pretende averiguar en cuál de esos conjuntos hay que incluirlo. En este tipo de tareas, el número de clases suele ser finito y reducido. Es el caso de tareas como Reconocimiento Óptico de Caracteres (OCR, *Optical Character Recognition*) [59].

En las tareas de interpretación se busca una descripción más detallada del objeto. Es decir, se supone que la complejidad del objeto es tal que no basta con asignarlo a una posible clase dentro del universo de los objetos, sino que hay que obtener una descripción del objeto con un cierto sentido dentro de ese universo. Es el caso de tareas como el Reconocimiento Automático del Habla en su vertiente de habla continua [65].

Las tareas de interpretación son en su mayoría mucho más complejas que

las de clasificación, las cuales se han convertido en las tareas por excelencia del RF. En cualquier caso, para ambos tipos de tarea es necesario representar el objeto a tratar en un formato que sea tratable por un computador, es decir, codificarlo. Según el esquema de codificación seguido, se habla de **Reconocimiento Geométrico de Formas** (RGF) (también llamado estadístico) [15] o de **Reconocimiento Sintáctico de Formas** (RSF) (también llamado estructural) [17].

En el RGF, la representación de los objetos se realiza de manera vectorial, con el llamado vector de características. Esta representación hace que sea sencillo aplicar sobre los objetos análisis estadístico (de ahí el sobrenombre de reconocimiento estadístico). Por otro lado, en el RSF la representación de los objetos se realiza habitualmente mediante cadenas o estructuras más complejas (grafos o árboles, de ahí que se le llame estructural). Para esta representación se pueden aplicar los resultados aportados por la teoría de lenguajes formales [26]. Sin embargo, hay que destacar que la frontera que separa ambas ramas es bastante difusa y que algunas técnicas que se aplican de manera más habitual en una de las ramas pueden adaptarse a la otra.

Siguiendo ya sea el esquema de RGF como el de RSF, el funcionamiento de un sistema de RF es siempre el mismo. Existe una primera fase de entrenamiento, en la cual se le aportan al sistema los datos necesarios para que, mediante un determinado proceso, aprenda la naturaleza de cada una de las clases[1]. Tras esta fase de entrenamiento, se puede pasar al uso del sistema, en el cual se le aporta al sistema el objeto desconocido y aquél, tras realizar un proceso, responde a qué clase pertenece. Atendiendo a la forma del proceso que lleva a cabo el sistema de RF tanto en la fase de entrenamiento (cómo representa la naturaleza de las clases) como en la de uso (cómo usa la representación de las clases para hacer la clasificación), se habla de **métodos de clasificación paramétricos** y de **métodos de clasificación no paramétricos**. Esta clasificación se hace según los modelos que se usan para representar las clases.

En los métodos de clasificación paramétricos los modelos vienen formados por ciertas características fundamentales (parámetros). Es decir, en la fase de entrenamiento se extraen a partir de los datos los parámetros de los modelos y en la fase de uso se utilizan para verificar la pertenencia o no del objeto a dicha clase. El método paramétrico por excelencia es el método bayesiano [15]. En este caso, el modelo es una distribución estadística. Así, en la fase de entrenamiento se estiman los parámetros de esa distribución estadística que siguen los datos de cada clase. En la fase de uso se utilizan test estadísticos para saber con qué probabilidad pertenece el objeto a una cierta clase (es decir, con qué probabilidad pertenece a la distribución asociada).

Los métodos de clasificación no paramétricos se caracterizan por la ausencia de un modelo basado en parámetros, ya que en este caso son los propios objetos proporcionados en la fase de entrenamiento los que actúan como modelos. Los métodos no paramétricos más conocidos son las ventanas de Parzen y la regla de los k vecinos más próximos (k-NN, *k Nearest Neighbours*) [15]. El caso de k-NN

[1] Por simplicidad, suponemos que nos enfrentamos a una tarea de clasificación.

es el de una regla extremadamente simple y con excelentes resultados, la cual se basa en una medida de disimilitud o distancia entre los objetos. De esta manera, un objeto se clasifica en una clase u otra según el valor de la distancia entre él y los objetos proporcionados en la fase de entrenamiento. Cuando los objetos son representados como vectores numéricos, una medida de distancia extremadamente usual es la distancia euclídea. Sin embargo, si los objetos se representan por cadenas, la elección de la medida de disimilitud es un tema más delicado, ya que éstas no tienen una representación clara en un espacio euclídeo y, por tanto, requieren de la definición de otro tipo de distancias. La distancia más popular entre cadenas es la distancia de edición (o de Levenshtein) [80], aunque existen muchas otras propuestas que serán estudiadas con más detenimiento en la Sección 1.5.

A lo largo de esta tesis, los problemas de RF a los que nos vamos a enfrentar serán problemas de clasificación de objetos representados por cadenas mediante técnicas no paramétricas (k-NN).

1.2. Clasificadores basados en distancias

La tarea fundamental dentro del Reconocimiento de Formas que se aborda en esta tesis es la tarea de clasificación. Como indicamos en la Sección 1.1, los métodos de clasificación se subdividen en paramétricos y no paramétricos. Vamos a hacer en primer lugar una revisión del método paramétrico más frecuente, el método bayesiano, que nos aporta ciertos resultados interesantes para nuestro desarrollo posterior con métodos no paramétricos.

Sea un conjunto de clases $\Omega = \{\omega_1, \omega_2, \ldots, \omega_s\}$ y sea un objeto representado por x. Suponiendo que se conoce la probabilidad *a priori* de cada una de las clases $P(\omega_i)$ y que se conoce la distribución de probabilidad con que ω_i puede generar el objeto x, es decir, $p(x|\omega_i)$, la probabilidad de que x se asigne a ω_i siguiendo la regla de Bayes [15] es:

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)} \tag{1.1}$$

donde $p(x) = \sum_{i=0}^{s} p(x|\omega_i)P(\omega_i)$.

Así, el proceso de clasificación con el método bayesiano se basa en obtener $P(\omega_i|x)$ usando las probabilidades *a priori* calculadas previamente, para clasificar x en aquella clase ω_i que dé la máxima probabilidad [15].

Vamos a examinar ahora la relación que tiene con el error de clasificación el usar la clasificación mediante la regla de Bayes. En primer lugar, notaremos la probabilidad de acertar al clasificar x en la clase ω_i mediante un cierto criterio de clasificación como $A(\omega_i|x)$. Claramente, se da que $A(\omega_i|x) = P(\omega_i|x)$, con lo cual se puede definir el riesgo condicional del error como $R(\omega_i|x) = 1 - A(\omega_i|x) = 1 - P(\omega_i|x)$. Ya que $\sum_{j=1}^{s} P(\omega_j|x) = 1$, tendremos que:

$$R(\omega_i|x) = \sum_{j=1}^{s} P(\omega_j|x) - P(\omega_i|x) = \sum_{j=1, j\neq i}^{s} P(\omega_j|x)$$

Siendo D el criterio de clasificación tomado y notando entonces por $D(x)$ la clase en la que se clasifica x siguiendo dicho criterio, el error acumulado global será equivalente a:

$$R = \int_E R(D(x)|x) \cdot p(x)\mathrm{d}x \tag{1.2}$$

donde E es el conjunto de datos sobre el que se está haciendo la clasificación.

Por tanto, la regla de clasificación óptima será aquella que minimice el valor de R dado por (1.2), lo que equivale a minimizar el error cometido individualmente para cada uno de los objetos x. Así pues, esta regla de clasificación D deberá ser:

$$D(x) = \omega_i = \underset{i=1,\ldots,s}{\operatorname{argmin}} R(\omega_i|x) = \underset{i=1,\ldots,s}{\operatorname{argmin}} \sum_{j=1,j\neq i}^{s} P(\omega_j|x) =$$

$$= \underset{i=1,\ldots,s}{\operatorname{argmax}} 1 - \sum_{j=1,j\neq i}^{s} P(\omega_j|x) = \underset{i=1,\ldots,s}{\operatorname{argmax}} P(\omega_i|x)$$

es decir, la regla de clasificación que minimiza el error es aquella que toma la clase de máxima probabilidad *a posteriori* dado x, o sea, la regla de clasificación de Bayes.

Por tanto, tenemos que con el método de clasificación bayesiano obtenemos el mínimo error. El problema es que este método requiere una estimación fiable de la densidad de probabilidad $p(x|\omega_i)$ la cual no es, en muchos casos, fácil de obtener. Generalmente, se asume que dicha distribución sigue una forma predeterminada (por ejemplo, una distribución normal) y se calculan los parámetros de dicha distribución predeterminada mediante métodos de inferencia estadística [15].

El uso de métodos no paramétricos de clasificación viene a tratar de soslayar esta deficiencia de los métodos paramétricos. Estos métodos no paramétricos representan las probabilidades de clasificación mediante los propios datos, sin hacer ninguna estimación de parámetros a partir de los mismos.

Para ello se define inicialmente la probabilidad de que un objeto x se sitúe en una región $\mathcal{R}$ como:

$$P = \int_{\mathcal{R}} p(x')\mathrm{d}x'$$

Con esta definición, tenemos que P es un promedio de la densidad de probabilidad $p(x)$, con lo cual se puede estimar p mediante P. Si asumimos que existe un total de n muestras que siguen la distribución p, la probabilidad de que k de ellas se sitúen en la región $\mathcal{R}$ viene dada por la regla binomial $P(k) = \binom{k}{n} P^k(1-P)^{n-k}$. Para un valor de n lo suficientemente grande se dará que la esperanza de k vale $E(k) = nP$, con lo cual se puede aproximar (al ser P una estimación de p) que $p \simeq \frac{k}{n}$.

Igualmente podemos asumir que la distribución $p(k)$ es continua y no varía en la región $\mathcal{R}$, es decir, $\mathcal{R}$ es lo bastante pequeña. Por tanto, será posible la aproximación $\int_{\mathcal{R}} p(x')\mathrm{d}x' \simeq p(x)V$, donde V es el volumen de $\mathcal{R}$.

A partir de estas dos asunciones, tendremos que $p(x) \simeq \frac{k}{nV}$. La convergencia de esta probabilidad se dará cuando el número de datos n tiende a infinito, momento en el cual la aproximación se puede tomar como igualdad. Por tanto, tendríamos $p(x) = \frac{k}{nV}$ en el caso ideal, en el cual V se aproxima a cero y k a infinito. En los casos reales, los volúmenes de las regiones son relativamente grandes y el número de datos es limitado, con lo cual la estimación de dicha probabilidad debe hacerse fijando alguno de esos parámetros. Fijar el volumen V nos lleva a las ventanas de Parzen, mientras que fijar k nos lleva a la regla de los k-vecinos más cercanos [15].

En nuestro caso vamos a optar por fijar k, ya que fijar los volúmenes de las regiones lleva en muchas ocasiones a problemas de sensibilidad respecto a la elección de un volumen inicial. Al fijar k, lo que realmente ocurre es que hacemos el volumen de las regiones variable en función de los datos, ya que la región asociada a x será una hiperesfera centrada en el propio x y cuyo radio será el suficiente para contener k muestras de los datos disponibles.

Notaremos para cada clase ω_i su número de muestras disponibles por n_i y el número de muestras incluídas en una hiperesfera de volumen V por k_i. Por tanto, dada la definición de $p(x)$ previa, tendremos que:

$$p(x|\omega_i) = \frac{k_i}{n_i V}$$

Igualmente, tendremos que la probabilidad *a priori* de la clase ω_i será:

$$P(\omega_i) = \frac{n_i}{n}$$

Así, dado que la probabilidad incondicional $p(x)$ estaba definida como $p(x) = \frac{k}{nV}$, podemos calcular la probabilidad *a posteriori* usando la regla de Bayes dadas estas definiciones de probabilidades *a priori*. Dicha probabilidad será:

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)} = \frac{k_i}{k} \tag{1.3}$$

Por tanto, la probabilidad de que x se clasifique en cierta clase vendrá dada por el número de muestras de esa clase que quedan comprendidas en la hiperesfera definida con centro en x y que contiene k muestras. Evidentemente, las k muestras que quedan contenidas en dicha hiperesfera son las k muestras más cercanas a x según una cierta distancia definida entre objetos de dicho universo.

Por tanto, la regla de los k vecinos más cercanos (notada normalmente por k-NN, de k *Nearest Neighbour*) se puede resumir en, dado un objeto x, encontrar las k muestras disponibles más cercanas al mismo usando una cierta distancia definida, y clasificar x en la clase a la que pertenezcan la mayoría de las k muestras. Un caso particular es la regla del vecino más cercano, es decir, con $k = 1$; en ese caso, se clasifica x en la clase a la que pertenece la muestra más cercana a ella. Dicha regla se usa en el caso de que k-NN produzca un empate (es decir, dos o más clases aportan el mismo número de vecinos que además es el máximo).

Una propiedad interesante de la regla k-NN es su relación con el error cometido usando la clasificación por regla de Bayes. Sabemos que el error cometido por la regla de Bayes, notado generalmente por R^*, es el menor posible, y por tanto será la cota inferior del error que se puede obtener usando cualquier otra regla de clasificación (lo cual incluye a k-NN). Igualmente, es posible demostrar que existe una cota superior del error cometido por k-NN, que está fuertemente relacionado con el error de Bayes. En concreto, se puede demostrar que el error R de clasificación por k-NN cumple la relación [15]:

$$R^* \leq R \leq R^*(2 - \frac{s}{s-1}R^*) \tag{1.4}$$

donde s recordemos que es el número de clases con el que estamos trabajando.

El error usando k-NN se aproxima al error de Bayes a medida que el número de muestras y el número de vecinos k empleado tiende a infinito [15]. Sin embargo, esta situación en la práctica es inviable, ya que ni se dispone de un número infinito de datos ni se puede hacer un cálculo de infinitos vecinos. Por tanto, lo ideal es obtener una muestra reducida y lo suficientemente representativa, de manera que los k vecinos que se escojan sean representativos de su clase y permitan una clasificación adecuada del objeto x. La obtención de esa muestra reducida y representativa se lleva a cabo mediante las técnicas de extracción de prototipos que se presentan en la Sección 1.3.

1.3. Técnicas de extracción de prototipos

En esta Sección vamos a dar cuenta de técnicas que nos permiten extraer los prototipos adecuados en técnicas no paramétricas, en concreto, en clasificación por k vecinos más cercanos (k-NN). La necesidad de estas técnicas estriba en dos factores fundamentales.

Por un lado, el clasificador k-NN resulta sensible a la presencia de prototipos "ruidosos", es decir, objetos que a pesar de pertenecer a la clase no resultan representativos de la misma o que se sitúan en zonas de confusión con otra clase. Por tanto, estos elementos pueden llevar a confusión con elementos de otra clase (por lo cual se les llama "fuera de norma" o con el término anglosajón *outliers*). Así pues, sería conveniente la definición de un proceso que nos permitiera distinguir cuáles de esos prototipos de los que se dispone inicialmente deben ser eliminados y cuáles deben ser conservados. A este tipo de técnicas se les llama técnicas de edición [83, 13].

Por otro lado, una de las características del clasificador k-NN es su alto coste computacional, pues debe comparar con todos los prototipos presentes y ello supone un cálculo de distancia para cada uno de ellos. Por tanto, la necesidad de reducir el número de prototipos al mínimo sin perder calidad en la clasificación (es decir, sin aumentar la tasa de error) también es evidente. Las técnicas destinadas a seleccionar ese conjunto mínimo de prototipos se conocen como técnicas de condensado [25] y están especialmente indicadas para el clasificador NN.

Pasamos ahora a describir las técnicas de edición y condensado básicas propuestas a lo largo de la historia del RF y una técnica más (que puede considerarse una técnica mixta) basada en el concepto de media.

1.3.1. Técnicas de edición

Como se ha comentado previamente, el objetivo de las técnicas de edición es la eliminación del conjunto de prototipos de aquellas muestras que, debido a su alto grado de ruido o a su situación en ciertas zonas de confusión con otras clases, provocan una mala clasificación de las muestras a clasificar. Intuitivamente, se ve que los prototipos que presenten estas características deben ser aquellos que están proximos a los prototipos de otras clases (y que en general se encuentran en las cercanías de la frontera de decisión entre clases).

Un método general de edición actúa usando una cierta regla de clasificación sobre un conjunto de prototipos, de manera que va eliminando de ese conjunto de prototipos aquellos que resultan mal clasificados siguiendo tal regla de clasificación. Ese proceso de clasificación-eliminación se va repitiendo hasta que se alcance cierto criterio de terminación (por ejemplo, no eliminar más prototipos).

El algoritmo de edición más popular es el *multiedit* [13], el cual presenta como característica fundamental su tendencia asintótica al error mínimo teórico cuando el conjunto original de prototipos tiende a infinito. El esquema básico de funcionamiento de dicho algoritmo es:

1. Partición: realizar la partición del conjunto total de prototipos S en N subconjuntos disjuntos, $S_1, S_2, \ldots, S_N$, con $N \geq 3$.
2. Clasificación: clasificar las muestras de S_i usando como prototipos las muestras de S_j para $j = 1, \ldots, N$, $j \neq i$.
3. Edición: eliminar todas las muestras clasificadas incorrectamente en el paso 2.
4. Unión: unir todas las muestras disponibles de nuevo en S
5. Terminación: si en las últimas I iteraciones no se ha producido ninguna eliminación, dar S como conjunto editado final, si no volver al paso 1.

Aplicando dicho algoritmo sobre un conjunto de muestras, se consigue un nuevo conjunto de prototipos de menor tamaño y con menos prototipos ruidosos, lo cual favorece la mejor tasa de aciertos del clasificador k-NN. Sin embargo, el conjunto final de prototipos suele seguir siendo demasiado grande como para permitir un funcionamiento eficiente a nivel computacional del clasificador k-NN.

1.3.2. Técnicas de condensado

El objetivo de las técnicas de condensado es el aumento de la eficiencia computacional de los clasificadores k-NN evitando el aumento en el error de clasificación de los mismos. En realidad, su aplicación se hace básicamente para

el clasificador NN, pues para k-NN es más favorable que exista un número alto de prototipos. Se trata de hacer una adecuada selección de prototipos, de manera que finalmente se dejan en el conjunto aquellos que resultan fundamentales para la adecuada clasificación de las muestras de prueba. Estos prototipos son aquellos que marcan adecuadamente el límite de las clases (es decir, sus fronteras de decisión). Dichos prototipos se caracterizan por, o bien estar en las cercanías de dicha frontera de decisión o bien por ser prototipos que representan una alta densidad de muestras.

Si nos centramos en el caso de que sean prototipos cercanos a las fronteras de decisión, el problema que se plantea en este caso es inmediato: si los prototipos ruidosos se sitúan en la cercanía de los límites con otras clases, entonces las técnicas de condensado van a tomar con gran probabilidad muestras ruidosas como prototipos, con la consiguiente degradación de las prestaciones de clasificación. Esto provoca que los métodos de condensado en estos casos no sean usados por sí mismos, sino que suelen utilizarse tras un paso previo de edición cuyo objetivo es, precisamente, hacer que el condensado no obtenga muestras ruidosas como prototipos.

Este hecho es aplicable a condensados basados en la propiedad de consistencia [25], pero existen otros tipos de condensado que no requieren de esta edición inicial. Es el caso del condensado LVQ [39], que se caracteriza por la búsqueda de prototipos que representen una alta densidad de muestras.

Las técnicas de condensado se dividen, a su vez, en técnicas de selección [25, 18, 74, 40] (los prototipos finales son un subconjunto de los originales) y en técnicas de reemplazo [39, 19, 62, 84, 10] (los prototipos finales no son necesariamente un subconjunto de los originales y se obtienen por combinación de éstos). El algoritmo más popular de condensado es el *condensing* [25], el cual actúa de acuerdo con el siguiente esquema:

1. Inicialmente, una muestra va al conjunto S y el resto va al conjunto G.
2. Se toma ordenadamente cada muestra de G y se clasifica (habitualmente mediante NN) usando S como conjunto de prototipos. Si la muestra resulta mal clasificada, se transfiere a S, si no, permanece en G.
3. Si tras el paso 2 no ha habido transferencias a S o si G resulta vacío, se devuelve S como nuevo conjunto de prototipos; si no, volver al paso 2.

Como se ve, este es un esquema de selección pues toma como prototipos finales muestras del conjunto original. Este método es un método voraz (se toma una ordenación de las muestras y se va haciendo la selección según dicha ordenación) y es sensible a la inicialización (la muestra que se asigna inicialmente a S).

1.3.3. El uso de la media como prototipo

Como alternativa a los esquemas clásicos de selección de prototipos, existe una idea intuitiva alternativa. Esta idea consiste en tomar como prototipo de la

clase aquél que se encuentre en el centro geométrico del conjunto de muestras dentro de su espacio de representación.

Dado un conjunto de muestras $C = \{\mathbf{y^1}, \mathbf{y^2}, \ldots, \mathbf{y^{|C|}}\}$, donde $\mathbf{y^j} \in \mathbb{R}^n$ (espacio euclídeo de dimensión n) para $j = 1, \ldots, |C|$, y $|C|$ representa el tamaño (número de muestras) de C, el *vector media* de C correspondería a aquel vector que, en promedio, está a menor distancia de todas las muestras de dicho conjunto. Es decir, suponiendo la distancia euclídea d, el vector media del conjunto de muestras es aquel vector $\bar{\mathbf{x}}$ que minimiza:

$$\sum_{j=1}^{|C|} d(\bar{\mathbf{x}}, \mathbf{y^j}) = \sum_{j=1}^{|C|} ||\bar{\mathbf{x}} - \mathbf{y^j}||^2 = \sum_{j=1}^{|C|} \sum_{i=1}^{n} (\bar{x_i} - y_i^j)^2 \tag{1.5}$$

donde $\bar{x_i}$ representa la i-ésima componente del vector $\bar{\mathbf{x}}$. Por tanto, tendríamos que $\bar{\mathbf{x}} = \text{argmin}_{\mathbf{x} \in \mathbb{R}^n} \sum_{j=1}^{|C|} \sum_{i=1}^{n} (x_i - y_i^j)^2$. En principio, la búsqueda de este vector sería inviable dado que el número de elementos de $\mathbb{R}^n$ es infinito y resultaría extremadamente costoso hacer una búsqueda exhaustiva. Sin embargo, existe una solución analítica al problema de minimización propuesto. Si se deriva la ecuación dada en (1.5) y se iguala a cero, al despejar el valor de $\mathbf{x}$ nos da que el vector media tiene el valor:

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{|C|} y^i}{|C|} \tag{1.6}$$

y esta solución analítica podría usarse como representante de la clase C.

Este uso podría verse como una técnica de condensado por reemplazo, pues crea un nuevo conjunto de prototipos (uno por clase) a partir de los que se tenían previamente. El problema es que usar sólo este prototipo puede acarrear una serie de dificultades al usar clasificadores k-NN que puede hacer que los resultados no sean los adecuados.

Por un lado, se está asumiendo que la media es un adecuado representante de la clase. Sin embargo, las muestras de dicha clase siguen una cierta distribución de probabilidad para la cual la media no tiene por qué ser un representante adecuado. Incluso si nos vamos a un caso más específico como es el de distribuciones normales (gaussianas) existe un parámetro determinante que no se tiene en cuenta: la varianza de dicha distribución. Sin dicho parámetro es imposible hacerse una idea de la dispersión de los datos y, por tanto, determinar si la muestra a clasificar presenta más probabilidad de pertenecer a la gaussiana generada por una media o por otra. Por tanto, las fronteras de separación de las distintas clases no quedan adecuadamente determinadas (pues representar las clases sólo con la media indica que se asume que todas las clases presentan la misma varianza).

Para evitar tal problema, la idea sería representar la clase como una combinación de varias gaussianas. Llevado al extremo, cada muestra generaría una

gaussiana y la clasificación sería equivalente al uso de un clasificador por vecino más cercano con todas las muestras como prototipos [15]. En un caso menos extremo, se tomaría la clase como una unión de varias subclases, y de cada una de estas subclases se extraería la media como representante y generadora de la gaussiana correspondiente. El conjunto de medias extraído para cada subclase sería el conjunto de prototipos de dicha clase.

Esta aproximación sería, claramente, mucho más precisa que extraer una sola media por clase. Además, a diferencia de si se extrae una sola media por clase, posibilita el uso de clasificadores k-NN al tener más de un prototipo disponible por cada una de las clases.

El problema de obtener las subclases posibles de una clase es un problema equiparable al de obtener las clases de un conjunto de muestras no etiquetado, es decir, al problema de la búsqueda de agrupamientos o *clustering*. Por tanto, se llevará a cabo una revisión de los métodos de agrupamiento en la Sección 1.4.

En este apartado hemos visto como, desde el punto de vista de la representación vectorial, el vector media es un prototipo adecuado para los problemas de clasificación. Inmediatamente surge la posibilidad de extender dicho razonamiento al caso de la representación en la que los prototipos son cadenas. Sin embargo, no existe fórmula equivalente a la dada en la Ecuación (1.6) y, tal y como veremos en la Sección 1.6, será necesario buscar una formulación alternativa para obtenerla.

1.4. Técnicas de agrupamiento

Las técnicas de agrupamiento buscan la obtención de agrupamientos naturales dentro de un conjunto de muestras; es decir, vienen a hacer una división más o menos natural dentro del conjunto de datos disponible. Dichas técnicas desempeñan un papel importante dentro del aprendizaje no supervisado [15], pero también son utilizadas para poder distinguir subclases dentro de una clase determinada.

La obtención de agrupamientos naturales responde a muy diversos criterios. En general, el factor fundamental es el uso de una medida de disimilitud entre agrupamientos. Siguiendo esta medida, lo que se consigue es distinguir si dos muestras pertenecen al mismo agrupamiento (disimilitud baja) o no (disimilitud alta). Esta medida también se aplica agrupamiento a agrupamiento, a fin de propiciar fusiones y separaciones entre los diversos agrupamientos obtenidos. El objetivo de esto es conseguir agrupamientos lo más adecuados posibles.

Formalmente, un proceso de agrupamiento sobre un conjunto de datos $X = \{x_1, x_2, \ldots, x_n\}$ va a obtener c subconjuntos $X_1, X_2, \ldots, X_c$, $X_i \neq \emptyset$ para $i = 1, \ldots, c$, tales que $X = X_1 \cup X_2 \cup \cdots \cup X_c$ y disjuntos entre sí ($X_i \cap X_j = \emptyset \quad \forall i \neq j$), de manera que los datos de cada X_i tengan entre sí una alta similitud. Esta imposición de que sean conjuntos disjuntos entre sí es una asunción necesaria para los métodos clásicos, pero los métodos difusos no imponen esta restricción, permitiendo que ciertos elementos pertenezcan a más de un subconjunto con una cierta probabilidad (certidumbre). La medida de similitud ideal debería ser

invariante a posibles reescalados del conjunto de datos, a fin de conservar el concepto de "agrupamiento natural" en cualquier representación de los datos.

Según el método usado para hacer el particionado, se habla de diversas técnicas de agrupamiento, que a su vez forman familias dependiendo de la filosofía general que aplique el método. Las dos familias más importantes son las técnicas de agrupamiento jerárquicas [2, 29, 75] y las técnicas basadas en minimización del error cuadrático [15]. A continuación, expondremos la filosofía básica de cada una.

1.4.1. Agrupamientos jerárquicos

La filosofía del agrupamiento jerárquico se basa en partir de que cada muestra es de por sí un agrupamiento, y a partir de ahí se van fusionando agrupamientos según un cierto criterio paso a paso, sin posibilidad de que una muestra ya integrada en un agrupamiento pueda salir del mismo. El proceso se realiza hasta alcanzar el número de agrupamientos deseado o hasta que estos formen la partición más satisfactoria, siendo el punto final el que todas las muestras se encuentren en un solo agrupamiento. A este tipo de agrupamiento jerárquico se le denomina aglomerativo [2].

También existe la posibilidad de hacerlo en sentido contrario: se parte de un único agrupamiento inicial que recoge todas las muestras, y a cada paso uno de los agrupamientos se divide en dos según un cierto criterio. El punto final sería alcanzado cuando cada muestra forme por sí misma un agrupamiento. A este tipo de métodos se les conoce como divisivos, como contraposición a los descritos en el párrafo previo.

Un agrupamiento jerárquico se representa por un árbol llamado dendrograma, el cual muestra cómo se van produciendo los agrupamientos (Figura 1.1). También es representable por la secuencia de matrices de distancias entre agrupamientos que se va produciendo.

La primera cuestión evidente que surge al observar el proceso es definir el criterio que se usa para la fusión/separación, ya que según el criterio que se siga los agrupamientos obtenidos difieren. Estos criterios se basan en medidas de distancia sobre ciertos elementos de cada agrupamiento o sobre alguna propiedad en conjunto del agrupamiento [82]. Esto hace que los métodos de agrupamiento jerárquicos sean subóptimos, ya que no existe un criterio fundamentado formalmente que garantice llegar a un óptimo global.

Un primer método propuesto es el conocido como método de enlace simple (*single-link method*) o de los vecinos más cercanos [70, 58]. En este método, se computa la distancia entre los agrupamientos como la distancia entre las muestras más cercanas entre los mismos. Una vez hecho el cálculo, el paso de agrupamiento consiste en unir los dos agrupamientos que se encuentran a menor distancia según dicho criterio. El problema de esta aproximación es que puede provocar la unión en un agrupamiento de muestras que realmente se encuentran muy distantes entre sí, quedando una partición poco natural en algunas situaciones.

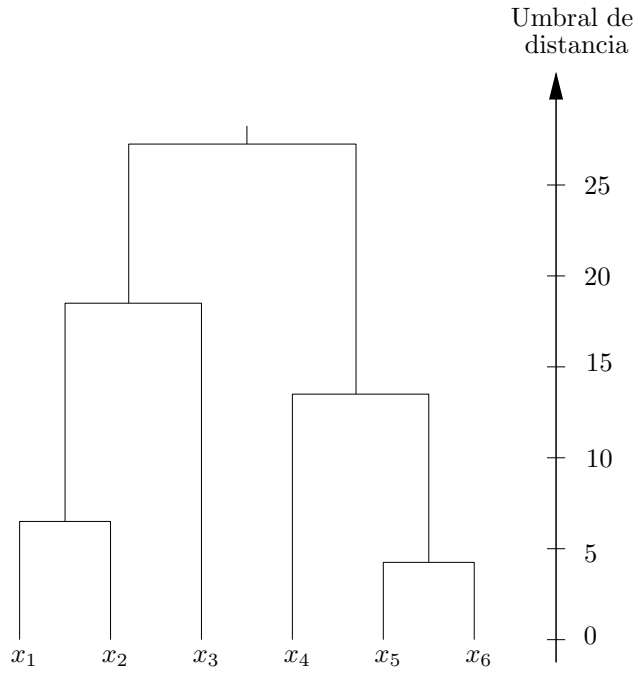


Figura 1.1: Dendrograma: representación gráfica de un agrupamiento jerárquico

De manera inmediata, surge el método con la filosofía opuesta, es decir, mantener en distintas particiones las muestras más alejadas. Este método se conoce como método de enlace completo (*complete-link method*) o de los vecinos más lejanos [73, 12]. Aquí, la definición de distancia entre agrupamientos equivale a la mayor distancia entre las muestras de uno y otro. De nuevo se unen los agrupamientos que con este criterio den menor distancia entre ellos, pero el efecto final es la alta cohesión interna de los agrupamientos.

Estos métodos presentados pueden ser usados teniendo una matriz de disimilitudes (o distancias) entre todas las muestras a agrupar. Sin embargo, en el caso de que las muestras se sitúen en un espacio euclídeo y, por tanto, se use la distancia euclídea, se puede recurrir al método de suma de cuadrados o de Ward [81]. La unión de agrupamientos se hace basándose en el sumatorio de distancias cuadráticas entre las muestras incluídas en el agrupamiento resultante, escogiendo aquella unión que dé una menor distancia relativa (es decir, el sumatorio de distancias cuadráticas se divide entre el número de muestras del agrupamiento para garantizar que el tamaño del agrupamiento no influye).

Una de las ventajas de los métodos de agrupamiento jerárquicos viene dada por la facilidad de implementación de todos ellos usando la llamada recurrencia de Lance-Williams [42]. Esta recurrencia da una fórmula general que permite hallar la disimilitud entre un agrupamiento y el que se formaría al unir otros dos, teniendo diversos parámetros que, según su instanciación, dan la medida de disimilitud de cada uno de los métodos.

1.4.2. Agrupamientos basados en sumas de cuadrados

El agrupamiento basado en sumas de cuadrados funciona tratando de minimizar la dispersión interna dentro de cada agrupamiento (distancia intraagrupamiento) y de maximizar la dispersión entre los distintos agrupamientos (distancia interagrupamiento). Todos los métodos siguen la máxima de, a partir del conjunto inicial de muestras, dividirlo en un número determinado de agrupamientos que optimicen dicho criterio. Sin embargo, la optimización de dicho criterio suele tener unos costes computacionales muy altos, por lo cual la mayor parte de los métodos aplicados son subóptimos. Dichos métodos se diferencian en el criterio de optimización (cómo miden en global las distancias intra e interagrupamiento) y en el procedimiento de optimización que se adopte.

Para los criterios de optimización, se define la matriz de covarianzas del conjunto total de muestras $\hat{\Sigma}$, la matriz de dispersión intraagrupamiento S_W y la matriz de dispersión interagrupamiento $S_B = \hat{\Sigma} - S_W$ [82]. Los criterios de optimización básicos son así:

- Minimización de la diagonal de S_W
- Minimización de $|S_W|/|\hat{\Sigma}|$
- Maximización de la diagonal de $S_W^{-1} S_B$
- Minimización de la diagonal de $\hat{\Sigma}^{-1} S_W$

Respecto al procedimiento de optimización adoptado, el más popular dentro de los métodos basados en sumas de cuadrados es el método de las k-medias, también conocido como *isodata* [15].

1.4.3. Métodos k-medias y k-medianas de obtención de agrupamientos

El método k-medias [15] es el más popular entre los métodos de agrupamiento basados en suma de cuadrados. Este método es subóptimo y se basa en minimizar el primer criterio que hemos nombrado previamente (minimización de la diagonal de S_W).

La idea del método es sencilla; se parte del número de agrupamientos que buscamos k, y se seleccionan inicialmente k muestras de entre las disponibles. Cada una de esas k muestras es la semilla de un agrupamiento, y se asocia el resto de muestras al agrupamiento definido por estas k iniciales, haciendo este agrupamiento por mínima distancia, y se calcula el valor del criterio descrito previamente. Tras este primer paso, se recalculan las medias de cada uno de estos agrupamientos y se usan para volver a agrupar todas las muestras; se calcula el valor del criterio de minimización y si resulta mayor o igual que en la iteración previa, se detiene el proceso. En caso contrario, se sigue iterando.

Este problema de obtener k agrupamientos admite un enfoque desde el punto de vista combinatorio que ha sido estudiado desde hace tiempo y que es conocido como el problema de las k-medias [60]. El problema de las k-medias se puede ver como una generalización del problema de hallar la media de un conjunto de muestras. Así, si el problema de hallar la media de un conjunto de muestras consiste en hallar el punto del espacio de representación cuya suma de distancias al conjunto de muestras sea mínima, el problema de las k-medias trata de hallar los k puntos (representantes) del espacio de representación tal que la suma de distancias de cada muestra a su representante más cercano sea mínima.

Así, si P es el conjunto de muestras sobre un cierto espacio de representación $\mathbb{E}$ con una cierta distancia d definida, dicha formulación viene expresada por hallar el conjunto de puntos $Q \subset \mathbb{E}$, con $|Q| = k$, que cumpla la Ecuación (1.7).

$$Q = \underset{Q \subset \mathbb{E}, |Q|=k}{\operatorname{argmin}} \sum_{p \in P} \min_{q \in Q} d(p, q) \tag{1.7}$$

Evidentemente, la solución de este problema es una solución al problema de obtener k agrupamientos, ya que cada representante generaría un agrupamiento formado por las muestras más cercanas al mismo. Este problema es NP-Duro [60] (dependiendo del espacio de representación $\mathbb{E}$), por lo cual sólo se pueden dar soluciones aproximadas de manera eficiente. Una de esas soluciones es la técnica de las k-medias descrita previamente.

Una variante que surge de manera inmediata es restringir el espacio de búsqueda al conjunto de muestras. Así, al igual que para hallar la mediana de un conjunto de muestras se busca la muestra de dicho conjunto que minimice la suma de distancias de cada muestra a dicha mediana, se puede generalizar

este concepto para k muestras. Así, si P es el conjunto de muestras sobre un cierto espacio de representación $\mathbb{E}$ con una cierta distancia d definida, dicha formulación viene expresada por hallar el conjunto de muestras $Q \subset P$ con $|Q| = k$ que cumpla la Ecuación (1.8).

$$Q = \underset{Q \subset P, |Q|=k}{\operatorname{argmin}} \sum_{p \in P} \min_{q \in Q} d(p, q) \tag{1.8}$$

Este problema es conocido como el problema de las k-medianas [60]. De nuevo, la solución de este problema nos genera k agrupamientos dentro del conjunto de muestras P, cada uno de ellos generado por un representante (muestra de Q). El problema de las k-medianas es un caso particular del problema general de localización de prototipos [54], y como tal es NP-Duro [35].

Por tanto, de nuevo debe de optarse por buscar soluciones aproximadas en un tiempo razonable. La técnica más usada para esta resolución aproximada es el algoritmo de las k-medianas [33]. La forma de proceder de este algoritmo es muy semejante a la del k-medias. Se parte de k muestras iniciales, las cuales actúan como representantes y generan sus agrupamientos asociados. Tras este agrupamiento inicial, el paso de reagrupamiento consiste en calcular las medianas de cada agrupamiento obtenido y usar este conjunto de medianas como nuevo conjunto de representantes. De este conjunto de representantes se obtienen los nuevos agrupamientos y se vuelve a iterar. El fin del proceso se da cuando los agrupamientos no varían de un paso a otro.

1.4.4. Métodos de inicialización para k-medias y k-medianas

Hemos visto que una de las características tanto del k-medias como del k-medianas es su dependencia de la inicialización, ya que una adecuada selección de las semillas de los agrupamientos puede llevar a un agrupamiento final con un mejor valor del criterio de optimización.

Este es un problema que se ha abordado con diversos heurísticos. El método trivial consiste en escoger k muestras aleatoriamente del conjunto y usarlas como representantes iniciales [15]. Evidentemente, no se puede garantizar que esta inicialización lleve a buenos resultados, ya que existe la posibilidad de tomar muestras poco representativas de la distribución real de los datos.

Una alternativa es la selección supervisada de muestras como representantes [34]. En este tipo de selección se escogen muestras que son claramente representativas del conjunto de datos a estudiar, con lo cual se obtienen k representantes iniciales que tienen una calidad garantizada. El problema es que este tipo de inicialización es principalmente manual, es decir, no es posible llevarla a cabo con métodos completamente automáticos.

Un método usado habitualmente es el conocido como método voraz [60]. Este es un método genérico subóptimo para resolver el problema de escoger k representantes del conjunto de muestras disponible que optimicen una cierta función objetivo. Si la función objetivo a optimizar es z y el conjunto de muestras es P, la obtención del conjunto de representantes Q se hace en sucesivos pasos

que obtienen los conjuntos Q^t, con $t = 0, \ldots, k$, siendo finalmente $Q = Q^t$. En el caso de minimización, la obtención de los distintos Q^t se realiza siguiendo la fórmula:

$$Q^t = \begin{cases} \emptyset & t = 0 \\ Q^{t-1} \cup \{q_t\} & t > 0 \quad \text{con } q_t = \underset{p \in P - Q^{t-1}}{\operatorname{argmin}} z(Q^{t-1} \cup \{p\}) \end{cases} \tag{1.9}$$

Como vemos, el proceso consiste en ir añadiendo cada vez al conjunto previo la muestra aún no escogida que minimice la función objetivo, hasta que se obtienen las k muestras necesarias.

Particularizando para un conjunto de datos finito en el que está definida una medida de distancia, definimos el conjunto total de muestras T y el conjunto de representantes S, con $|S| = k$. La función objetivo a minimizar se define así como:

$$z(S) = \sum_{t \in T} \min_{s \in S} d(s, t) \tag{1.10}$$

donde d es la distancia definida entre las muestras. Como vemos, se trata de escoger un conjunto S de k muestras y luego asignar cada muestra $t \in T$ al elemento de $s \in S$ más cercano. La suma total de distancias entre t y s es el criterio a optimizar.

Así, el método voraz de optimización para este caso quedaría descrito mediante la fórmula:

$$S^t = \begin{cases} \emptyset & t = 0 \\ S^{t-1} \cup \{s_t\} & t > 0 \quad \text{con } s_t = \underset{r \in T - S^{t-1}}{\operatorname{argmin}} \sum_{u \in T} \min_{v \in S^{t-1} \cup \{r\}} d(u, v) \end{cases} \tag{1.11}$$

donde a cada iteración se añade la muestra $r \in T$ que aún no está en S tal que al añadirla a S conseguimos el valor mínimo para la función descrita en (1.10).

Otra de las alternativas de inicialización es el método de optimización conocido como método de intercambio [60]. En dicho método, se tiene un conjunto de k representantes iniciales, que puede haberse escogido usando alguno de los métodos descritos previamente (aleatorio, voraz,...). Partiendo de esos k representantes, el método trata de hallar en cada paso un par (r, m), r miembro del actual conjunto de representantes y m muestra no incluída en dicho conjunto, de manera que al excluir r de, e incluir m en el conjunto de representantes se optimice la función objetivo.

Es decir, si tenemos el conjunto de muestras T y un cierto conjunto de representantes Q, la actualización de Q a Q' para la minimizar la función objetivo z se realiza siguiendo la fórmula:

$$Q' = Q - \{r\} \cup \{m\} \quad \text{con } (r, m) = \underset{(q,t) \in Q \times T - Q}{\operatorname{argmin}} z(Q - \{q\} \cup \{t\}) \tag{1.12}$$

Es decir, se escoge el par de muestras tal que al eliminar una y añadir la otra se mejore lo más posible la función objetivo. Este proceso se haría hasta que cualquier cambio de Q a Q' diera un incremento de la función objetivo, es decir, no existe ningún posible intercambio que permita obtener un valor menor de z.

En el caso de espacios con una medida de distancia definida, la función objetivo es la misma que para el método voraz (la descrita en la Ecuación (1.10)), y por tanto describiríamos el método de optimización por intercambio en dicho caso mediante la siguiente fórmula:

$$S' = S - \{s'\} \cup \{s\} \quad \text{con } (s', s) = \underset{(r,t)\in S\times T-S}{\operatorname{argmin}} \sum_{u\in T} \underset{v\in S-\{r\}\cup\{t\}}{\text{mín}} d(u,v) \quad (1.13)$$

Es decir, se realiza el intercambio entre las dos muestras tal que se dé la menor distancia acumulada, tomando la distancia de las muestras no tomadas siempre a su representante más cercano (incluyendo y excluyendo las muestras que se prueban en el paso actual).

Como hemos comentado previamente, para obtener las k muestras iniciales necesarias para realizar los intercambios puede usarse alguno de los métodos descritos previamente. En el caso de usar el método voraz para obtener esas k muestras, dicha aproximación es la conocida como la voraz-intercambio [33]. Otra variación de esta técnica de optimización es la generalización de la misma a varias muestras a la vez; es decir, en vez de limitar el intercambio a una muestra en cada iteración, se generaliza a intercambiar k muestras (dicho k no tiene relación con el número final de representantes) en cada iteración. Dicha variante se conoce como k-intercambio [60].

Otro hecho que también se debe destacar es que el algoritmo k-medianas es en realidad una simplificación de este método de intercambio. La restricción que aporta k-medianas es que los prototipos a intercambiar pertenecen al mismo agrupamiento (ya que uno es el representante actual, que ha generado dicho agrupamiento, y el otro debe ser la mediana, que por definición pertenece a dicho agrupamiento), y no al conjunto total de muestras disponibles.

Hasta ahora, los métodos nombrados (voraz, intercambio y voraz-intercambio) son en realidad técnicas genéricas de optimización que pueden aplicarse para obtener las k muestras iniciales necesarias para aplicar k-medias o k-medianas. Sin embargo, existe otro método específico que se ha usado con frecuencia como inicialización para k-medias, presentado en su día por Katsavounidis en [36]. Este método se puede ver como un método voraz aproximado, con la ventaja de ser mucho más eficiente [34]. Basándose en dicho método, se propuso el algoritmo maxmin [34] como técnica de inicialización para el método k-medianas.

El algoritmo maxmin parte de una muestra inicial arbitraria, que va a ser la generadora de la secuencia de representantes. Para llegar a obtener los k representantes, en cada paso va a tomar la muestra más alejada del conjunto de representantes actuales. Así, dado un conjunto de muestras P entre las cuales existe una medida de distancia d, en la iteración t tendremos el conjunto de representantes Q^t formado según la siguiente fórmula:

$$Q^t = \begin{cases} \text{muestra\_arbitr}(P) & t = 0 \\ Q^{t-1} \cup \{q_t\} & t > 0 \quad \text{con } q_t = \underset{p\in P-Q^{t-1}}{\operatorname{argmax}} \underset{q\in Q^{t-1}}{\text{mín}} d(p,q) \end{cases} \quad (1.14)$$

Como vemos, a diferencia de los métodos previos aquí se hace uso de una medida de distancia d y no de una función objetivo general. Por tanto, esto implica que

Entrada: P conjunto de datos, k número de agrupamientos a obtener
Salida: $Q \subset P$ conjunto de representantes de los agrupamientos
Inicio
$Q = \emptyset$
q = muestra_arbitr(P)
Para $t = 1, \ldots, k$ /* Hasta alcanzar el número de agrupamientos */
  $Q = Q \cup \{q\}$ /* Añadimos la seleccionada anteriormente */
  $maxmin = 0$
  ParaTodo $u \in P - Q$ /* Para todas las muestras no tomadas */
    $duv = \infty$
    ParaTodo $v \in Q$
      /* Se toma la distancia al representante más cercano */
      Si $d(u, v) < duv$ **Entonces** $duv = d(u, v)$ **FSi**
    FParaTodo
    Si $duv > maxmin$ **Entonces** /* Si está más lejos, se toma */
      $q = u$
      $maxmin = duv$
    FSi
  FParaTodo
FPara
Devolver Q

Figura 1.2: Esquema del algoritmo maxmin

sólo puede aplicarse el algoritmo maxmin para espacios métricos. El esquema general de este algoritmo se presenta en la Figura 1.2, donde d indica la función de distancia usada. Puede verse que a cada iteración prueba todas las muestras que aún no están en Q (conjunto de representantes) y calcula la distancia de esa muestra al representante más cercano. Finalmente se añade la muestra que presenta la mayor distancia a su representante más cercano. El proceso se repite hasta que se tienen las k muestras que serán los representantes de los k agrupamientos.

En [34] se hace un estudio de la eficacia de varios algoritmos para la selección inicial de prototipos para aplicar posteriormente el método k-medianas. En dicho estudio se muestra que el algoritmo maxmin consigue mejores resultados respecto a la elección aleatoria y equiparables a los que obtiene el método voraz-intercambio, siendo con respecto a este último método más eficiente computacionalmente.

1.5. Medidas de disimilitud entre cadenas

Como hemos visto en la Sección 1.1, para la aplicación de técnicas no paramétricas de clasificación es necesaria la definición de una medida de disimilitud entre los objetos. Dicha medida, debe cumplir una serie de condiciones

para garantizar la corrección de los resultados de clasificación. Así, dados dos objetos codificados cualesquiera o_1 y o_2, la medida de disimilitud d debe cumplir las siguientes condiciones [82]:

- $d(o_1, o_2) \geq 0$
- $d(o_1, o_1) = 0$
- $d(o_1, o_2) = d(o_2, o_1)$

Si además de estas tres condiciones se cumple la condición de desigualdad triangular:

$$d(o_1, o_2) + d(o_2, o_3) \geq d(o_1, o_3) \qquad \forall o_1, o_2, o_3$$

entonces se tiene que d es una métrica, y se habla de ella como una medida de distancia.

Cuando los objetos se representan mediante vectores, existen gran cantidad de distancias [82]. En el caso de la representación por cadenas también ha habido diversas propuestas de medidas de disimilitud entre cadenas [72], algunas de las cuales vamos a detallar a continuación.

1.5.1. Distancia de edición

El problema de hallar la disimilitud entre dos cadenas sobre un determinado alfabeto puede verse como una cantidad numérica que determine el "esfuerzo" necesario para convertir una cadena en otra. Así, cuanto menor sea el esfuerzo necesario para esa transformación, más próximas estarán estas cadenas.

A la hora de definir cómo se realiza dicha transformación, se pueden definir multitud de operadores básicos que la realicen. Históricamente, se han definido como operadores básicos los de sustitución, inserción y borrado. Dadas las cadenas s y t a comparar, de longitudes $|s|$ y $|t|$, $s = s_1 \cdot s_2 \cdots s_{|s|}$ y $t = t_1 \cdot t_2 \cdots t_{|t|}$, el operador de sustitución empareja un símbolo de s con un símbolo de t. El operador de inserción hace que un símbolo en la cadena s no esté emparejado con ningún símbolo de t. El operador de borrado hace que un símbolo de t no se empareje con ninguno de s (es, por tanto, el dual de inserción). En la Figura 1.3 podemos ver la reprensentación gráfica habitual de las operaciones de edición, donde λ representa la cadena vacía (de manera que (λ, b) representa la inserción y (a, λ) representa el borrado).

Es evidente que otras muchas operaciones elementales de edición pueden ser definidas. Una de ellas puede ser la transposición [72], en la cual se consideran secuencias de dos símbolos que intercambian su posición (es decir, la secuencia ab se cambiaría por ba, con $a, b \in \Sigma$). Aunque dicha operación equivale a dos sustituciones, es posible que en la tarea con la que se trabaje tenga sentido que el "esfuerzo" total de dicha operación sea menor que la suma del "esfuerzo" de ambas sustituciones. De todas maneras, nuestro trabajo se centrará en las tres operaciones de edición clásicas.

Siguiendo la nomenclatura propuesta por Wagner y Fischer [80], se definen las operaciones de edición elementales por pares $(a, b) \neq (\lambda, \lambda)$, donde a y b son

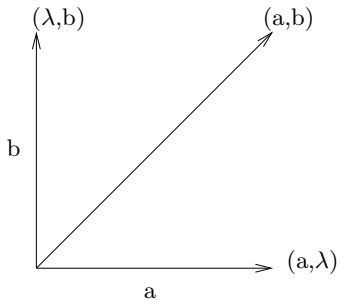


Figura 1.3: Representación gráfica de las operaciones de edición.

cadenas de, como máximo, un símbolo. Así, una sustitución viene representada por (a, b), un borrado por (a, λ) y una inserción por (λ, b). Para transformar una cadena $s \in \Sigma^*$ en una cadena $t \in \Sigma^*$ se sigue una secuencia de operaciones de edición elementales $\mathcal{T}$, llamada transformación.

Supongamos por ejemplo que tenemos definidas las cadenas $s = abbca$ y $t = bacaba$, definidas sobre el alfabeto $\Sigma = \{a, b, c\}$. Una posible transformación de s en t vendría definida por la siguiente secuencia:

$$s = abbca \Rightarrow^{(a,\lambda)} bbca \Rightarrow^{(b,b)} bbca \Rightarrow^{(\lambda,a)} babca \Rightarrow^{(b,\lambda)} baca$$

$$baca \Rightarrow^{(c,c)} baca \Rightarrow^{(\lambda,a)} bacaa \Rightarrow^{(a,b)} bacab \Rightarrow^{(\lambda,a)} bacaba = t$$

A cada una de estas operaciones se le asocia un peso γ determinado (que computaría el "esfuerzo" de la operación). En el caso de la distancia de edición clásica, estos pesos quedan definidos como:

$$\gamma(a, b) = \begin{cases} 1 & a \neq b \\ 0 & a = b \end{cases} \qquad \gamma(a, \lambda) = 1 \qquad \gamma(\lambda, b) = 1 \qquad \forall a, b \in \Sigma$$

Sin embargo, es posible asociar cualquier peso a estas operaciones de edición, algo que resulta más que útil de cara aplicaciones reales. Esto se debe a que el "esfuerzo" asociado a sustituir dos símbolos cualquiera no tiene por qué ser siempre el mismo. Por ejemplo, si las componentes elementales que representan dichos símbolos son muy distintas resulta razonable pensar que la sustitución de una por otra requiera de más "esfuerzo" que la sustitución entre dos símbolos que representen componentes más semejantes. Igualmente puede suceder en los casos de inserción y borrado (insertar o borrar cierto símbolo puede ser más costoso que hacerlo para otro).

Así, se puede considerar la posibilidad de que los pesos asociados a cada operación de edición elemental dependan de los símbolos comparados, y no que sean unitarios tal como se define en la distancia de edición clásica. También sería posible hacer que dichos pesos dependan de otros factores aparte de los propios símbolos implicados, como por ejemplo, del contexto (símbolos adyacentes en las cadenas implicadas en la transformación), aunque generalmente, tal y como

γ	a	b	c	λ
a	0	1	2	1
b	1	0	1	2
c	2	1	0	1
λ	1	2	1	-

Figura 1.4: Matriz de pesos para $\Sigma = \{a, b, c\}$

propusieron Wagner y Fischer [80], únicamente se tienen en cuenta los propios símbolos implicados. A este tipo de distancia que se basa en pesos no unitarios se la conoce generalmente como distancia de edición ponderada, y, evidentemente, la distancia de edición clásica es un caso particular de la misma.

Normalmente, el peso asociado a las operaciones de edición se representa mediante la llamada matriz de pesos, que se define como una matriz $\{\Sigma \cup \{\lambda\} \times \Sigma \cup \{\lambda\}\} - \{(\lambda, \lambda)\} \rightarrow \mathbb{R}$. En dicha matriz, las filas y columnas quedan indizadas por el alfabeto Σ sobre el cual se construyen las cadenas y la cadena vacía λ. Así, si llamamos M a la matriz de pesos, tendremos que $\gamma(a, b) = M(a, b)$, para $a, b \in \Sigma \cup \{\lambda\}$. Un ejemplo de matriz de pesos, para el alfabeto $\Sigma = \{a, b, c\}$, puede verse en la Figura 1.4.

Se puede extender la definición de γ sobre transformaciones completas, de manera que $\gamma(\mathcal{T})$ es la suma de todos los $\gamma(a, b)$ para todas las ediciones elementales (a, b) de la transformación $\mathcal{T}$. Por tanto, para el ejemplo de transformación que hemos mostrado arriba, suponiendo que usamos la matriz de pesos de la Figura 1.4, tendríamos:

$$\gamma(\mathcal{T}) = \gamma(a, \lambda) + \gamma(b, b) + \gamma(\lambda, a) + \gamma(b, \lambda) + \gamma(c, c) + \gamma(\lambda, a) + \gamma(a, b) + \gamma(\lambda, a) =$$

$$1 + 0 + 1 + 2 + 0 + 1 + 1 + 1 = 7$$

Evidentemente, a la hora de calcular la distancia de edición entre cadenas se debe considerar la mejor transformación posible de una cadena a otra, entendiendo por mejor la que necesite un menor esfuerzo. Así pues, la definición de distancia de edición entre dos cadenas sobre el alfabeto Σ, s y t, se define por:

$$d(s, t) = \min_{\mathcal{T} \in T(s,t)} \{\gamma(\mathcal{T})\} \tag{1.15}$$

donde $T(s, t)$ es el conjunto de todas las posibles transformaciones de s a t. En el caso de nuestro ejemplo, la transformación óptima sería la siguiente:

$$s = abbca \Rightarrow^{(a,\lambda)} bbca \Rightarrow^{(b,b)} bbca \Rightarrow^{(b,a)} baca \Rightarrow^{(c,c)} baca$$

$$\Rightarrow^{(a,a)} baca \Rightarrow^{(\lambda,b)} bacab \Rightarrow^{(\lambda,a)} bacaba = t$$

cuyo peso sería 4.

Esta definición de distancia de edición en función de las transformaciones cumple la desigualdad triangular independientemente del peso asociado a cada una de las posibles operaciones elementales. Para que sea una métrica es necesario exigir que se cumplan además las condiciones [53]:

- $\gamma(a,a) = 0 \qquad \forall a \in \Sigma$
- $\gamma(a,b) > 0 \qquad \forall a \neq b, \quad a,b \in \Sigma \cup \{\lambda\}$
- $\gamma(a,b) = \gamma(b,a) \qquad \forall a,b \in \Sigma \cup \{\lambda\}$

Esto implica que la definición clásica pueda ser considerada propiamente una distancia, mientras que para la definición ponderada es necesario imponer dichas restricciones a la hora de elegir los pesos para que pueda ser considerada una distancia propiamente dicha. Por tanto, la matriz de pesos será generalmente simétrica, con ceros en su diagonal principal y el resto de elementos mayores que cero.

Otra condición que suele imponerse para la matriz de pesos es la de evitar que una sustitución sea menos favorable que la inserción-borrado de los símbolos implicados. Es decir, se suele diseñar la matriz de pesos de manera que $\gamma(a,b) < \gamma(a,\lambda) + \gamma(\lambda,b)$. Como regla general, se suele restringir a que γ cumpla a cada par de símbolos la desigualdad triangular, es decir, $\gamma(a,b) \leq \gamma(a,c) + \gamma(c,b)$ para $a \neq b \neq c$, con $a,b,c \in \Sigma \cup \{\lambda\}$.

Si dicha propiedad se cumple, tenemos la ventaja de que es posible el cálculo de la distancia de edición usando la representación por secuencias de edición [80]. Una secuencia de edición p para las cadenas s y t es una secuencia de pares de enteros (i_k, j_k), con $0 \leq k \leq m$, donde m es la longitud de la secuencia, tal que se cumple:

$$\begin{aligned} 0 \leq i_k \leq |s|, 0 \leq j_k \leq |t|, (i_0, j_0) = (0,0), (i_m, j_m) = (|s|, |t|) \\ 0 \leq i_k - i_{k-1} \leq 1, 0 \leq j_k - j_{k-1} \leq 1, \forall k \leq 1 \\ i_k - i_{k-1} + j_k - j_{k-1} \geq 1 \end{aligned} \tag{1.16}$$

es decir, una secuencia de pares de puntos tales que dicho par representa los símbolos por los que se va haciendo la transformación en la primera y en la segunda cadena, respectivamente. Así, por ejemplo, tomando de nuevo las cadenas $s = abbca$ y $t = bacaba$, una posible secuencia de edición sería:

$$s = abbca \Rightarrow (0,0) - (0,1) - (1,2) - (2,2) - (2,3) -$$

$$-(3,4) - (4,4) - (5,5) - (6,5) \Rightarrow bacaba = t$$

Esta forma de representación es muy poco intuitiva. Por tanto, es usual representar dicha secuencia de edición como un camino dentro de un grafo llamado grafo de edición (representable como una matriz), que indica las sucesivas ediciones que se van realizando en la secuencia de edición. Para el ejemplo que estamos siguiendo, dicha representación coincidiría con lo mostrado en la Figura 1.5.

Dos puntos sucesivos de la secuencia de edición corresponden a una operación de edición elemental, de manera que para asociar el peso de una secuencia de edición p para s y t se sigue la fórmula:

$$\gamma(p) = \sum_{k=1}^{m} \gamma(s_{i_{k-1}+1 \ldots i_k}, t_{j_{k-1}+1 \ldots j_k}) \tag{1.17}$$

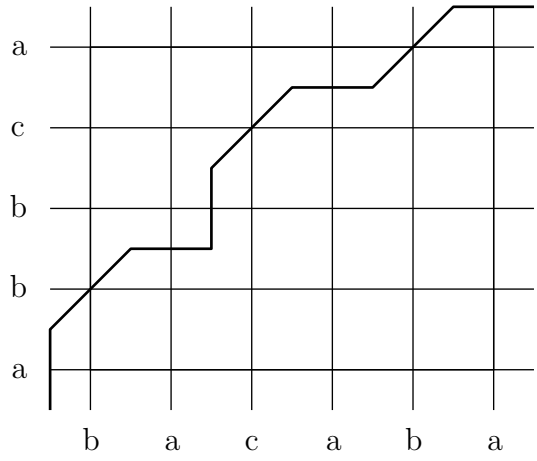


Figura 1.5: Representación habitual de una secuencia de edición

donde $s_{i...j}$ indica la subcadena de s desde la posición i a la posición j de la misma si $i \leq j$, o λ cuando $i > j$.

Así, la distancia de edición se puede definir en función de las secuencias de edición, de manera que:

$$d(s,t) = \min_{p \in P(s,t)} \{\gamma(p)\} \tag{1.18}$$

donde $P(s,t)$ es el conjunto de todas las secuencias de edición posibles para las cadenas s y t.

Para calcular esta distancia, el planteamiento es pensar que a cualquier punto (i,j) de la secuencia de edición, que pertenece al conjunto $\{0,\ldots,|s|\} \times \{0,\ldots,|t|\}$, se puede llegar desde los puntos $(i-1,j-1)$, $(i-1,j)$ y $(i,j-1)$, (excepto en los casos límite en los que $i=0$ o $j=0$) usando respectivamente sustituciones (cambia s_i por t_j), inserciones (inserta s_i) y borrados (borra t_j). Dicho punto se corresponde con el fin de las secuencias de edición entre las subcadenas $s_{1...i}$ y $t_{1...j}$. Por tanto, el cálculo de la distancia puede hacerse con la función recursiva:

$$\begin{aligned} d(s_{1...i}, t_{1...j}) = \min\{&d(s_{1...i-1}, t_{1...j} + \gamma(s_i, \lambda), \\ &d(s_{1...i-1}, t_{1...j-1} + \gamma(s_i, t_j), \\ &d(s_{1...i}, t_{1...j-1} + \gamma(\lambda, t_j)\} \end{aligned} \tag{1.19}$$

que para el caso de las cadenas completas, evidentemente, se corresponde con hacer la recurrencia desde el punto $(|s|,|t|)$.

Esta función recursiva puede calcularse usando un algoritmo de Programación Dinámica propuesto en [80], cuyo esquema básico vemos en la Figura 1.6. Su coste computacional es de $O(|s||t|)$, que en el caso espacial puede reducirse a $O(\min\{|s|,|t|\})$ si no es necesario recuperar la secuencia de edición.

Entrada: $s, t \in \Sigma^*$, γ función de pesos
Salida: $D \in \mathbb{R}^{(|s|+1)\times(|t|+1)}$ matriz de Programación Dinámica con las distancias de edición, tal que $d(s,t) = D(|s|,|t|)$
Inicio
$D(0,0) = 0$
Para $i = 1, \ldots, |s|$ /* Inicialización para borrados */
$D(i,0) = D(i-1,0) + \gamma(s_i, \lambda)$
FPara
Para $j = 1, \ldots, |t|$ /* Inicialización para inserciones */
$D(0,j) = D(0,j-1) + \gamma(\lambda, t_j)$
FPara
Para $i = 1, \ldots, |s|$ /* Para cada punto del grafo */
Para $j = 1, \ldots, |t|$
$D(i,j) = \text{mín}\{D(i-1,j) + \gamma(s_i,\lambda), D(i,j-1) + \gamma(\lambda,t_j),$
$D(i-1,j-1) + \gamma(s_i,t_j)\}$
FPara
FPara
Devolver D

Figura 1.6: Algoritmo de Programación Dinámica de Wagner y Fischer que permite el cálculo de la distancia de edición clásica entre dos cadenas s y t

1.5.2. Distancia de edición normalizada

Uno de los problemas que presenta la distancia de edición tal y como se ha definido hasta ahora es su sensibilidad a la longitud de las cadenas. Es decir, a mayor longitud de las cadenas comparadas, la distancia de edición tiende a ser también mayor. Así, por ejemplo, sobre $\Sigma = \{a, b\}$, las cadenas *aa* y *ab* presentarían una distancia de edición clásica de valor 1, mientras que *aaaaaaabaa* y *aabaaaaaaa* presentan una distancia de edición clásica de valor 2. Sin embargo, en el primer caso difieren en el 50 % de los símbolos mientras que en el segundo sólo difieren en el 20 %. Por tanto, la distancia de edición no parece adecuada cuando se trata de obtener comparaciones entre cadenas de diversa longitud.

Así, la idea inmediata que acude es la de definir una distancia entre cadenas que resulte independiente de la longitud de las cadenas a comparar, midiendo realmente el grado de disimilitud entre ellas y no la disimilitud en valores absolutos. Siguiendo estas directrices, se propuso la distancia de edición normalizada [53].

Dada una secuencia de edición p, se define su peso como $W(p) = \gamma(p)$, siendo la definición de γ la vista en (1.17). De la misma manera, se define la longitud de la secuencia de edición p, $L(p)$, que es el número de operaciones de edición elementales de p. Así, se define la distancia de edición normalizada entre dos cadenas s y t sobre Σ como:

$$d_N(s,t) = \min_{p \in P(s,t)} \frac{W(p)}{L(p)} \tag{1.20}$$

Entrada: p^* secuencia de edición arbitraria para s y t
Salida: d distancia de edición normalizada entre s y t, p^* secuencia de edición asociada
Inicio
$d^* = \frac{W(p^*)}{L(p^*)}$ /* Valor inicial */
Repetir
$d' = d^*$
$p^* = \text{argmin}_{p \in P(s,t)}(W(p) - d'L(p))$ /* Aplica prog. fraccional */
$d^* = \frac{W(p^*)}{L(p^*)}$
Hasta $d^* = d'$ /* Hasta la convergencia */
Devolver d^*, p^*

Figura 1.7: Algoritmo para el cálculo de la distancia de edición normalizada y su secuencia de edición asociada

donde $P(s,t)$ es nuevamente el conjunto de todas las posibles secuencias de edición entre s y t.

La computación de la distancia de edición normalizada se puede hacer usando como matriz de pesos γ cualquiera de las que se pueda usar para el cálculo de la distancia de edición ponderada. El algoritmo que se usa para su cálculo se basa en técnicas de programación fraccional y fue presentado en [79]. Su esquema básico se puede ver en la Figura 1.7. Como se ve, el proceso consiste en partir de una secuencia de edición inicial p^* (por ejemplo, la obtenida aplicando la distancia de edición no normalizada), de la cual se conoce su peso $W(p^*)$ y su longitud $L(p^*)$. Con esto se tiene un valor inicial de la distancia de edición normalizada d', que se usa para calcular una nueva secuencia de edición p^* usando la minimización sobre la regla dada por programación fraccional $(W(p) - d'L(p))$. Este proceso se repite hasta que no haya diferencias entre la distancia computada en una iteración y la computada en la iteración previa. El coste de dicho algoritmo es, en término promedio, del mismo orden que el cálculo de la distancia de edición no normalizada [79].

Esta medida de disimilitud presenta la característica de no ser formalmente una métrica (es decir, una distancia), ya que la desigualdad triangular no siempre se cumple. Se pueden encontrar ejemplos que muestren este hecho para ciertas matrices de pesos en las cuales una inserción o borrado tiene un peso mucho menor que cualquier otro peso de la matriz [53]. Sin embargo, en la práctica estas situaciones no suelen darse y la distancia de edición normalizada suele cumplir con todas las condiciones necesarias para ser una métrica.

1.6. El problema de la cadena media

La idea intuitiva de cadena media se corresponde con la equivalente a la del vector media para un conjunto de datos representado mediante cadenas. La definición inicial de cadena media se debe a Kohonen [38], en cuyo trabajo se

define por primera vez el concepto y se ofrecen resultados[2].

1.6.1. Definición de cadena media

Dado un alfabeto Σ, definimos su monoide libre asociado, notado por Σ^*, como el conjunto de todas aquellas cadenas de longitud finita formadas con los símbolos de Σ. Sea d una medida de disimilitud o distancia definida sobre Σ^*. Dado $S = \{s^1, s^2, \ldots, s^n\} \subset \Sigma^*$, la cadena media de S, m_S viene definida, según lo propuesto por Kohonen, por:

$$m_S = \underset{s \in \Sigma^*}{\operatorname{argmin}} \sum_{i=1}^{n} d(s, s^i) \tag{1.21}$$

Como vemos, dicha definición es muy semejante a la del vector media, pues busca dentro del espacio de representación (en este caso, el monoide libre sobre el alfabeto Σ) aquel elemento que minimice la suma de distancias a todas las muestras de S. En este caso no es posible caracterizar la cadena media como se hizo para el vector media en (1.6), ya que no es posible efectuar la derivación de la función dada en (1.21) para encontrar la solución analítica al problema de minimización. Es más, está demostrado que el problema de obtener la cadena media es NP-Duro [11].

Respecto a la medida de disimilitud d, no está definida; es decir, cualquier medida de disimilitud que se pueda definir sobre Σ^* es plausible para la definición de la cadena media. Esto se debe a que es la propia medida de disimilitud la que genera el espacio métrico (siempre y cuando cumpla las propiedades necesarias descritas en la Sección 1.5). El concepto de media sólo tiene sentido en un espacio métrico definido, es decir, la media para un cierto conjunto de muestras en un espacio métrico no se corresponde con la de las mismas muestras representadas en otro espacio métrico distinto. Por tanto, la obtención de la media con cualquier medida de disimilitud d es correcta al estar las muestras representadas en el espacio métrico definido por d.

Así, cualquiera de las medidas de disimilitud propuestas en la Sección 1.5 puede usarse para esta definición. Igualmente, otras medidas (como las descritas en [38]) son aplicables, aunque algunas medidas sólo son aplicables en casos restringidos (por ejemplo, la distancia de Hamming sólo es aplicable cuando se trata de cadenas de la misma longitud).

Evidentemente, el escoger una medida de disimilitud u otra depende de lo adecuada que resulte tal medida para la aplicación correspondiente, ya que el fin último de obtener la cadena media es su aplicación en tareas de RF. Si para la tarea final es más adecuada cierta medida de disimilitud, lo más conveniente es usar esa propia medida en la definición de cadena media.

[2]En dicho artículo, se nota la cadena media mediante el término anglosajón *(generalized) median string*, lo cual puede llevar a confusión, pues la traducción de *median* es "mediana".

Entrada: $S = \{s^1, s^2, \ldots, s^n\}$
Salida: $sm \in S$ tal que sm es la cadena mediana de S
$sm = s_1$
mejordist = ∞
Para i = 1,2,...,n /* Para todas las cadenas */
    dist=0.0
    Para j = 1,2,...,n /* Calcula distancia acumulada */
        dist=dist+$d(s^i, s^j)$
    FPara
    Si dist≤mejordist **Entonces** /* Si es mejor, la almacena */
        mejordist = dist
        $sm = s^i$
    FSi
FPara
Devolver sm

Figura 1.8: Algoritmo para la obtención de la cadena mediana.

1.6.2. La cadena mediana

La cadena mediana (conocida también por su término anglosajón *set median*) es una de las aproximaciones clásicas y más usuales a la cadena media exacta. Su principal característica es que su cálculo presenta un coste polinómico y que resulta extremadamente sencillo definir el algoritmo para extraerla. En cambio, su calidad como prototipo es un tanto discutible.

Sea un alfabeto Σ y sea su monoide libre asociado Σ^*. Sea d una medida de disimilitud definida sobre Σ^*. Dado $S = \{s^1, s^2, \ldots, s^n\} \subset \Sigma^*$, la cadena mediana de S, sm_S, viene definida por:

$$sm_S = \underset{s \in S}{\operatorname{argmin}} \sum_{i=1}^{n} d(s, s^i) \tag{1.22}$$

Dicha definición es muy semejante a la definición de cadena media dada en (1.21), pero aporta una restricción fundamental: el espacio de búsqueda pasa de ser Σ^* a ser S. Esto tiene una importancia determinante desde el punto de vista computacional, pues ahora la obtención de dicha cadena tiene un coste polinómico.

El algoritmo para la obtención de la cadena mediana es el que se presenta en la Figura 1.8. Este algoritmo calcula para cada cadena del conjunto la distancia acumulada al resto de cadenas, y va quedándose con aquella cadena que presente menor distancia acumulada. Como puede verse claramente, el coste temporal de dicho algoritmo es $O(n^2 \cdot C)$, donde C es la complejidad temporal del cálculo de $d(s, t)$. En general, para las distancias de edición que se suelen usar (las descritas en la Sección 1.5), este coste temporal es del orden de $|s| \cdot |t|$, con lo cual se puede concluir de manera general que el algoritmo de obtención de la cadena mediana presenta una complejidad temporal del orden $O(n^2 \cdot l^2)$, donde $l = \max_{s \in S} |s|$.

El uso de la cadena mediana como aproximación a la cadena media es, posiblemente, el más extendido y antiguo. Ya en el artículo en el que Kohonen definía la cadena media [38], también se daba la definición de cadena mediana y se aplicaba en la obtención de la forma correcta sobre conjuntos de cadenas distorsionadas; es decir, dada una palabra, se obtenía un conjunto de palabras distorsionadas a partir de la misma, y de este conjunto se obtenía la media y la mediana con el objetivo de ver su semejanza con la palabra original. En este mismo trabajo, muestra como la cadena mediana presenta una menor calidad con respecto a la cadena media tanto en identificación del patrón correcto como en clasificación.

Estos resultados ya expuestos en dicho trabajo llevan a considerar que el uso de la cadena mediana como prototipo puede no resultar adecuado, y que aproximaciones más plausibles a la cadena media deberían ser encontradas y utilizadas.

1.7. Objetivos de la tesis

En esta tesis, dados los preliminares expuestos en el presente capítulo, el objetivo principal es la resolución del problema de la cadena media y su aplicación a tareas de clasificación en Reconocimiento de Formas. Debido a las características del problema, debemos limitarnos a obtener soluciones aproximadas del mismo, y aplicar dichas soluciones a aquellas tareas de Reconocimiento de Formas en las que los objetos se representan mediante cadenas. Así pues, los objetivos a cubrir en esta tesis son:

- Definir algoritmos que proporcionen buenas aproximaciones a la cadena media.
- Aplicar las aproximaciones a la cadena media obtenidas en tareas de clasificación mediante clasificadores k-NN.
- Estudiar técnicas que permitan reducir el coste computacional de obtener dichas aproximaciones, evaluando su impacto en la calidad de la clasificación.
- Definir un algoritmo que obtenga una solución exacta al problema de la cadena media.
- Comparar la calidad en clasificación de la cadena media exacta con las diversas aproximaciones propuestas.
- Validar los resultados de clasificación experimentando con diversos corpora, reales y sintéticos.
- Proponer métodos de obtención de agrupamientos basados en las aproximaciones propuestas.

- Adaptar los métodos de obtención de cadenas medias aproximadas a cadenas cíclicas.
- Comparar los resultados de clasificación obtenidos por k-NN con los resultados obtenidos con otros modelos y métodos de clasificación.

1.8. Resumen

En este capítulo hemos abordado los conceptos básicos a los que nos iremos refiriendo a lo largo de esta tesis. En primer lugar, hemos dado una descripción global del Reconocimiento de Formas y sus diferentes subdivisiones (problemas de clasificación o interpretación, Reconocimiento Geométrico o Sintáctico, métodos paramétricos o no paramétricos) para poder situarnos claramente dentro del ámbito de problemas de clasificación mediante métodos no paramétricos en el caso de representar los objetos mediante cadenas. El método no paramétrico de clasificación adoptado es la regla k-NN, y hemos mostrado su relación con el método paramétrico óptimo (la regla de Bayes). El uso de estos métodos no paramétricos nos ha encaminado hacia dos problemas clave: la extracción de prototipos adecuados y las medidas de disimilitud a usar cuando los objetos a comparar son cadenas. Respecto al primer problema, hemos dado un somero repaso a las técnicas clásicas de extracción de prototipos, además de plantear el uso de la media de una clase o subclase como prototipo de la misma. El hecho de tener que obtener diversos prototipos para una clase nos ha llevado a considerar métodos de obtención de agrupamientos, que nos permiten particionar una clase en varias subclases, pudiendo obtener de cada subclase los prototipos adecuados. Así pues, se ha incluído una sección dedicada a comentar a nivel introductorio los métodos de agrupamiento. En cuanto al problema de obtener una medida de disimilitud entre cadenas, hemos comentado una serie de medidas de distancia entre cadenas, indicando las propiedades básicas de cada una de ellas. Se ha definido el concepto de media para conjuntos de cadenas, la llamada cadena media, algo necesario para aplicar la extracción de prototipos por medias para espacios de cadenas, y se ha mostrado las dificultades que encierra su cálculo, definiendo también el concepto de cadena mediana que podría usarse como prototipo alternativo. Finalmente, se han enumerado los objetivos a cubrir en esta tesis.

Capítulo 2

Aproximación voraz a la cadena media

En este capítulo vamos a describir un método para aproximar la cadena media basado en una técnica voraz. La aplicación de dicha técnica voraz para obtener prototipos y los resultados de clasificación obtenidos con los mismos serán comparados con los resultados que se obtienen mediante el uso de la cadena mediana.

2.1. La aproximación constructiva voraz

En general, cuando se busca la solución de un problema de optimización, es posible aplicar un algoritmo heurístico voraz que nos dé una solución aproximada razonablemente buena para dicho problema. Esta solución, según las necesidades del problema que se esté resolviendo, puede tomarse como solución final o como solución inicial para un proceso posterior que busque la solución exacta. Básicamente, si tenemos un conjunto de elementos básicos posibles, este método va construyendo, partiendo inicialmente del conjunto vacío, una secuencia de subconjuntos de elementos básicos que nos optimicen la función objetivo. Si definimos la función objetivo z sobre un subconjunto Q del conjunto de elementos básicos $T = \{1, 2, \ldots, n\}$, el algoritmo voraz general es el presentado en la Figura 2.1 para el caso de la minimización [60]. En dicho algoritmo se parte de una solución vacía y a cada iteración se toma aquel elemento de T que unido al conjunto previo de soluciones da mejor valor de la función objetivo. El proceso continúa hasta obtener una solución que presenta peor valor de la función objetivo que la solución previa.

Este método es aplicable a la obtención de la cadena media, pues este problema puede verse como un problema de optimización, donde el índice a optimizar es la suma de distancias de una cadena al conjunto de cadenas. En este caso, los elementos básicos a ir añadiendo a la solución actual son nuevos símbolos.

Así se llega a la llamada aproximación constructiva y voraz, propuesta en [9].

Entrada: $T = \{1, 2, \ldots, n\}$ conjunto de elementos básicos
Salida: Q conjunto que optimiza la función objetivo
Inicio
$Q^0 = \emptyset$
$t = 1$
Hacer
/* Escogemos el mejor elemento */
$j_t = \text{argmin}_{j \in T - Q^{t-1}} z(Q^{t-1} \cup \{j\})$
/* Si mejora lo añadimos */
Si $z(Q^{t-1} \cup \{j_t\}) < z(Q^{t-1})$ **Entonces** $Q^t = Q^{t-1} \cup \{j_t\}$
Sino $Q^t = Q^{t-1}$
FSi
Mientras $z(Q^t) < z(Q^{t-1})$ **Y** $Q^t \neq T$
Devolver Q^t

Figura 2.1: Algoritmo voraz de optimización

La idea consiste en ir construyendo la aproximación a la cadena media símbolo a símbolo, de manera que se va computando un prefijo de longitud creciente de dicha aproximación. En cada momento se tiene una cierta cadena candidata, y se generan a partir de ellas tantas cadenas como símbolos tenga el alfabeto, añadiendo dichos símbolos por el final. De ese conjunto nuevo de cadenas se escoge como nueva candidata la mejor de todas ellas. Este proceso prosigue hasta que se cumple un cierto criterio de parada. Una descripción algorítmica de este proceso se presenta en la Figura 2.2. En este algoritmo, se itera mientras haya una mejora de la distancia acumulada. A cada iteración, se añade a la cadena actual cada símbolo del alfabeto, computando para cada uno el valor de la distancia acumulada a los prefijos óptimos (los que dan menor distancia). Finalmente, se escoge el símbolo con menor distancia acumulada, se añade definitivamente a la cadena actual para formar la siguiente candidata y se vuelve a iterar.

En principio, este algoritmo viene a tener un coste temporal $O(l^2 \cdot |\Sigma| \cdot n)$, con l longitud máxima de las cadenas de S, por cada iteración principal. Esto se debe a que en cada iteración, para cada símbolo de Σ, es necesario calcular las distancias (coste cuadrático con la longitud de las cadenas en el caso de distancias de edición) entre la nueva candidata m' y todas las cadenas de S. Si suponemos que como mucho hace un número de iteraciones igual a la longitud de la cadena de longitud máxima, tendremos que este algoritmo presenta un coste total de $O(l^3 \cdot |\Sigma| \cdot n)$.

Sin embargo, la implementación del algoritmo descrita en [9] tiene la peculiaridad de estar basada en el algoritmo de Programación Dinámica que calcula la distancia de edición (no normalizada). Esto hace que presente un interesante coste temporal que luego examinaremos, pero a cambio tiene la dificultad de no ser adaptable (al menos con facilidad) a otras medidas de disimilitud. El algoritmo viene descrito en la Figura 2.3.

Entrada: $S = \{s^1, s^2, \ldots, s^n\}$
Salida: m cadena media aproximada
Inicio
$m = \lambda$ /* Candidata inicial: cadena vacía */
$dmin = \sum_{i=1}^{n} d(s^i, m)$
$dminaux = 0$
Mientras $dminaux < dmin$ /* Mientras mejore */
  $dm = \infty$
  ParaTodo $a \in \Sigma$
    $m' = m \cdot a$ /* Nuevas candidatas */
    /* Distancia acumulada a los prefijos óptimos */
    $dm' = \sum_{i=1}^{n} \text{mín}_{s' \in \text{pref}(s^i)} d(s', m')$
    Si $dm' < dm$ **Entonces** /* Si mejora la toma */
      $dm = dm'$
      $minm = m'$
    FSi
  FParaTodo
  $dminaux = \sum_{i=1}^{n} d(s^i, minm)$
  /* La toma si hay mejora global */
  Si $dminaux < dmin$ **Entonces** $m = minm$ **FSi**
FMientras
Devolver m

Figura 2.2: Esquema básico del algoritmo constructivo voraz. La función pref(s) proporciona el conjunto de prefijos de la cadena s.

Entrada: $S = \{s^1, s^2, \ldots, s^n\}$
Salida: m cadena media aproximada
Variables:
  $punt_s$: **vector** $[0 \ldots |s|, 0 \ldots 1]$ de enteros $(\forall s \in S)$
  tmp_s : **vector** $[0 \ldots |s|, \Sigma]$ de enteros $(\forall s \in S)$
  fin : **vector** $[0 \ldots \max]$ de enteros
Inicio
$m = \lambda$
ParaTodo $s \in S$ /* Inicializa pesos de inserciones */
  $punt_s[0,0] = 0$
  Para $j = 1, \ldots, |s|$ $punt_s[j, 0] = j \times ins(s_j)$ **FPara**
FParaTodo
Para $j = 1$ **Hasta** $\max_{s \in S} |s|$ /* Hasta $|s|$ símbolos */
  $k = j$ **mod** 2; $l = (j - 1)$ **mod** 2 /* Índices a intercambiar */
  $simbmin = \infty$
  ParaTodo $a \in \Sigma$
    $add = 0$
    ParaTodo $s \in S$ /* Para todas las cadenas */
      $tmp_s[0,a] = j$; $min = \infty$
      Para $i = 1, \ldots, |s|$ /* Distancia a los todos prefijos */
        $tmp_s[i, a] = \min \{punt_s[i, l] + borr(a), tmp_s[i - 1, a] + ins(s_i),$
                $punt_s[i - 1, l] + sust(s_i, a)\}$
        Si $min > tmp_s[i, a]$ **Entonces** $min = tmp_s[i, a]$ **FSi**
      FPara
      $add = add + min$ /* Hace el acumulado */
    FParaTodo
    /* Toma el símbolo de menor acumulado */
    Si $simbmin > add$ **Entonces** $simbmin = add$; $simb = a$; **FSi**
  FParaTodo
  $m = m \cdot simb$ /* Añade símbolo */
  $fin[j] = \sum_{s \in S} tmp_s(|s|, simb)$ /* Calcula criterio de parada */
  ParaTodo $s \in S$
    Para $i = 1, \ldots, |s|$ $punt_s[i, k] = tmp_s[i, simb]$ **FPara**
  FParaTodo
FPara
Devolver el prefijo de m de longitud l tal que $fin[l] < fin[l + 1]$

Figura 2.3: Algoritmo constructivo voraz de aproximación a la cadena media.

El algoritmo mantiene dos estructuras fundamentales. La primera de ellas es el conjunto de matrices de distancias para cada cadena $s \in S$; en este caso, se han implementado mediante un simple vector ($punt_s$), ya que no es necesario recuperar el camino y, como se indica en [80], se puede implementar el algoritmo de Programación Dinámica de esta manera. Inicialmente, $punt_s$ se inicializa con la acumulación de los pesos de inserción de cada símbolo de s (el valor $ins(s_i)$).

La otra estructura, la llamada fin, es un vector que indica, para la posición i (que varía desde 0 hasta la longitud máxima de las cadenas de S) el valor de la distancia acumulada de la candidata calculada hasta dicha posición a todas las cadenas de S. Esta estructura se usa para determinar qué prefijo de la cadena calculada es el que debe devolverse, es decir, determina el criterio de parada del algoritmo (aunque en esta implementación no provoca la parada).

En la Figura 2.4 se puede ver un ejemplo que clarifica el funcionamiento del algoritmo, usando la distancia de edición con pesos unitarios. Como se puede ver, inicialmente, la candidata es la cadena vacía, y se crean todas las matrices de distancia correspondientes (la primera columna), en las cuales todo son inserciones (de ahí que en el algoritmo se use el valor ins y en el ejemplo se observa que son valores incrementales de uno en uno). Posteriormente, a cada iteración se prueban todos los símbolos del alfabeto sobre la cadena actual (que inicialmente es la cadena vacía). Para ello, a la cadena actual se le añaden todos los símbolos y se recalculan las columnas actuales para cada cadena de S (es decir, todas las $|\Sigma|$ cadenas que surgen se comparan con todas las cadenas de S). Como se ve en el ejemplo, a partir de la cadena vacía λ, surgen las cadenas a y b y se calculan las distancias.

En esta construcción además se guarda el valor mínimo de la columna (variable min, marcado en negrita en las columnas del ejemplo) y se acumula (variable add). El símbolo que se añade finalmente a m ($simb$) es aquel que consigue el menor acumulado de mínimos ($simbmin$). Como se ve en el ejemplo para la primera iteración, la cadena a presenta una suma de mínimos igual a 1 y la cadena b una suma de mínimos de 2, por lo cual se toma la cadena a para la siguiente iteración.

Respecto al criterio de parada, se usa la estructura fin para almacenar las distancias acumuladas de cada una de las cadenas candidatas, es decir, el valor de la última fila en la columna i-ésima. Cuando ese valor es mayor para la columna actual respecto a la anterior, se acaba el proceso (la cadena escogida será la candidata en la iteración previa). En el ejemplo, como se ha escogido la cadena a, se calcula la suma de distancias acumuladas (en cursiva en el ejemplo) para esa cadena, con valor 7. Posteriormente, se van escogiendo las cadenas ab, con acumulado 4, abb, con acumulado 3 y finalmente $abbb$, con acumulado 4. Como dicho acumulado supera al obtenido previamente, se deshace esa última iteración y se devuelve abb.

El coste temporal del algoritmo es $O(l^2 \cdot |\Sigma| \cdot n)$, ya que se hace el bucle hasta la longitud máxima en el caso peor, y en este bucle para cada símbolo de Σ se hace el cálculo de la columna completa (que como mucho es de la máxima longitud l) para las n cadenas de S. Por tanto, el cómputo de esta aproximación presenta una complejidad temporal menor que el cálculo de la cadena mediana cuando el

número de símbolos es menor que el número de cadenas (lo cual suele ser más que usual). También presenta una rebaja en un orden de magnitud respecto al algoritmo general presentado en la Figura 2.2. En [9] se demuestra también que, en general, esta aproximación presenta un mejor comportamiento a la hora de identificar patrones correctos. Sin embargo, no hay resultados respecto a su uso como prototipos en clasificación.

La principal desventaja de este algoritmo es la dificultad de adaptarlo a otras distancias, y más específicamente a la distancia de edición normalizada que definimos en el apartado 1.5.2. Este problema se debe a la no monotonicidad del mejor camino en el cómputo de la distancia de edición normalizada; es decir, un subcamino (que parta del inicio) de un camino óptimo no es necesariamente el óptimo para los prefijos implicados.

Veamos un ejemplo simple que lo demuestra. Si tenemos las cadenas *abbb* y *bbbb* y calculamos su distancia de edición normalizada usando la matriz de pesos definida en la Figura 2.5, su camino óptimo es el que se muestra en la Figura 2.6, con un valor de $0'75$. Sin embargo, si tomamos los prefijos *ab* y *bb*, si tomáramos el subcamino correspondiente como óptimo, éste nos daría un valor de distancia de $1'5$, mientras que el auténtico camino óptimo, mostrado en la Figura 2.7, presenta un valor de la distancia de $1'33$. Por tanto, no es posible aplicar el algoritmo constructivo voraz, siguiendo esta implementación, para la distancia de edición normalizada, ya que se asumen que los cálculos previos son válidos al ir añadiendo símbolos y eso no es cierto en la versión normalizada.

Cabe de todas maneras la posibilidad de tomar como otra aproximación el uso de la distancia de edición normalizada conservando el camino previo, o el aplicar técnicas de postnormalización [51, 37] al añadir cada símbolo. Pero la única posibilidad de usar estrictamente la distancia de edición normalizada en esta aproximación voraz es incorporarla en el algoritmo general presentado en la Figura 2.2 (es decir, d en dicho algoritmo debe ser la distancia de edición normalizada). Como dicho algoritmo presenta un coste temporal semejante a los que se presentarán posteriormente en el Capítulo 3 y posee un espacio de búsqueda más reducido, resulta discutible la necesidad de implementarlo.

2.2. Experimentos comparativos

En esta sección vamos a aplicar la aproximación voraz presentada en la Sección 2.1 y la cadena mediana para extraer prototipos de diversos agrupamientos y usarlos con clasificadores k-NN, comparando los resultados de clasificación que ofrecen ambos tipos de prototipos. Estos experimentos se realizarán sobre un corpus de cromosomas que se describe en el apartado 2.2.1.

2.2.1. Corpus de cromosomas *Copenhagen*

Dentro del campo de la citogenética, una de las tareas primordiales es la clasificación de cromosomas de seres humanos [41]. En el ser humano, la dotación cromosómica consta de 22 pares de cromosomas homólogos (autosomas) y un par

$\Sigma = \{a, b\}$
$S = \{aabb, ab, bbb\}$

Iteración 1: Candidata = λ
Cadenas a verificar = {a,b}

b	4	*3*
b	3	2
a	2	1
a	1	**0**
	0	1
		a

b	2	*1*
a	1	**0**
	0	1
		a

b	3	*3*
b	2	2
b	1	1
	0	**1**
		a

b	4	*3*
b	3	2
a	2	2
a	1	1
	0	**1**
		b

b	2	*1*
a	1	1
	0	**1**
		b

b	3	*2*
b	2	1
b	1	**0**
	0	1
		b

Valor para *a*: 0 + 0 + 1 = 1 Valor para *b*: 1 + 1 + 0 = 2

Nueva candidata: a Valor de finalización: 3 + 1 + 3 = 7

Iteración 2: Candidata = a
Cadenas a verificar = {aa,ab}

b	3	*2*
b	2	1
a	1	**0**
a	0	1
	1	2
		aa

b	1	*1*
a	0	**1**
	1	2
		aa

b	3	*3*
b	2	2
b	1	2
	1	**2**
		aa

b	3	*2*
b	2	1
a	1	1
a	0	**1**
	1	2
		ab

b	1	**0**
a	0	1
	1	2
		ab

b	3	*2*
b	2	1
b	1	**1**
	1	2
		ab

Valor para *aa*: 0 + 1 + 2 = 3 Valor para *ab*: 1 + 0 + 1 = 2

Nueva candidata: *ab* Valor de finalización: 2 + 0 + 2 = 4

Iteración 3: Candidata = ab
Cadenas a verificar = {aba,abb}

b	2	*2*
b	1	2
a	1	**1**
a	1	2
	2	3
		aba

b	0	**1**
a	1	2
	2	3
		aba

b	2	*2*
b	1	2
b	1	**2**
	2	3
		aba

b	2	*1*
b	1	**1**
a	1	2
a	1	2
	2	3
		abb

b	0	**1**
a	1	2
	2	3
		abb

b	2	*1*
b	1	**1**
b	1	2
	2	3
		abb

Valor para *aba*: 1 + 1 + 2 = 4 Valor para *abb*: 1 + 1 + 1 = 3

Nueva candidata: *abb* Valor de finalización: 1 + 1 + 1 = 3

Iteración 4: Candidata = abb
Cadenas a verificar = {abba,abbb}

b	1	*2*
b	1	2
a	2	**2**
a	2	3
	3	4
		abba

b	1	**2**
a	2	3
	3	4
		abba

b	1	*2*
b	1	**2**
b	2	3
	3	4
		abba

b	1	**1**
b	1	2
a	2	2
a	2	3
	3	4
		abbb

b	1	**2**
a	2	3
	3	4
		abbb

b	1	**1**
b	1	2
b	2	3
	3	4
		abbb

Valor para *abba*: 2 + 2 + 2 = 6 Valor para *abbb*: 1 + 2 + 1 = 4

Nueva candidata: *abbb* Valor de finalización: 1 + 2 + 1 = 4

Nuevo valor de finalización mayor que el antiguo: fin del proceso.

Cadena devuelta: **abb**

Figura 2.4: Ejemplo de funcionamiento del algoritmo constructivo voraz de aproximación a la cadena media.

γ	a	b	λ
a	0	3	2
b	3	0	2
λ	2	2	-

Figura 2.5: Matriz de pesos para $\Sigma = \{a, b\}$

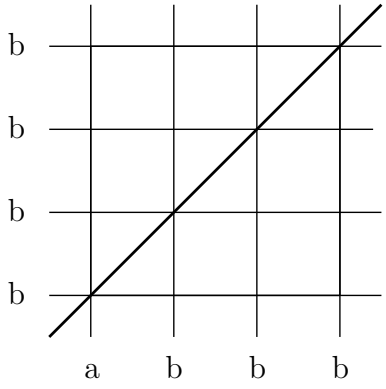


Figura 2.6: Camino óptimo para la distancia de edición normalizada para *abbb* y *bbbb*.

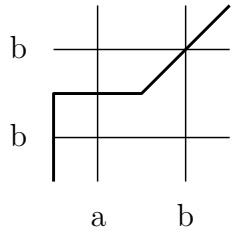


Figura 2.7: Camino óptimo para la distancia de edición normalizada para los prefijos *ab* y *bb*.

Figura 2.8: Metafase de un varón normal

de cromosomas sexuales. La diferenciación entre los distintos pares de cromosomas se suele realizar durante la mitosis celular, especialmente en la metafase, pues es el momento en el cual el material genético está más condensado y, por tanto, más distinguibles son los cromosomas.

Para la clasificación de los cromosomas por parte de los expertos humanos, se suele partir de una imagen del conjunto de cromosomas del individuo, como pueda ser la de la Figura 2.8, y se realiza la clasificación usando básicamente las características siguientes:

- Tamaño
- Posición del centrómero
- Bandeo del cromosoma

El uso de técnicas de RF para la clasificación (semi)automática de cromosomas resulta, claramente, de una gran ayuda [8]. En primer lugar es necesario aislar el cromosoma a clasificar (que ya de por sí es un problema importante [1]).

Tras ello, se realiza una extracción de características del mismo, ya sea de forma vectorial [64] o en forma de cadena de caracteres [22].

En nuestro caso, el corpus de cromosomas *Copenhagen* [44] se codifica como cadenas de caracteres. Este corpus presenta los cariotipos de un total de 180 células sanguíneas de 12 donantes sanos (7 masculinos y 5 femeninos). La adquisición del conjunto de cromosomas se hizo a partir de negativos de las fotografías de la mitosis celular, adquiriéndose las imágenes con un periodo de muestreo de $\frac{1}{8}\mu$m. En dicho proceso de adquisición, los cromosomas se alinearon a lo largo de su eje longitudinal y se mantuvo en todos los casos la misma polaridad. Ciertos cromosomas que presentaban dificultades especiales (solapamiento con otros cromosomas, exceso de curvatura, etc.) fueron desechados, quedando la adquisición final en un total de 6985 cromosomas. Por cada cromosoma, junto a la imagen, se codificó la posición del centrómero.

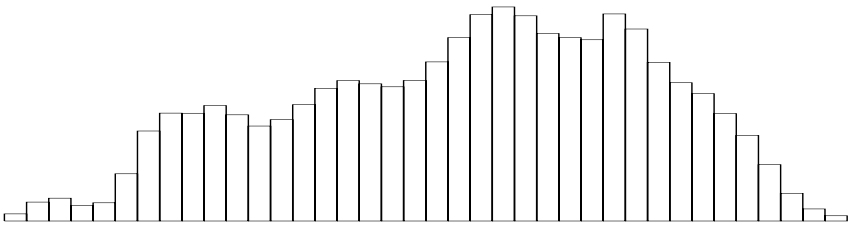
Por cada cromosoma codificado se realiza una traza. Esta traza es una secuencia de medidas calculada como sigue: se recorre el eje longitudinal del cromosoma, trazando las cuerdas normales a dicho eje y promediando el nivel de brillo de la imagen para dicha cuerda. Una vez obtenida la traza, se reduce su nivel de ruido aplicando una máscara de convolución sobre la misma.

Tras ello, una técnica de análisis descrita en [21] se aplica sobre dicha traza suavizada, a fin de obtener las inflexiones entre extremos consecutivos, que en cierta manera representarán las transiciones entre bandas de distinto brillo dentro del cromosoma. Se tomará el valor extremo (máximo o mínimo) de la banda estimada como el valor que representa a dicha banda. La traza resultante se denomina traza idealizada.

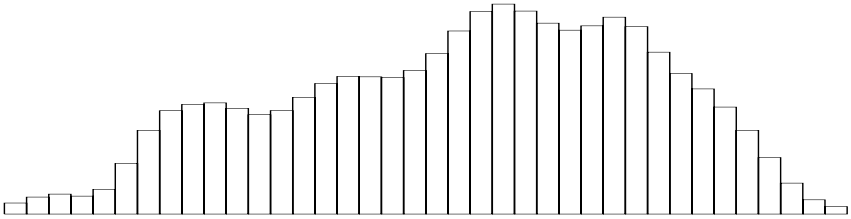
A partir de la traza idealizada es sencillo obtener una representación simbólica. Dicha representación asignaría un valor a cada uno de los niveles presentes en la traza idealizada, reduciendo de manera heurística el total de niveles a 6, de manera que tendríamos cadenas sobre el alfabeto $\Sigma = \{1, 2, 3, 4, 5, 6\}$. Sin embargo, esta representación no es la más adecuada ya que la fuente de información primordial a la hora de clasificar cromosomas es la diferencia entre las bandas del mismo. Por tanto, la codificación adecuada debe enfatizar las transiciones entre las distintas bandas del cromosoma.

Para ello se usa la llamada cadena de diferencias. Esta cadena se obtiene a partir de la cadena de seis símbolos del proceso previo, asignando un símbolo distinto según la diferencia existente entre dos símbolos consecutivos. El alfabeto usado es $\Sigma = \{e, d, c, b, a, =, A, B, C, D, E\}$, en el que cada símbolo corresponde a la diferencia $\{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$, respectivamente. En todos los casos, el símbolo inicial y el final de la cadena de diferencias se determina asumiendo que la cadena sobre $\Sigma = \{1, 2, 3, 4, 5, 6\}$ está limitada por ceros. Una revisión de todo el proceso se puede ver en la Figura 2.9.

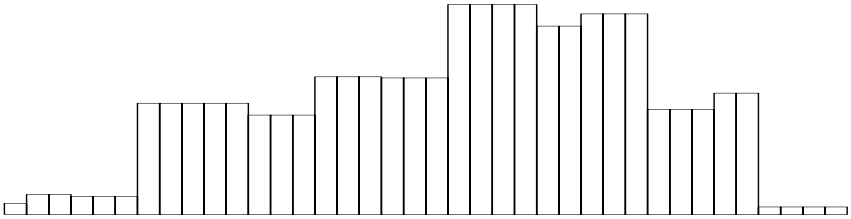
El corpus final excluye varios cromosomas (entre ellos los sexuales), de manera que finalmente incluye únicamente 200 cromosomas de cada una de las 22 clases, es decir, un total de 4400 cromosomas codificados como cadenas de diferencias. Existe una variante de dicho corpus que incluye la posición del centrómero (marcada con el símbolo X), pero que no usaremos en nuestros experimentos. Un resumen de las características de este corpus puede verse en



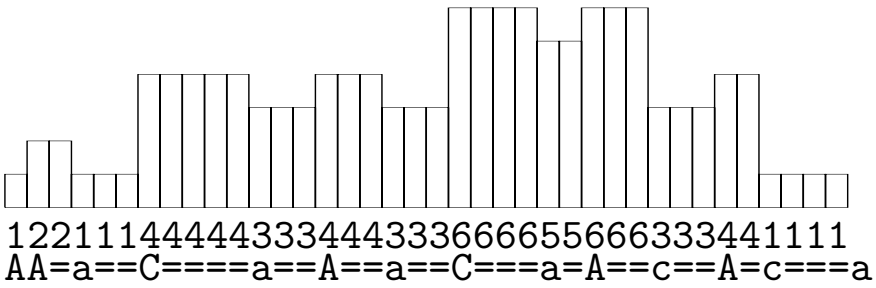
(a) Traza de un cromosoma de la clase 13



(b) Traza suavizada mediante la máscara de convolución $(1, 2, 1)$



(c) Traza *idealizada* obtenida tras localizar inflexiones entre extremos consecutivos



(d) Arriba: traza cuantificada no linealmente en 6 niveles. En medio: representación mediante una cadena de símbolos en $\{1, 2, 3, 4, 5, 6\}$. Abajo: cadena de símbolos en $\{\ldots, a, =, A, \ldots\}$ correspondientes a diferencias de $\{\ldots, -1, 0, +1, \ldots\}$ entre símbolos consecutivos.

Figura 2.9: Obtención de la cadena de diferencias a partir de la traza del cromosoma

Cuadro 2.1: Características del corpus *Copenhagen*

Número de clases	22
Número de objetos	4400
Tamaño del alfabeto	11
Longitud de las cadenas (mínima-máxima)	21-106

Cuadro 2.2: Resultados de clasificación para el corpus *Copenhagen* con diversos métodos

Técnica de clasificación	% Error
k-NN (distancia normalizada)	3'9 %
Redes restringidas de Markov [24]	7'3 %
Algoritmo ECGI [77]	7'5 %
Perceptrón multicapa [77]	9'1 %
Modelos ocultos de Markov [78]	9'3 %

el Cuadro 2.1.

De cara al proceso experimental, debido a que el corpus es bastante reducido se impone el uso de la técnica de validación cruzada [15]. En nuestro caso, el conjunto de datos se ha dividido en dos mitades a y b de 2200 cadenas, 100 cadenas por clase, empleándose a para obtener prototipos y b como conjunto de prueba, y viceversa. Los resultados de clasificación obtenidos son, evidentemente, la media para las dos clasificaciones realizadas. Los resultados obtenidos hasta ahora con diversos métodos de clasificación usando este esquema de validación cruzada pueden verse en el Cuadro 2.2.

Igualmente, de cara a obtener varios prototipos por clase y poder así utilizar clasificadores k-NN, se impone la división en agrupamientos de cada clase. En nuestro caso, el algoritmo de obtención de agrupamientos usando es el k-medianas (descrito en el apartado 1.4.3) con inicialización por maxmin (tal y como se ha descrito en el apartado 1.4.4) [33]. Concretamente, en nuestros experimentos hemos usado desde un solo agrupamiento por clase (es decir, no hacer divisiones por clase) hasta 100 (es decir, cada muestra es un agrupamiento por sí misma), yendo de 1 a 9 y de 10 a 100 en intervalos de 10 agrupamientos. El número de agrupamientos coincide con el número de prototipos que se tendrán disponibles por clase, por lo cual hacer 100 agrupamientos por clase corresponde a usar el conjunto total de entrenamiento como conjunto de prototipos.

La obtención de prototipos y la clasificación se realizó usando la matriz de pesos de la Figura 2.10. Esta matriz asigna pesos de inserción y borrado distintos a los usados en la mayor parte de los trabajos con este corpus [32] y que ha demostrado otorgar mejores resultados de clasificación [48]. La asociación de dichos pesos de inserción y borrado corresponde al promedio de los pesos de

γ	e	d	c	b	a	=	A	B	C	D	E	λ
e	0	1	2	3	4	5	6	7	8	9	10	5
d	1	0	1	2	3	4	5	6	7	8	9	5
c	2	1	0	1	2	3	4	5	6	7	8	4
b	3	2	1	0	1	2	3	4	5	6	7	4
a	4	3	2	1	0	1	2	3	4	5	6	3
=	5	4	3	2	1	0	1	2	3	4	5	3
A	6	5	4	3	2	1	0	1	2	3	4	3
B	7	6	5	4	3	2	1	0	1	2	3	4
C	8	7	6	5	4	3	2	1	0	1	2	4
D	9	8	7	6	5	4	3	2	1	0	1	5
E	10	9	8	7	6	5	4	3	2	1	0	5
λ	5	5	4	4	3	3	3	4	4	5	5	-

Figura 2.10: Matriz de pesos usada para el corpus *Copenhagen*

sustitución de cada símbolo, ya que parece evidente que para el símbolo que, en promedio, cueste menos de sustituir por cualquier otro símbolo, será menos costoso insertarlo o borrarlo. Los valores de los pesos de sustitución se asignan en función de la diferencia de la variación de niveles de brillo que cada símbolo indica (es claro que dos símbolos que indiquen variaciones muy diferentes son más costosos de sustituir entre sí que dos símbolos que indiquen variaciones más parecidas).

2.2.2. Comparación entre la cadena mediana y la aproximación voraz

La primera aproximación descrita para la cadena media es la que hemos llamado aproximación constructiva voraz, descrita en la Sección 2.1. Por tanto, una primera comparación entre el prototipo obtenido mediante este método y la cadena mediana resulta necesaria. Dado que la aproximación voraz sólo es aplicable para la distancia de edición no normalizada, se hace necesario utilizar esa misma medida de distancia en la obtención de la cadena mediana y en la clasificación posterior, a fin de que ambos prototipos se obtengan en igualdad de condiciones.

Los resultados se presentan en las gráficas de la Figura 2.11 desde 1 a 100 agrupamientos para clasificadores por 1 y 12 vecinos (para otro número de vecinos presenta un comportamiento semejante). Como se puede observar, la cadena mediana presenta un comportamiento mucho mejor en clasificación con respecto a la aproximación voraz, aunque las diferencias se van reduciendo a medida que el número de agrupamientos aumenta (debido sobre todo al incremento de agrupamientos unitarios, en los cuales ambos prototipos coinciden). Los intervalos de confianza no se muestran pues las diferencias son claramente significativas hasta un número elevado de agrupamientos (del orden de 50 en el clasificador

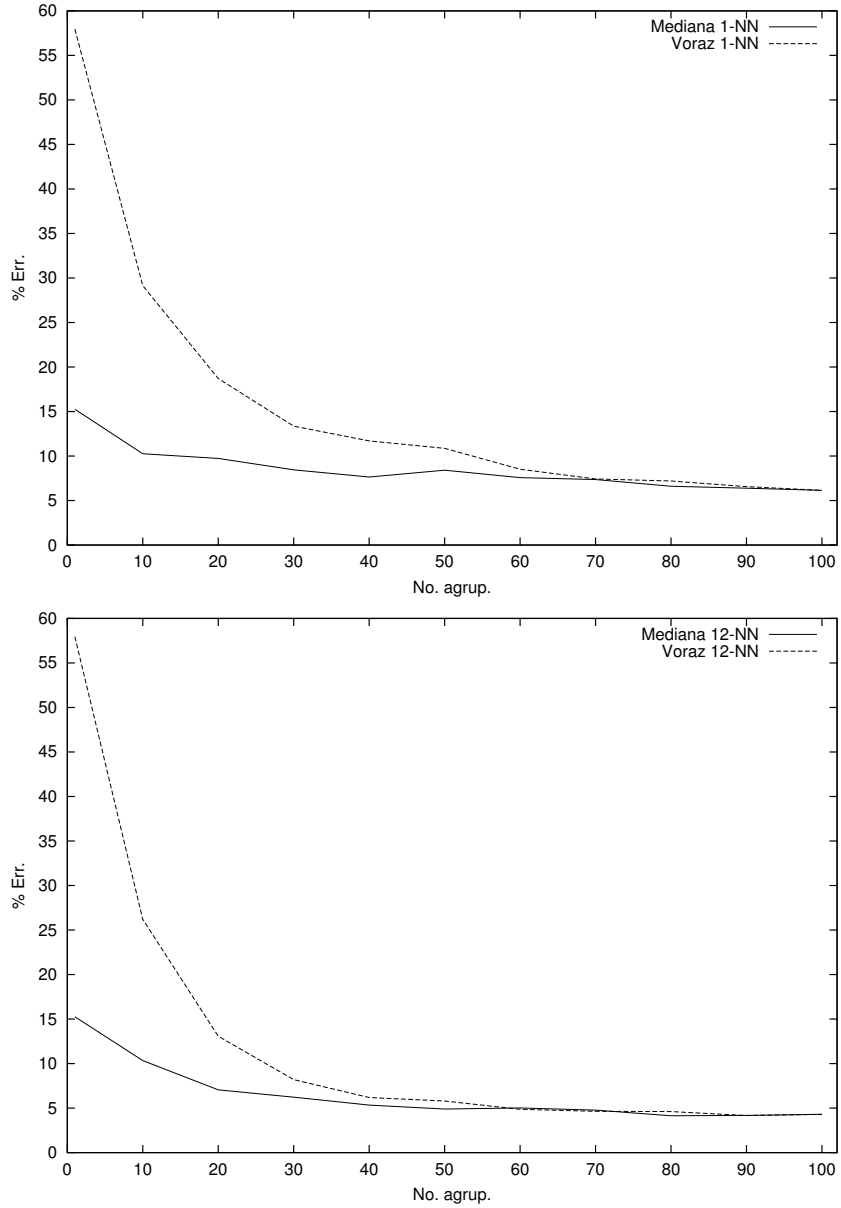


Figura 2.11: Resultados de clasificación usando la cadena mediana y la cadena media aproximada voraz, obtenidas para la distancia de edición no normalizada usando clasificadores 1-NN y 12-NN por distancia de edición no normalizada

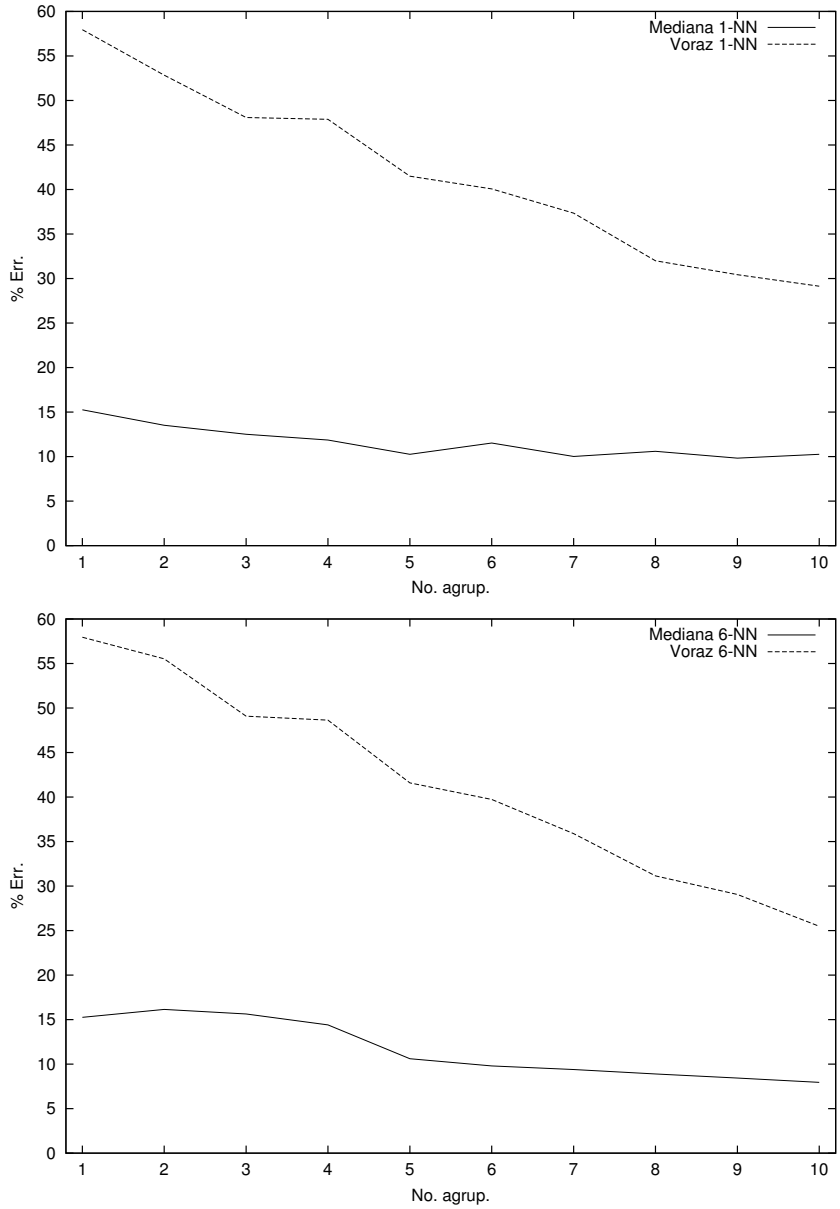


Figura 2.12: Detalle de los resultados de clasificación de 1 a 10 agrupamientos usando la cadena mediana y la cadena media aproximada voraz, obtenidas para la distancia de edición no normalizada usando clasificadores 1-NN y 6-NN por distancia de edición no normalizada

1-NN). El detalle para un número de agrupamientos entre 1 y 10, para 1 y 6 vecinos, se muestra en la Figura 2.12, donde se confirma la gran ventaja que presenta la cadena mediana.

También existe la posibilidad de usar la distancia de edición normalizada en la obtención de la cadena mediana y la posterior clasificación. Sin embargo, a priori parece claro que la aproximación voraz va a ofrecer peores resultados en dicha situación, ya que la aproximación voraz está concebida en base a la distancia de edición no normalizada. Esta intuición previa viene confirmada por los resultados de clasificación que se pueden ver en las gráficas de las Figuras 2.13 y 2.14. Sólo se muestran estos cuatro casos, ya que en los restantes el comportamiento es, de nuevo, muy semejante. Tampoco se muestran los intervalos de confianza ya que de nuevo las diferencias que presentan son significativas hasta un cierto número de agrupamientos.

Para comparar también la ventaja que otorga usar la distancia de edición normalizada frente a la no normalizada, en la Figura 2.15 y 2.16 se muestran los resultados de clasificación para ambos casos con la cadena mediana, incluyendo los intervalos de confianza del 95 %. Los intervalos de confianza nos muestran ahora que las diferencias no son realmente significativas, pero en todo caso funciona, en términos absolutos, mejor la distancia de edición normalizada que la no normalizada.

Por tanto, de estos resultados se puede concluir que la aproximación constructiva voraz a la cadena media no resulta un prototipo adecuado para la clasificación, ya que la cadena mediana resulta en general mucho más efectiva como representante de un conjunto de cadenas. También se concluye que, aunque las diferencias en clasificación no son especialmente significativas, el uso de la distancia de edición normalizada nos llevará a obtener mejores resultados que usando la no normalizada.

2.3. Resumen

En esta capítulo hemos ofrecido una primera aproximación a la cadena media (ya que este es un problema NP-Duro), usando un algoritmo constructivo voraz. Como el esquema general propuesto presenta un coste elevado (orden cúbico con la longitud de las cadenas), se recurre a una implementación con un coste menor con la desventaja de que sólo puede usarse la definición no normalizada de distancia de edición para su obtención. A fin de verificar la idoneidad de esta aproximación frente a la cadena mediana, se han realizado experimentos sobre un corpus de cromosomas, el corpus *Copenhagen*, demostrando que esta aproximación no resulta de la calidad suficiente respecto a la cadena mediana. Por último, se ha constatado que el uso de la distancia de edición normalizada otorga resultados ligeramente mejores en clasificación que la distancia no normalizada.

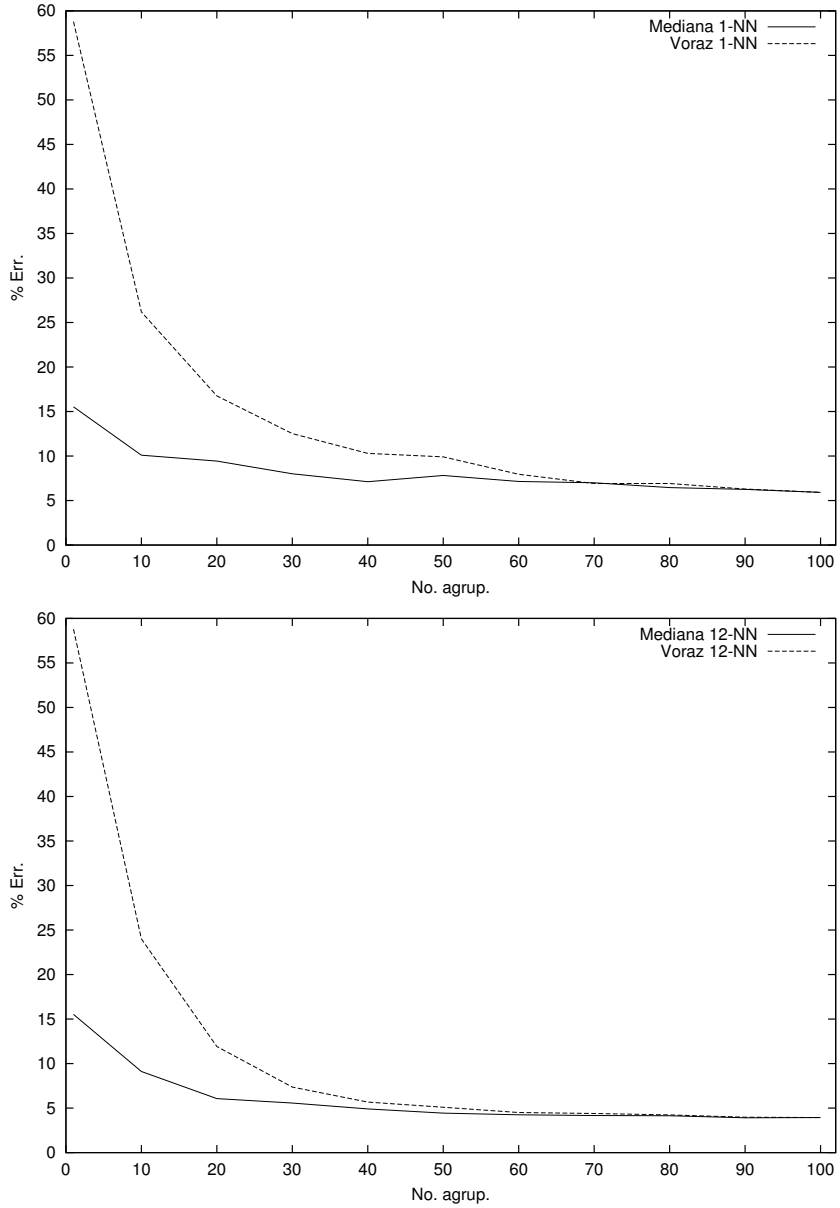


Figura 2.13: Resultados de clasificación con la distancia de edición normalizada usando la cadena mediana obtenida con distancia de edición normalizada y la cadena media aproximada voraz para 1-NN y 12-NN

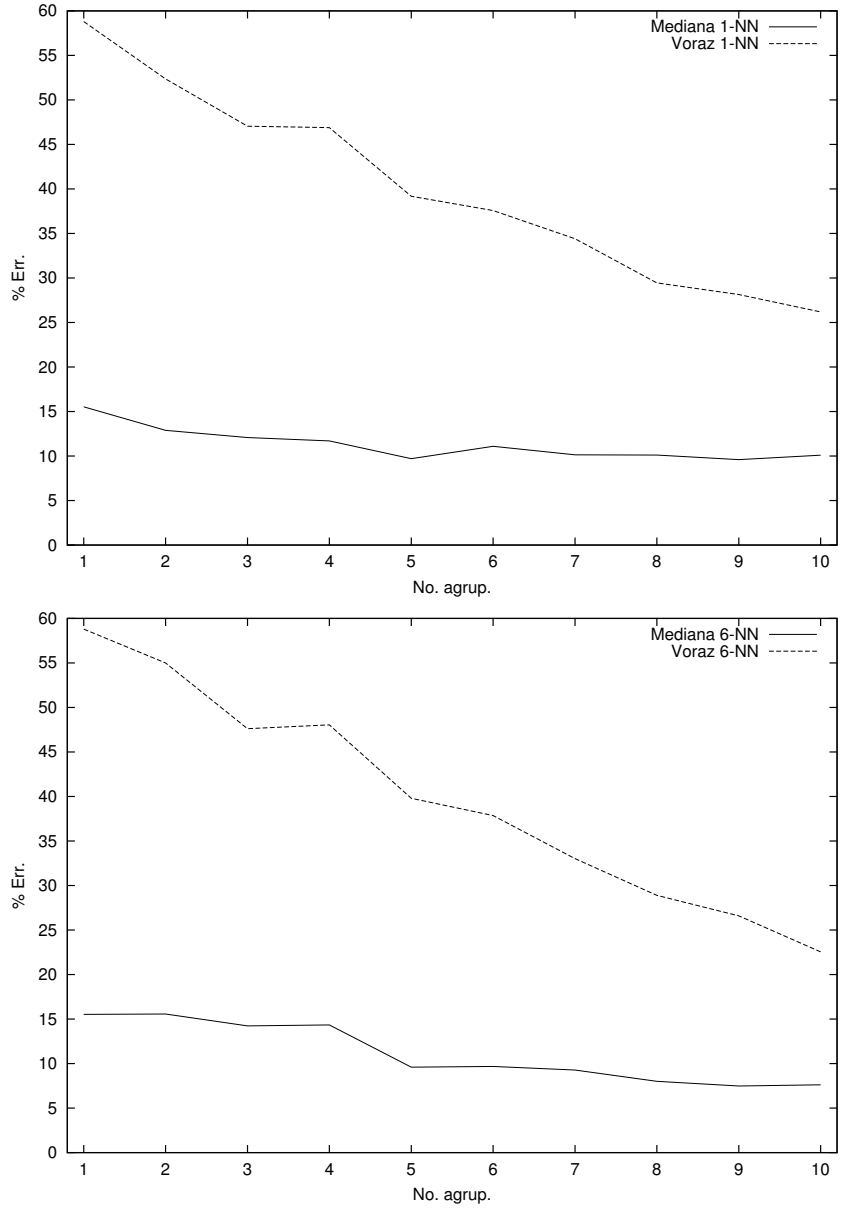


Figura 2.14: Detalle de los resultados de clasificación desde 1 a 10 agrupamientos con la distancia de edición normalizada usando la cadena mediana obtenida con distancia de edición normalizada y la cadena media aproximada voraz para 1-NN y 6-NN

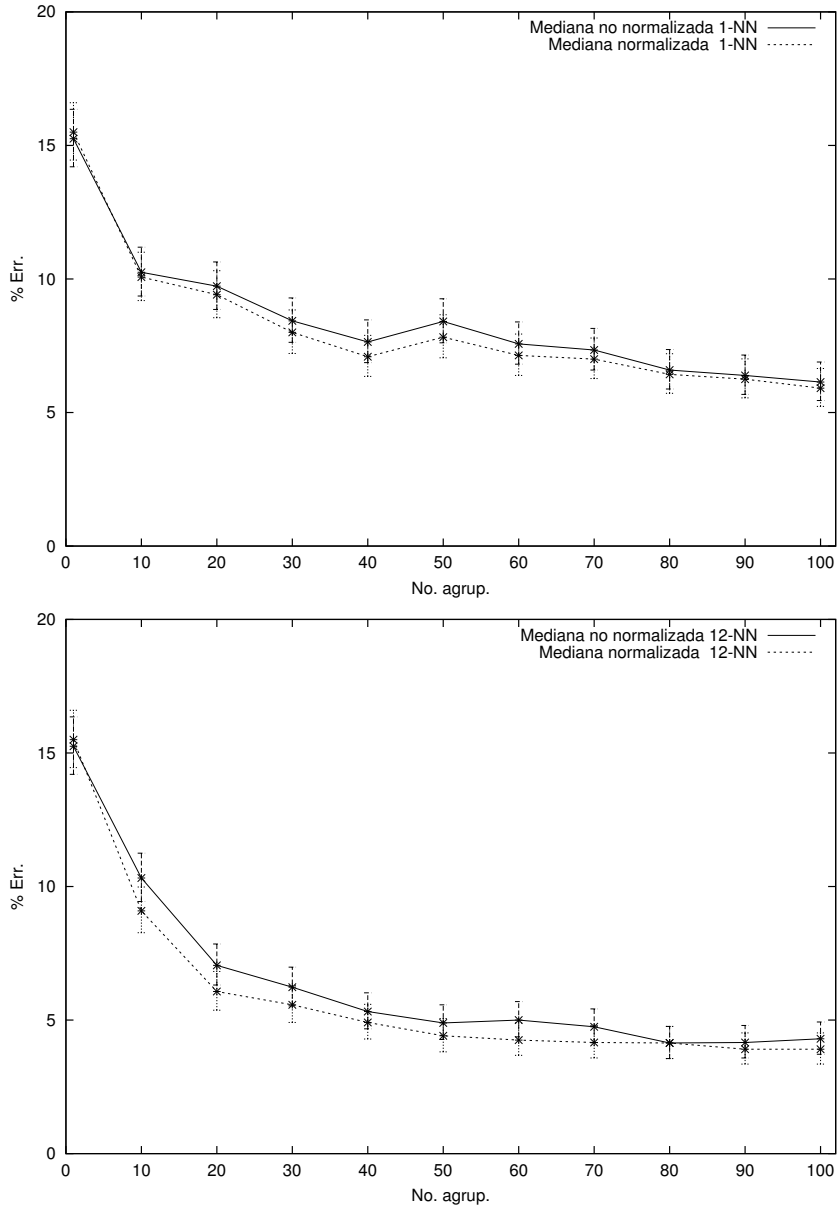


Figura 2.15: Resultados de clasificación para la cadena mediana según se use distancia de edición no normalizada y normalizada, para 1 y 12 vecinos

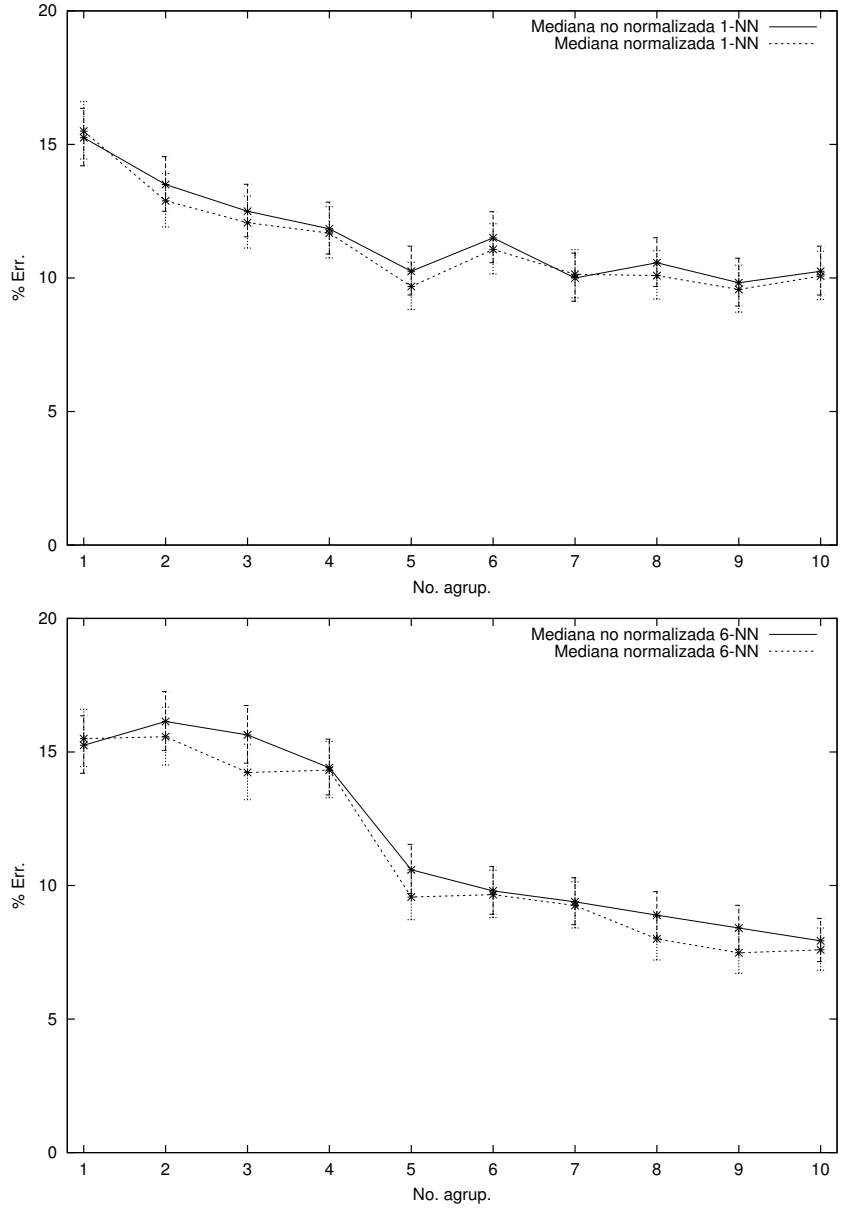


Figura 2.16: Detalle para el intervalo de 1 a 10 agrupamientos de los resultados de clasificación para la cadena mediana según se use distancia de edición no normalizada y normalizada, para 1 y 6 vecinos

Capítulo 3

Aproximaciones iterativas a la cadena media

En este capítulo vamos a pasar a analizar otras posibles aproximaciones a la cadena media que se pueden obtener para su aplicación práctica, lo cual constituye el tema fundamental de esta tesis. En primer lugar, se van a describir los métodos para obtener las aproximaciones basados en un proceso iterativo perturbativo. Luego se describirán someramente otras alternativas para obtener aproximaciones y finalmente se expondrán resultados experimentales que permiten comparar el comportamiento en clasificación de las diversas aproximaciones propuestas en primer lugar.

3.1. Aproximaciones iterativas a la cadena media

El uso de la cadena mediana o la aproximación constructiva voraz descrita en la Sección 2.1 resultan aproximaciones simples y poco costosas. Sin embargo, ya se han mostrado resultados que muestran el pobre rendimiento de la aproximación voraz en clasificación. Por otra parte, en el artículo original en el que se define la cadena media [38], Kohonen plantea la obtención de la cadena media mediante un proceso iterativo de perturbación, y muestra que dicho proceso ofrece mejores resultados que la cadena mediana, aunque no realiza experimentos exhaustivos de clasificación ni describe dicho proceso perturbativo.

Basándose en esta idea, en esta sección exponemos una serie de métodos basados en perturbación sistemática que nos permiten obtener aproximaciones a la cadena media exacta. Estos métodos están fundamentados en la idea del algoritmo heurístico de intercambio [60] descrito en el apartado 1.4.4. Su esquema algorítmico para el caso de minimización se puede ver en la Figura 3.1. La idea de dicho algoritmo consiste en tomar uno a uno todos los elementos de la solución actual e intercambiar el elemento actual por todos los posibles elementos

Entrada: $T = \{1, 2, \ldots, n\}$ conjunto de elementos básicos, $Q^0 \subset T$ solución inicial
Salida: $Q \subset T$ solución final
Inicio
$Q = Q^0$
Hacer
  $Q' = Q$
  $auxz = \infty$
  /* Para todos los eltos. de la sol. actual */
  ParaTodo $q \subset Q$
    /* Para todos los eltos. que no son de la sol. actual */
    ParaTodo $t \subset T - Q$
      /* Si al intercambiar mejora */
      Si $z(Q - \{q\} \cup \{t\}) < auxz$ **Entonces**
        $auxz = z(Q - \{q\} \cup \{t\})$
        $Q' = Q - \{q\} \cup \{t\}$
      FSi
    FPara
  FPara
Mientras $z(Q') < z(Q)$ /* Mientras mejore la nueva solución */
Devolver Q

Figura 3.1: Esquema general del algoritmo de intercambio

que no estén en dicha solución. En todos estos intercambios se evalúa la función objetivo a optimizar, y se escoge aquel intercambio que dé un mejor valor de la función objetivo, consiguiendo una posible solución. Si esta solución es mejor que la de la iteración previa, se toma como nueva solución y se vuelve a realizar el proceso. En caso contrario, se devolvería como solución final la del paso previo. Este esquema es una particularización del algoritmo de k-intercambio [60], donde los intercambios se hacen en conjuntos de k elementos.

Esta idea, en términos de un espacio de cadenas, consiste en aplicar alteraciones sobre cada posición de la cadena solución actual, pudiendo ser dichas alteraciones inserciones, borrados o sustituciones. La función objetivo que evaluará la calidad de la solución es la distancia acumulada al conjunto de cadenas del cual buscamos la cadena media.

3.1.1. Método iterativo separado

Basándose en la idea de un método por perturbación sistemática, una primera aproximación consistiría en aplicar, sucesivamente, las operaciones de sustitución, inserción y borrado sobre cada posición de la cadena a perturbar. Cuando la perturbación efectuada lleve a una cadena cuya distancia acumulada sea menor que la de la cadena previa, esta nueva cadena será tomada como nueva solución.

La propuesta es que este método realice en primer lugar todas las susti-

1. Entrada: cadena inicial s
2. Aplicar sustituciones sobre todas las posiciones de s y repetir mientras se siga mejorando; depositar cadena resultado en s'
3. Aplicar borrados sobre todas las posiciones de s' y repetir mientras se siga mejorando; depositar cadena resultado en s''
4. Si $s'' \neq s'$ aplicar sustituciones sobre s'' y depositar resultado en s'
5. Aplicar inserciones sobre todas las posiciones de s' y repetir mientras se siga mejorando; depositar cadena resultado en s'
6. Si $s' \neq s$, hacer $s = s'$ y volver a 2
7. Resultado: s

Figura 3.2: Esquema básico del método iterativo separado para la obtención de la cadena media aproximada

tuciones posibles sobre la cadena actual, buscando una mejora. Tras ello, se probará con todos los posibles borrados con el mismo objetivo. Si ha habido una mejora en el paso de borrado, se volverá a aplicar sustituciones (ya que si no ha habido mejora no tiene sentido). Tras ello, se probará con las inserciones en todas las posiciones de la cadena (teniendo en cuenta que las inserciones se pueden efectuar antes del primer símbolo de la cadena y después del último símbolo de la misma). Si este último proceso ha conseguido una mejora, se volvería a iterar.

Un esquema que describe este proceso se presenta en la Figura 3.2. Como vemos, el esquema presentado sólo cubre una de las combinaciones posibles de operaciones de edición (por ejemplo, se podría proponer un esquema en el que tras las inserciones se prueben de nuevo los borrados), pero parece un proceso que para obtener una solución aproximada es admisible. El algoritmo detallado de dicho proceso viene descrito en las Figuras 3.3 y 3.4, donde s_i denota el i-ésimo símbolo de la cadena s.

Una característica que presenta este algoritmo es que necesita como entrada una cadena inicial, sobre la cual se van a realizar las perturbaciones. Teniendo en cuenta que las perturbaciones provocan variaciones locales, parece razonable el usar como cadena inicial alguna cadena que esté razonablemente próxima a la cadena media exacta, ya que es aquella a la que buscamos aproximarnos mejor. Por tanto, una posible cadena inicial sería la cadena mediana. También es posible usar la cadena resultante del proceso constructivo voraz de la Sección 2.1. En general, cualquier cadena serviría, pero lo razonable parece usar alguna de esas dos aproximaciones iniciales, pues su coste de obtención resulta admisible y prácticamente despreciable respecto al coste del proceso de perturbación. También cabe identificar que el cálculo de la distancia acumulada se

Entrada: $m \in \Sigma^*$ solución inicial, $S = \{s^1, s^2, \ldots, s^n\}$
Salida: $m \in \Sigma^*$ solución final
Inicio
$d_a = \sum_{i=1}^{n} d(m, s^i)$
Hacer
  Hacer /* Aplica sustituciones */
    $d_p = d_a$
    Para $i = 1, \ldots, |m|$ /* En todas las posiciones */
      $sb = \emptyset$; $d_{aux} = \sum_{j=1}^{n} d(m, s^j)$
      ParaTodo $a \in \Sigma$ /* Para todos los símbolos */
        $aux = m$; $aux_i = a$
        Si $\sum_{j=1}^{n} d(aux, s^j) < d_{aux}$ **Entonces** /* Hay mejora */
          $sb = a$
          $d_{aux} = \sum_{j=1}^{n} d(aux, s^j)$
        FSi
      FParaTodo
      Si $sb \neq \emptyset$ **Entonces** $m_i = sb$ **FSi**
    FPara
    $d_a = \sum_{i=1}^{n} d(m, s^i)$
  Mientras $d_a < d_p$ /* Hasta que no hay mejora */
  $d_b = d_a$
  Hacer /* Aplica borrados */
    $d_p = d_a$
    $d_{aux} = \sum_{j=1}^{n} d(m, s^j)$
    Para $i = 1, \ldots, |m|$ /* En todas las posiciones */
      $aux = m_1 m_2 \cdots m_{i-1} m_{i+1} \cdots m_{|m|}$
      Si $\sum_{j=1}^{n} d(aux, s^j) < d_{aux}$ **Entonces** /* Hay mejora */
        $m = aux$
        $daux = \sum_{j=1}^{n} d(m, s^j)$
      FSi
    FPara
    $d_a = \sum_{i=1}^{n} d(m, s^i)$
  Mientras $d_a < d_p$ /* Hasta que no hay mejora */

Figura 3.3: Algoritmo del método iterativo separado para la obtención de la cadena media aproximada (parte 1 de 2).

Si $d_a < d_b$ **Entonces** /* Si borrados ha mejorado */
  Hacer /* Aplica sustituciones */
    $d_p = d_a$
    Para $i = 1, \ldots, |m|$ /* En todas las posiciones */
      $sb = \emptyset$; $d_{aux} = \sum_{j=1}^{n} d(m, s^j)$
      ParaTodo $a \in \Sigma$ /* Para todos los símbolos */
        $aux = m$; $aux_i = a$
        Si $\sum_{j=1}^{n} d(aux, s^j) < d_{aux}$ **Entonces** /* Hay mejora */
          $sb = a$
          $d_{aux} = \sum_{j=1}^{n} d(aux, s^j)$
        FSi
      FParaTodo
      Si $sb \neq \emptyset$ **Entonces** $m_i = sb$ **FSi**
    FPara
    $d_a = \sum_{i=1}^{n} d(m, s^i)$
  Mientras $d_a < d_p$ /* Hasta que no hay mejora */
FSi
$d_i = d_a$
Hacer /* Aplica inserciones */
  $d_p = d_a$
  Para $i = 1, \ldots, |m|$ /* En todas las posiciones */
    $sb = \emptyset$; $d_{aux} = \sum_{j=1}^{n} d(m, s^j)$
    ParaTodo $a \in \Sigma$ /* Para todos los símbolos */
      $aux = m_1 m_2 \cdots m_{i-1} \cdot a \cdot m_i m_{i+1} \cdots m_{|m|}$
      Si $\sum_{j=1}^{n} d(aux, s^j) < d_{aux}$ **Entonces** /* Hay mejora */
        $sb = a$
        $d_{aux} = \sum_{j=1}^{n} d(aux, s^j)$
      FSi
    FParaTodo
    Si $sb \neq \emptyset$ **Entonces** $m = m_1 \cdots m_{i-1} \cdot sb \cdot m_i \cdots m_{|m|}$ **FSi**
  FPara
  $d_a = \sum_{i=1}^{n} d(m, s^i)$
Mientras $d_a < d_p$ /* Hasta que no hay mejora */
Mientras $d_a < d_i$ /* Hasta que no hay mejora global */
Devolver m

Figura 3.4: Algoritmo del método iterativo separado para la obtención de la cadena media aproximada (parte 2 de 2).

hace aquí usando la definición de la cadena media dada en (1.21).

En cuanto a la complejidad computacional de este algoritmo, en principio sólo puede calcularse por iteración principal. Esto se debe a que el número de iteraciones totales del bucle externo (ni el de los bucles internos) está determinado. La posible cota del número de iteraciones del bucle principal vendría en función de tres factores:

1. La distancia acumulada de la cadena inicial a S, d_I
2. La distancia acumulada de la media real a S, d_m
3. El decremento mínimo de distancia acumulada entre una cadena y la cadena de la siguiente iteración, Δd

En función de estos tres factores, el número máximo de iteraciones sería $\frac{d_I - d_m}{\Delta d}$, ya que la mejor que se puede alcanzar desde la cadena inicial es la media exacta, lo cual garantiza una distancia finita que puede decrementarse en Δd a cada iteración hasta que se halla la cadena media exacta. Para cada bucle interno tendríamos una expresión semejante pero basándose únicamente en las cadenas obtenibles aplicando la operación de edición de dicho bucle y en las variaciones mínimas posibles con dicha operación.

El problema es que dos de estos factores son desconocidos: d_m y Δd. Para d_m veremos que es posible obtener cotas inferiores en el Capítulo 6. Sin embargo, en la obtención de Δd intervienen factores tan diversos como el número de cadenas de S, la longitud de las mismas, la distancia usada y el coste de las diversas operaciones de edición (coste que además puede depender de los símbolos implicados). Esto hace que sea un parámetro difícil de determinar.

Respecto a los bucles definidos, y suponiendo que la medida de disimilitud d es alguna de las distancias de edición propuestas en la Sección 1.5 (luego presentan un coste cuadrático), se dará que:

- Para la sustitución se tiene una complejidad temporal $O(l^3 \cdot |\Sigma| \cdot n)$ (suponiendo que l es la longitud máxima que va a alcanzar m, es decir, que no excede la máxima longitud de las cadenas de S), ya que para cada una de las l posiciones hay que aplicar las sustituciones con todos los $|\Sigma|$ símbolos y calcular la distancia acumulada a S, que tiene coste $O(l^2 \cdot n)$.
- Para el borrado se tiene $O(l^3 \cdot n)$, ya que no es necesario considerar los símbolos del alfabeto y sólo se calcula una distancia acumulada $O(l^2 \cdot n)$ para las l posiciones de la cadena.
- Para la inserción de nuevo $O(l^3 \cdot |\Sigma| \cdot n)$ (suponiendo nuevamente que la longitud de m no va a exceder l), pues el esquema es semejante al caso de la sustitución.

Por tanto, en global un bucle completo va a tener un coste $O(k \cdot l^3 \cdot |\Sigma| \cdot n)$, siendo k una constante que indica las veces que se repite como máximo alguno de los bucles de operaciones. Suponiendo que k sea moderado (que en la práctica lo es) y despreciable frente al resto de factores, podemos hablar de que en global cada iteración tiene un coste $O(l^3 \cdot |\Sigma| \cdot n)$.

1. Entrada: cadena inicial s
2. Hacer $s' = s$
3. Para cada posición i de s
   a) Aplicar todas las posibles sustituciones en la posición i de s y quedarse con la mejor; depositar resultado en s_s
   b) Aplicar el borrado en la posición i de s; depositar resultado en s_b
   c) Aplicar todas las posibles inserciones en la posición i de s y quedarse con la mejor; depositar resultado en s_i
   d) Tomar la mejor cadena entre $\{s, s_s, s_b, s_i\}$ y depositarla en s
4. Si $s' \neq s$ hacer $s' = s$ y volver a 3
5. Resultado: s

Figura 3.5: Esquema básico del método iterativo conjunto para la obtención de la cadena media aproximada.

Queda por tanto claro que este método tiene un coste computacional más elevado que los métodos expuestos en el apartado 1.6.2 o la Sección 2.1, pues eleva en un orden de magnitud el factor asociado a la longitud de las cadenas (pasa de l^2 a l^3) y, además, debe iterar varias veces haciendo ese mismo proceso. Sin embargo, lo que también es evidente es que el espacio de búsqueda se amplia notablemente respecto a los dos métodos presentados previamente, con lo cual es factible encontrar una solución más cercana a la cadena media exacta.

3.1.2. Método iterativo conjunto

Uno de los problemas que discutíamos que tiene el método presentado en el apartado 3.1.1 es la arbitrariedad en el orden de aplicación de las operaciones de edición. Es decir, dicho método aplica en primer lugar todas las posibles sustituciones, luego todos los posibles borrados y, por último, todas las posibles inserciones. Cualquiera podría cuestionar dicho orden y proponer que se hiciera en otro orden, el cual también sería perfectamente cuestionable.

La solución que se plantea a esto sería hacer todas las operaciones a la vez. Es decir, en vez de realizar primero las de un tipo, elegir la nueva cadena candidata y, tras ello, aplicar otro tipo de operación, se trataría de realizar todas las operaciones posibles y, tras evaluar las cadenas resultantes, elegir la nueva candidata. La restricción que se decide no eliminar es la de ir haciéndolo posición a posición.

Siguiendo esta idea, se propone un algoritmo [46] que, posición a posición, va probando todas las posibles sustituciones, borrados e inserciones en dicha posición, escogiendo de entre todas las cadenas resultantes y la cadena previa

aquella cadena que presente menor distancia acumulada a las cadenas de S. El esquema básico del proceso viene dado en la Figura 3.5, y el algoritmo detallado viene descrito en la Figura 3.6.

Como vemos, el algoritmo va posición a posición de la actual cadena candidata (que puede ir cambiando su longitud dentro de una misma iteración global debido a las posibles inserciones). En la posición actual, hace todas las posibles sustituciones y se queda con la cadena que menor distancia acumulada presenta entre todas las generadas (incluyendo la candidata actual). Después genera la cadena producto del borrado en la posición actual, y por último genera las cadenas producto de todas las posibles inserciones en la posición actual (que es *anterior* al símbolo de esa posición), quedándose con aquella que presente menor distancia acumulada. Por último, entre las tres seleccionadas (la de sustitución, la de borrado y la de inserción) y la candidata actual se selecciona la que presenta menor distancia acumulada, y queda como candidata para operar en la siguiente posición de la cadena (que en el caso de aplicar el borrado es la misma que en la iteración previa).

Respecto al coste temporal, al igual que pasaba en el algoritmo iterativo separado presentado en el apartado 3.1.1, en este algoritmo no es conocido *a priori* el número global de iteraciones (aunque estará acotado por una expresión semejante a la que dimos para el método separado), y sólo podemos calcular el coste de cada una de ellas. En cambio, ya no tenemos los bucles internos que se daban en la versión separada. El coste de cada una de estas iteraciones, asumiendo que la medida de disimilitud tiene coste cuadrático (es decir, usamos alguna de las propuestas de la Sección 1.5), viene a ser de orden $O(l^3 \cdot |\Sigma| \cdot n)$, suponiendo que la longitud de la cadena media aproximada no superará l (longitud máxima de las cadenas de S). Esto se debe a que para cada posición, en el caso de sustitución e inserción hay que hacer para los $|\Sigma|$ símbolos posibles el cálculo de la distancia acumulada (n distancias en total).

Así pues, este nuevo algoritmo presenta un coste computacional asintótico idéntico al propuesto previamente, y, por tanto, un orden mayor (respecto a l) que el método constructivo voraz o que el cálculo de la cadena mediana. Sin embargo, de nuevo nos permite explorar un espacio de soluciones más amplio, pudiéndonos así llevar a cadenas medias aproximadas más cercanas a la exacta, y además cuenta con la característica de no escoger arbitrariamente el orden de las operaciones de edición (pues se aplican todas las posibles y después se escoge).

Este espacio de soluciones que se explora es idéntico al del método separado, sólo que se explora de manera distinta. Para ver este hecho, basta con observar que las modificaciones básicas que hace uno de los métodos sobre una cadena s para obtener una cadena s' pueden realizarse de la misma manera con el otro método, empleando en general un número distinto de iteraciones. Por tanto, cualquier cadena obtenible desde una cierta cadena inicial con uno de los métodos es obtenible con el otro método. De todas formas, no podemos concluir si aplicar este nuevo método va a ser más o menos ventajoso con respecto al método separado a la hora de obtener una mejor aproximación, pues sólo el uso experimental del método nos puede clarificar esto.

Entrada: $m \in \Sigma^*$ solución inicial, $S = \{s^1, s^2, \ldots, s^n\}$
Salida: $m \in \Sigma^*$ solución final
Inicio
Hacer
  $d_p = \sum_{i=1}^{n} d(m, s^i)$
  $d_a = d_p$
  Para $i = 1, \ldots, |m|$ /* Para todas las posiciones */
    $m^{sub} = m$
    $sb = \emptyset$
    $d_{sub} = d_a$
    ParaTodo $a \in \Sigma$ /* Aplica sust. para todos los símbolos */
      $m_i^{sub} = a$
      Si $\sum_{j=1}^{n} d(m^{sub}, s^j) < d_{sub}$ **Entonces** /* Hay mejora */
        $sb = a$
        $d_{sub} = \sum_{j=1}^{n} d(m^{sub}, s^j)$
      FSi
    FParaTodo
    Si $sb \neq \emptyset$ **Entonces** $m_i^{sub} = sb$ **Sino** $m^{sub} = m$ **FSi**
    $m^{bor} = m_1 \cdots m_{i-1} m_{i+1} \cdots m_{|m|}$ /* Aplica borrado */
    $m^{ins} = m$
    $sb = \emptyset$
    $d_{ins} = d_a$
    ParaTodo $a \in \Sigma$ /* Aplica ins. para todos los símbolos */
      $m^{ins} = m_1 \cdots m_{i-1} \cdot a \cdot m_i \cdots m_{|m|}$
      Si $\sum_{j=1}^{n} d(m^{ins}, s^j) < d_{ins}$ **Entonces** /* Hay mejora */
        $sb = a$
        $d_{ins} = \sum_{j=1}^{n} d(m^{ins}, s^j)$
      FSi
    FParaTodo
    Si $sb \neq \emptyset$ **Entonces** $m^{ins} = m_1 \cdots m_{i-1} \cdot sb \cdot m_i \cdots m_{|m|}$
      Sino $m^{ins} = m$
    FSi
    $m = \text{argmin}_{s \in \{m, m^{sub}, m^{bor}, m^{ins}\}} \sum_{j=1}^{n} d(s, s^j)$ /* La mejor */
    $d_a = \sum_{j=1}^{n} d(m, s^j)$
  FPara
Mientras $d_a < d_p$ /* Hasta que no hay mejora global */
Devolver m

Figura 3.6: Algoritmo del método iterativo conjunto para la obtención de la cadena media aproximada.

Alineamiento	Interpretación
(a, λ, λ)	Inserción a respecto b y c
(λ, b, λ)	Inserción b respecto a y c
(λ, λ, c)	Inserción c respecto a y b
(a, b, λ)	Sustitución de a por b, borrado de c respecto a y b
(a, λ, c)	Sustitución de a por c, borrado de b respecto a y c
(λ, b, c)	Sustitución de b por c, borrado de a respecto b y c
(a, b, c)	Sustitución de a por b, b por c y a por c

Figura 3.7: Emparejamientos posibles para tres símbolos. Las inserciones tienen su interpretación dual como borrados y viceversa.

Igualmente queda abierta, como en el algoritmo iterativo separado, la cuestión de la cadena inicial sobre la cual se efectúan las perturbaciones. De nuevo los razonamientos aplicados sobre la versión separada son válidos para esta versión, y las cadenas iniciales podrían ser la cadena mediana o la cadena obtenida por el método voraz del apartado 2.1.

3.2. Otros métodos aproximados

Los métodos presentados en la Sección 3.1 se basan en la idea de ir perturbando una cadena candidata a fin de encontrar cadenas que presenten una menor distancia acumulada. Evidentemente, este no es el único mecanismo que se puede tomar para calcular aproximaciones a la cadena media de un conjunto de cadenas. Otro de los métodos más usuales, sobre todo en el campo de la biología (empleados para secuencias de proteínas y cadenas de ADN), es la obtención de alineamientos entre las secuencias [76].

Este alineamiento se corresponde, en el caso de dos cadenas, a la secuencia de operaciones de edición seguida para hallar la distancia entre dichas cadenas. A partir de esta secuencia de edición sería posible obtener una aproximación a la cadena media como aquella que presenta, en el orden de la secuencia, los símbolos de las sustituciones correctas, los de las inserciones/borrados y el símbolo con menor peso de sustitución para los dos símbolos implicados en sustituciones incorrectas.

Esta idea puede extenderse conceptualmente a más de dos cadenas, de manera que cada paso en el camino de edición indicaría el emparejamiento de los símbolos entre varias subcadenas, o su inserción con respecto a otras, o su borrado con respecto a otras, o su incorrecta sustitución con respecto a otras. Para clarificar esto, veamos un pequeño ejemplo con tres símbolos; siendo estos símbolos a emparejar a, b y c, las diversas posibilidades de emparejamiento quedan representadas en la Figura 3.7.

Con esta descripción, puede verse con claridad que desde cada nodo del grafo que represente un símbolo en el alineamiento se puede alcanzar un total

de $2^n - 1$ nodos, siendo n el número de cadenas total a alinear. Evidentemente, al igual que se hace en el caso del algoritmo de Programación Dinámica que calcula el alineamiento entre dos cadenas, el proceso necesario para calcular el alineamiento no tiene que calcular todas estas posibilidades a cada paso, pues algunas ya vienen resueltas en pasos previos. Así, es fácil ver que si para dos cadenas teníamos un coste de $O(l^2)$ siendo l la longitud máxima, se puede definir un algoritmo por Programación Dinámica para la obtención del alineamiento entre n cadenas cuyo coste será de $O(l^n)$.

Claramente, para un número moderado de cadenas ya tendríamos una complejidad temporal prohibitiva, con lo cual la necesidad de aplicar técnicas subóptimas que reduzcan esta complejidad es evidente. Las técnicas de búsqueda en haz [4] o el uso de ventanas de búsqueda parecen las más apropiadas a la hora de optimizar temporalmente este proceso.

Una alternativa a este método se basa en hacer los alineamientos par a par de cadenas, de manera que se obtiene el conjunto de alineamientos par a par óptimos del conjunto inicial y se vuelve a aplicar el proceso (ahora sobre el conjunto de alineamientos) hasta obtener un alineamiento final. Este método es especialmente aplicado en secuencias biológicas y está basado en el hecho de que secuencias homólogas tienen una dependencia evolucionaria, y por tanto es posible obtener un buen alineamiento siguiendo el árbol definido por los sucesivos alineamientos par a par. Este método es realmente un heurístico y no garantiza la solución óptima, pero los trabajos realizados con el mismo muestran un buen comportamiento del método [76].

Existe una aplicación que implementa dicho algoritmo, llamada CLUSTAL W [76], que es de pública disposición. Sin embargo, esta aplicación está limitada ya que no puede tratar con alfabetos generales (sólo admite secuencias de proteínas y de ADN) ni permite extraer de forma automática el mejor alineamiento encontrado (únicamente indica los alineamientos parciales y el árbol que se ha seguido hasta hallar el total de alineamientos), lo cual la hace poco aplicable a nuestro objetivo de obtener aproximaciones a la cadena media de un conjunto de cadenas cualquiera definidas sobre cualquier alfabeto.

3.3. Experimentos comparativos

En esta sección vamos a aplicar las aproximaciones presentadas en la Sección 3.1 para extraer prototipos de diversos agrupamientos y usarlos con clasificadores k-NN. Estos experimentos se realizarán sobre el corpus de cromosomas *Copenhagen*, ya descrito en el apartado 2.2.1.

3.3.1. Comparación entre la cadena mediana y la cadena media aproximada

La primera comparación que vamos a llevar a cabo es entre la cadena mediana, prototipo usado como aproximación clásica a la cadena media, y una de nuestras aproximaciones a la cadena media. En este caso, la aproximación

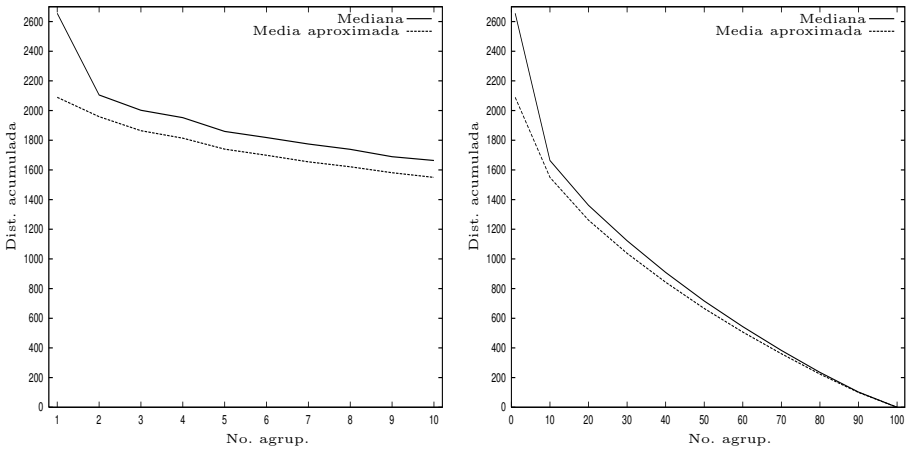


Figura 3.8: Suma de distancias acumuladas de la cadena mediana y la cadena media aproximada a partir de mediana y método iterativo conjunto a sus respectivos agrupamientos, para los intervalos de 1 a 10 y de 1 a 100 agrupamientos

tomada será la que parte de la cadena mediana y aplica posteriormente el proceso iterativo conjunto descrito en el apartado 3.1.2, (es decir, realizando todas las operaciones posibles sobre cada posición y tomando la mejor alternativa) usando la definición clásica de cadena media dada en (1.21) y la distancia de edición normalizada.

En primer lugar vamos a mostrar el valor de la distancia acumulada que se consigue con ambos tipos de prototipos. En las gráficas de la Figura 3.8 se muestra el valor de la suma de las distancias acumuladas de cada prototipo obtenido al agrupamiento que representan. Se ve con claridad que la aproximación escogida para la cadena media presenta, en todos los casos, menor distancia acumulada que la cadena mediana. Esta diferencia va disminuyendo a medida que aumenta el número de agrupamientos. Si las tasas de error en clasificación están correladas con el valor de la distancia acumulada, cabe esperar un comportamiento similar en la clasificación.

Se realizaron experimentos de clasificación usando un clasificador k-NN, con valores de k desde 1 (vecino más cercano) hasta 15. Las clasificaciones se hicieron para cada uno de los agrupamientos obtenidos tal y como está descrito en el apartado 2.2.1, usando una parte como entrenamiento y otra como test e intercambiándolas después a fin de realizar la validación cruzada. Los resultados a nivel de clasificación (con sus respectivos intervalos de confianza del 95 %) para los diversos valores de k se pueden ver en las gráficas de la Figura 3.9 para un número de agrupamientos de 1 a 100 en intervalos de diez y para $k = 1, 6, 12$. Las gráficas de la Figura 3.10 muestran los resultados para un número de agrupamientos de 1 a 10 y valores de $k = 1, 3, 6$.

A la vista de las gráficas, es evidente que la aproximación usada a la cade-

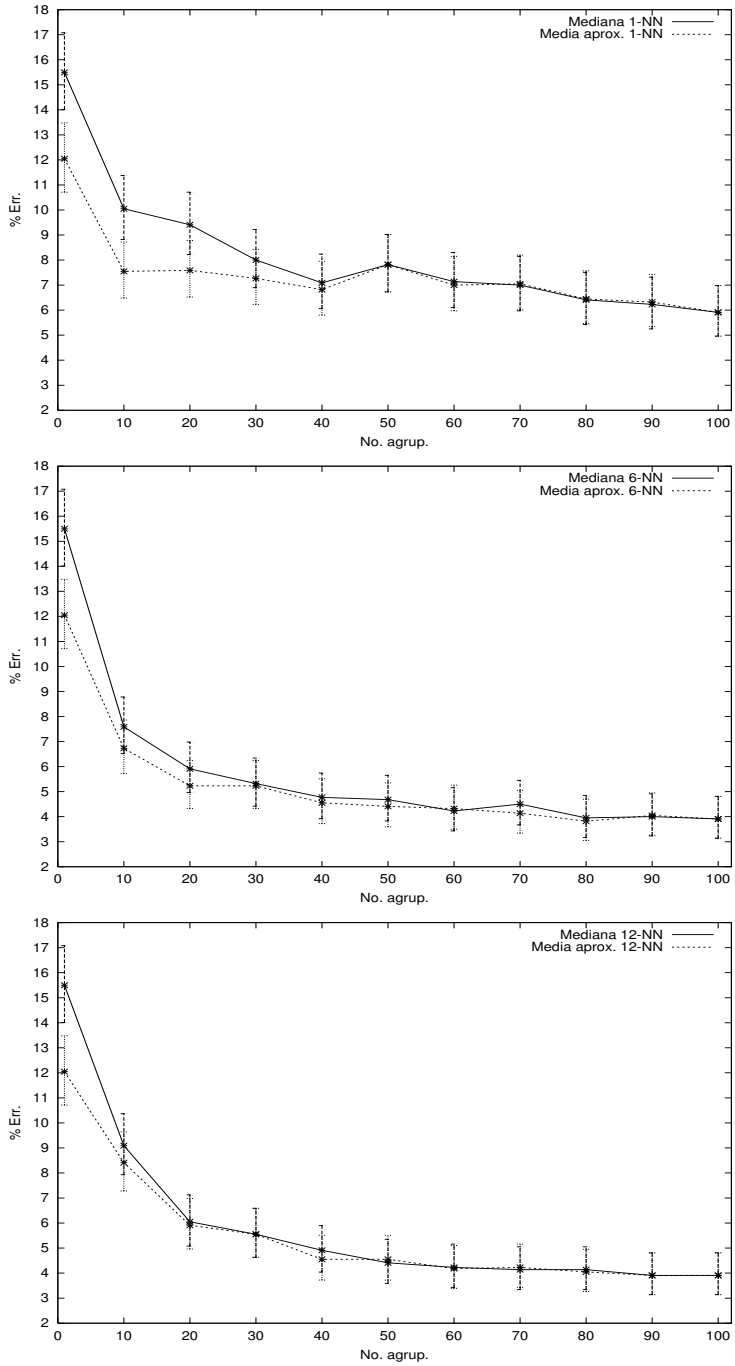


Figura 3.9: Resultados de clasificación usando la cadena mediana y la cadena media aproximada a partir de mediana y método iterativo conjunto para 1, 6 y 12-NN

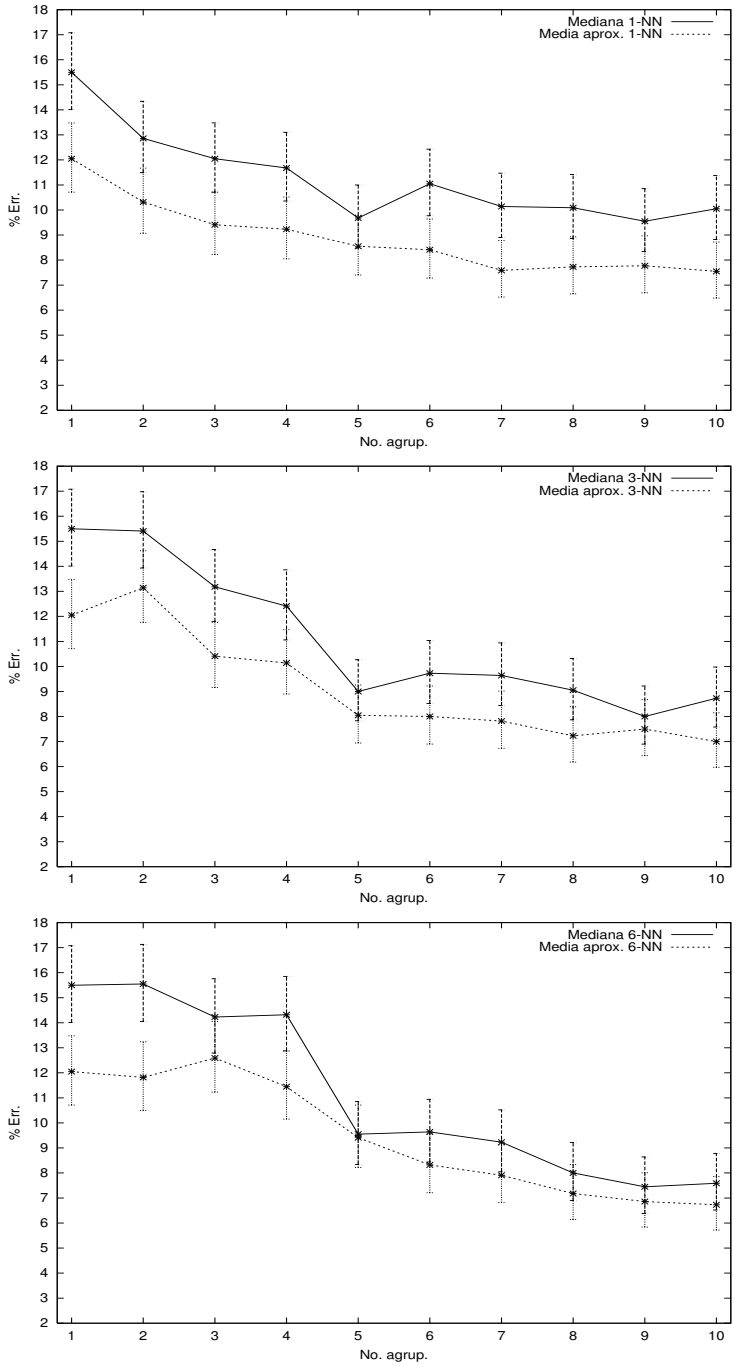


Figura 3.10: Resultados de clasificación usando la cadena mediana y la cadena media aproximada a partir de mediana y método iterativo conjunto para 1, 3 y 6-NN

na media presenta un comportamiento mucho mejor que la cadena mediana a nivel de clasificación. Además, es muy evidente la amplia correlación que existe entre el valor de la distancia acumulada y los resultados de clasificación, lo que viene a confirmar la idea intuitiva inicial. También resulta claro que, a medida que el número de agrupamientos (y por tanto el número de prototipos) va aumentando, estas diferencias se van reduciendo hasta un nivel despreciable. En concreto, para 1-NN se aprecia que a partir de 20 agrupamientos, los intervalos de confianza presentan un solapamiento moderado, haciendo ya despreciables estas diferencias. Este efecto también se da con el aumento del número de vecinos k usado en el clasificador. Se deduce de las gráficas que a mayor k el umbral (número de agrupamientos a partir del cual las diferencias son poco significativas) disminuye, pues puede verse que ya con $k = 6$ los intervalos de confianza presentan un solapamiento severo.

En cambio, para un número de agrupamientos pequeño (entre 1 y 10) las diferencias resultan, en su mayor parte, significativas. Para el caso de vecino más próximo, los solapamientos entre los intervalos de confianza son nulos o moderados (excepto para el caso de 5 agrupamientos, donde presenta una irregularidad que veremos que se repetirá en el resto de experimentos), y con el aumento del número de vecinos del clasificador estos solapamientos se hacen más acusados, indicando que las diferencias ya no son tan significativas. En todo caso, sí se siguen apreciando diferencias significativas para un número muy reducido de agrupamientos (de 1 a 4). Esto nos puede llevar a concluir que nuestra aproximación es competitiva respecto a la cadena mediana cuando se usa un número de agrupamientos pequeño y, además, el número de vecinos k usado en el clasificador es también reducido.

En el Apéndice B se presentan resultados complementarios obtenidos sobre un corpus de mayor tamaño utilizando estos tipos de prototipos. Los resultados mostrados allí confirman las conclusiones obtenidas para el corpus *Copenhagen*.

3.3.2. Comparación entre los diversos métodos de inicialización

Como describimos en la Sección 3.1, los métodos iterativos propuestos para aproximar la cadena media necesitan una cadena inicial sobre la cual realizar las perturbaciones. Aunque dicha cadena podría ser cualquiera, la propuesta era usar la cadena mediana o la obtenida por el proceso constructivo voraz. Esta propuesta de usar estas inicializaciones puede verse cuestionada por las significativas diferencias mostradas en el apartado 2.2.2. Por tanto, es evidente que hay que estudiar la posible influencia de cada una de dichas inicializaciones sobre los procesos de aproximación, a fin de verificar si una de las combinaciones presenta diferencias significativas con respecto a la otra o si los procesos son lo suficientemente robustos como para soslayar estas diferencias iniciales de las que se parte.

En las gráficas de la Figura 3.11 presentamos los resultados obtenidos usando un clasificador k-NN con $k = 1, 6, 12$. Para la optimización se usa el método conjunto, y la comparación se lleva a cabo entre la inicialización con la cadena

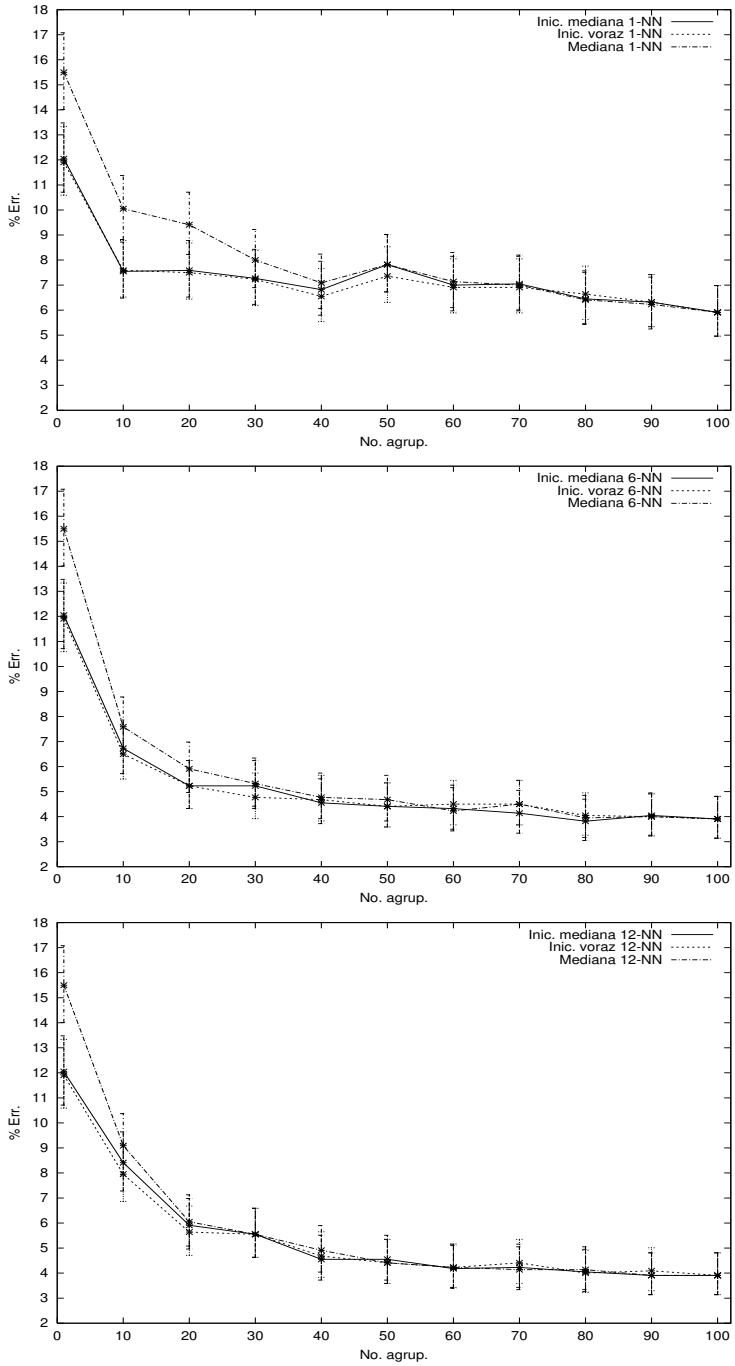


Figura 3.11: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la distinta inicialización, usando método iterativo conjunto para 1, 6 y 12-NN

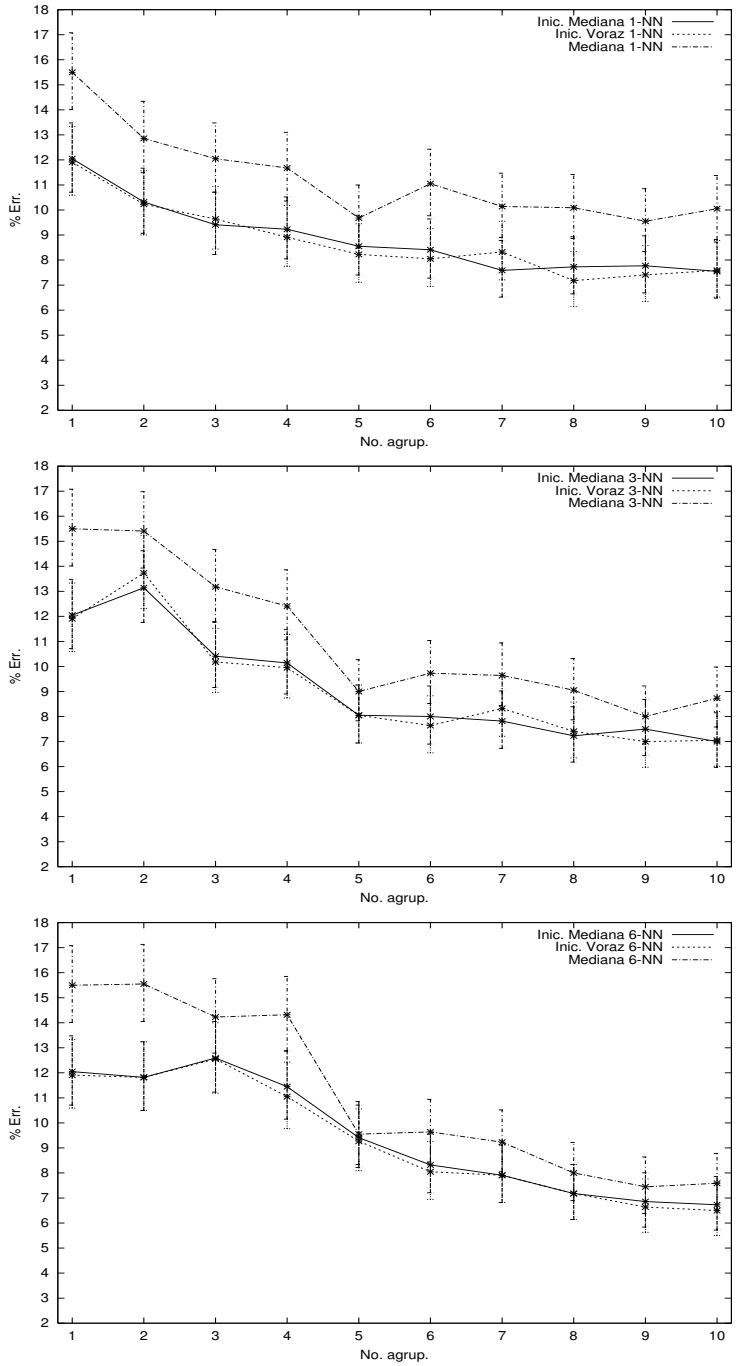


Figura 3.12: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la distinta inicialización, usando método iterativo conjunto para 1, 3 y 6-NN

mediana o con la cadena resultado del proceso voraz. El detalle para un menor número de agrupamientos con ambas inicializaciones (en las mismas condiciones) se puede ver en las gráficas de la Figura 3.12, esta vez para valores de $k = 1, 3, 6$.

En el Apéndice A, en la Sección A.1, se presentan los resultados (sin intervalos de confianza, pues resultan muy similares a los ya expuestos) para el método separado (Figuras A.1 y A.2), usando el mismo número de vecinos.

De las gráficas se puede deducir que apenas existen diferencias significativas entre usar un tipo de cadena inicial u otra (el solapamiento entre los intervalos de confianza es casi total en la mayoría de los casos). Las diferencias más significativas se dan puntualmente para cierto número de agrupamientos y cierto valor de k, siendo en estos casos a veces favorable a la inicialización por cadena mediana y en otros casos a la inicialización constructiva voraz. El comportamiento es similar sea cual sea el algoritmo de aproximación empleado (conjunto o separado)

Por tanto, parece concluirse que ambas alternativas de inicialización carecen de influencia en la obtención de la aproximación a la cadena media, o que en todo caso nuestros procesos de optimización son lo suficientemente robustos de cara a la inicialización usada (es decir, no son sensibles a la cadena candidata inicial).

3.3.3. Comparación entre los diversos métodos de optimización

Otra de las variables a considerar es el propio proceso de optimización empleado para la obtención de la cadena media aproximada. En la Sección 3.1 proponíamos dos métodos, llamados iterativo separado e iterativo conjunto, que son los que vamos a comparar. Recordemos que el iterativo separado aplicaba primero todas las sustituciones posibles, luego todos los borrados, después de nuevo las sustituciones y por último las inserciones, mientras que el iterativo conjunto aplicaba los tres tipos de operaciones sobre una posición y elegía. En las gráficas de la Figura 3.13 vemos los resultados de clasificación con la cadena mediana y cada una de las dos aproximaciones para $k = 1, 6, 12$, usando la cadena mediana como cadena candidata inicial. El detalle para un número de agrupamientos de 1 a 10 se muestra en las gráficas de la Figura 3.14 para $k = 1, 3, 6$.

De los resultados mostrados se puede deducir que, igual que en el caso de la inicialización, el proceso de optimización usado no provoca diferencias significativas en los resultados de clasificación (de nuevo los intervalos de confianza presentan un grado de solapamiento muy alto). Ocasionalmente hay diferencias en los resultados de clasificación en combinaciones puntuales del número de agrupamientos y el valor de k, pero no es posible concluir que, en general, uno de los métodos dé mejores prototipos que el otro. Los resultados con la inicialización voraz (mostrados en el Apéndice A, Figuras A.3 y A.4), vienen a confirmar estas conclusiones para esta otra alternativa de inicialización.

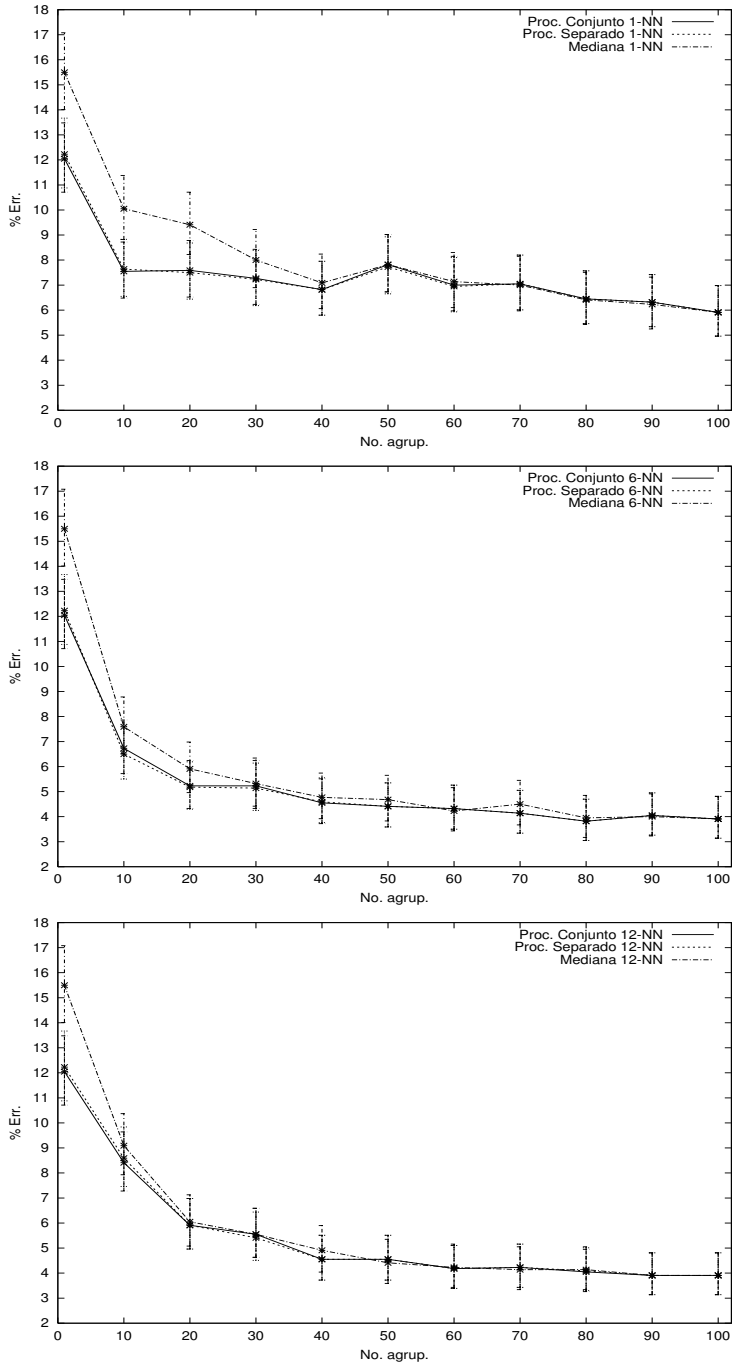


Figura 3.13: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según los distintos procesos de optimización, usando la cadena mediana como inicialización, para 1, 6 y 12-NN

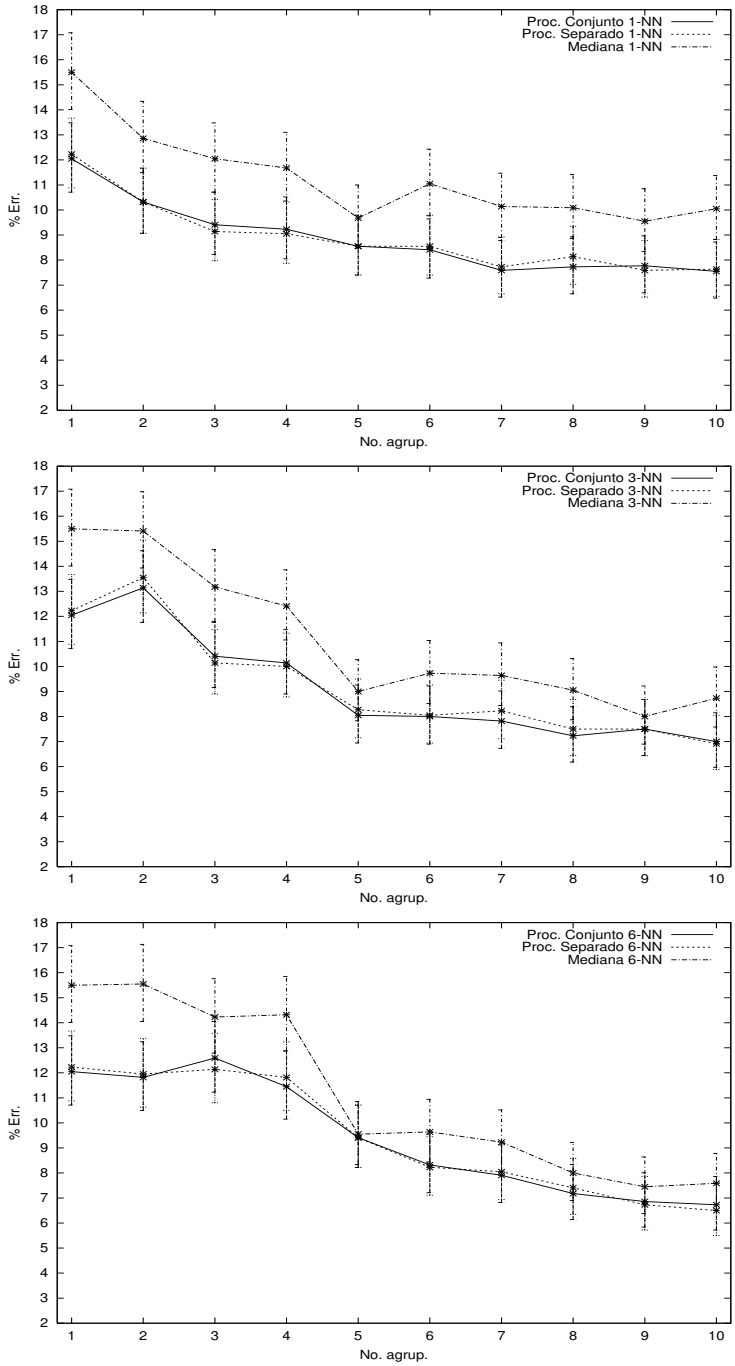


Figura 3.14: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según los distintos procesos de optimización, usando la cadena mediana como inicialización, para 1, 3 y 6-NN

3.4. Resumen

En este capítulo hemos presentado dos aproximaciones a la cadena media, pues al ser la obtención de la misma un problema NP-Duro es necesario buscar soluciones aproximadas mediante métodos heurísticos. Se han presentado dos métodos iterativos (separado y conjunto). Estos métodos iterativos se basan en perturbación sistemática a partir de una cadena inicial. También se ha hecho un breve repaso sobre métodos de obtención de aproximaciones a la cadena media no basados en perturbación de una cadena inicial. Para verificar la idoneidad de las aproximaciones propuestas frente a la cadena mediana, se han realizado exhaustivos experimentos sobre el corpus de cromosomas *Copenhagen*, estudiando la influencia de las diversas variantes de las aproximaciones propuestas (según inicialización y método usado). Estos resultados demuestran la mayor competencia de las aproximaciones frente al uso de la cadena mediana y, a su vez, que no existen notables diferencias entre las diversas alternativas.

Capítulo 4

Una definición alternativa de la cadena media

En este capítulo proponemos una definición alternativa de cadena media, distinta a la propuesta en el apartado 1.6.1, y que subsana ciertos problemas de dicha definición. Posteriormente, se dan resultados experimentales que comparan los diferentes resultados que se producen usando una definición u otra.

4.1. Cadena media cuadrática

La definición propuesta por Kohonen, introducida en el apartado 1.6.1, presenta una característica particular en el caso de buscar la media entre dos cadenas s y t. Usando la definición dada en (1.21), dentro de las posibles soluciones para la cadena media de $S = \{s, t\}$ estarían incluídas las propias s y t, lo cual va contra el concepto intuitivo de cadena media (la cadena que, en promedio, está más cercana a todas las cadenas de S).

Para dar un ejemplo, supongamos $\Sigma = \{a, b\}$, $s = ab$ y $t = ba$, usando como medida de disimilitud la distancia de edición clásica. Con dicha definición tendríamos que las cadenas a, b, aa, ab, ba, bb, aba y bab satisfacen la condición de cadena media para el conjunto $S = \{s, t\}$. Sin embargo, es plenamente intuitivo que ab y ba no son un buen ejemplo de cadena media, ya que son precisamente los ejemplos extremos de lejanía a la otra cadena dentro del conjunto de muestras. Una elección más razonable sería aba o bab.

Esta particularidad de la definición clásica de la cadena media es la que lleva a dar una definición alternativa a la misma, que solventa este problema particular para conjuntos de dos cadenas. Dicha definición, propuesta en [47], se inspira en la definición del vector media expuesta en la Ecuación (1.5).

En la definición de vector media, la distancia se basa en sumar las diferencias al cuadrado entre cada componente del vector. Este criterio, en espacios métricos, puede generalizarse para un orden m cualquiera, dando pie a la distancia de Minkowsky [82]. Esta distancia entre los vectores $\mathbf{x}$ y $\mathbf{y}$ se define para un

cierto orden m como:

$$d_M = \left(\sum_{i=1}^{n} |x_i - y_i|^m \right)^{\frac{1}{m}} \tag{4.1}$$

El valor de m a usar depende de la aplicación que se le tenga que dar a la distancia, ya que presenta la propiedad de enfatizar, a medida que el valor de m aumenta, las mayores diferencias entre las componentes de los vectores [82]. A medida que m tiende a infinito, la distancia de Minkowsky converge a la distancia de Chebyshev [82], que vale la máxima diferencia entre componentes. En [82] también se nombra que es poco usual usar distancias de Minkowsky con $m > 2$.

Siguiendo esta idea, la definición alternativa de cadena media que proponemos sustituye el uso de la medida de disimilitud por el de la medida de disimilitud al cuadrado. Así, dado un alfabeto Σ y su monoide libre asociado Σ^*, dada d una medida de disimilitud sobre Σ^* y dado $S = \{s^1, s^2, \ldots, s^n\} \subset \Sigma^*$, la cadena media de S según esta nueva propuesta, m_S viene definida por:

$$m_S = \underset{s \in \Sigma^*}{\operatorname{argmin}} \sum_{i=1}^{n} d(s, s^i)^2 \tag{4.2}$$

a la cual llamaremos cadena media cuadrática.

Se puede observar que elevar la distancia al cuadrado no modifica las propiedades de la definición de la medida de disimilitud. Es decir, si d cumple las propiedades de ser una medida de disimilitud, d^2 también lo será, pues:

- $d(s,s) = 0 \Leftrightarrow d(s,s)^2 = 0$
- $d(s,t) \geq 0 \Rightarrow d(s,t)^2 \geq 0$
- $d(s,t) = d(t,s) \Rightarrow d(s,t)^2 = d(t,s)^2$

Obsérvese en cambio que la desigualdad triangular no está garantizada (como contraejemplo, tenemos que $3+3 \geq 5$ pero $3^2+3^2 = 9+9 = 18 \ngeq 5^2 = 25$). Sin embargo, es posible demostrar para el caso de distancia de edición normalizada que cumpla la desigualdad triangular, se puede añadir una holgura h [32] para que la versión cuadrática la cumpla:

Proposición 1 *Dada la distancia de edición normalizada sobre una cierta matriz de pesos W, con peso máximo W_{MAX}, de manera que con dichos pesos la distancia de edición normalizada cumple la desigualdad triangular, para una holgura $h = W_{MAX}$ se cumple la desigualdad triangular para la versión cuadrática.*

Demostración: Dadas tres cadenas $s, t, r \in \Sigma^*$ y la distancia de edición normalizada con W_{MAX} peso máximo de una operación de edición, estamos buscando la holgura h tal que se cumpla:

$$(d(s,t)+h)^2 + (d(t,r)+h)^2 \geq (d(s,r)+h)^2$$

Desarrollando los cuadrados, esto nos queda:

$$d(s,t)^2 + h^2 + 2hd(s,t) + d(t,r)^2 + h^2 + 2hd(t,r) \geq d(s,r)^2 + h^2 + 2hd(s,r) \rightarrow$$

$$d(s,t)^2 + d(t,r)^2 + h^2 + 2h(d(s,t) + d(t,r)) \geq d(s,r)^2 + 2hd(s,r)$$

Si ahora tomamos $h = W_{MAX}$, nos quedará:

$$d(s,t)^2 + d(t,r)^2 + W_{MAX}^2 + 2W_{MAX}(d(s,t) + d(t,r)) \geq d(s,r)^2 + 2W_{MAX}d(s,r) \rightarrow$$

$$d(s,t)^2 + d(t,r)^2 - d(s,r)^2 + W_{MAX}^2 + 2W_{MAX}(d(s,t) + d(t,r) - d(s,r)) \geq 0$$

Para la distancia de edición normalizada se da que $0 \leq d(s', s'') \leq W_{MAX}$ para $s', s'' \in \Sigma^*$ [53], con lo cual tendremos que $d(s,t)^2 + d(t,r)^2 - d(s,r)^2 \geq -W_{MAX}^2$. Por tanto:

$$d(s,t)^2 + d(t,r)^2 - d(s,r)^2 + W_{MAX}^2 + 2W_{MAX}(d(s,t) + d(t,r) - d(s,r)) \geq$$

$$-W_{MAX}^2 + W_{MAX}^2 + 2W_{MAX}(d(s,t) + d(t,r) - d(s,r)) =$$

$$2W_{MAX}(d(s,t) + d(t,r) - d(s,r))$$

Y si d cumple la desigualdad triangular, tenemos que $d(s,t) + d(t,r) - d(s,r) \geq 0$, con lo cual:

$$d(s,t)^2 + d(t,r)^2 - d(s,r)^2 + W_{MAX}^2 + 2W_{MAX}(d(s,t) + d(t,r) - d(s,r)) \geq$$

$$2W_{MAX}(d(s,t) + d(t,r) - d(s,r)) \geq 0$$

Y se cumple la desigualdad triangular.
□

El uso de la distancia cuadrática garantiza que situaciones como las que ocurrían con la definición dada en (1.21) para dos cadenas no se van a producir, tal como establece la siguiente proposición:

Proposición 2 *Dado el conjunto de cadenas $S = \{s, t\}$, con $s \neq t$, si la medida de distancia usada cumple la desigualdad triangular, entonces la cadena media cuadrática de S, m_S, cumple que $m_S \neq s$ y $m_S \neq t$.*

Demostración: Si la distancia entre s y t vale $d_{s,t}$ y la medida de distancia usada cumple la desigualdad triangular, la distancia acumulada de cualquiera de las otras candidatas a cadena media también valdrá $d_{s,t}$, ya que esta es la mínima distancia posible. Si m es una de estas candidatas, tendremos que $d(m,s) = d'$. Por tanto, $d(m,t) = d_{s,t} - d'$.

Si tomáramos $m = s$ o $m = t$ como solución usando la definición cuadrática, tendríamos una distancia acumulada de $d_{s,t}^2 + 0^2 = d_{s,t}^2$.

Si tomamos que m es cualquiera de las otras candidatas se tendría una distancia acumulada de $d(m,s)^2 + d(m,t)^2 = d'^2 + (d_{s,t} - d')^2 = d_{s,t}^2 + 2d'^2 - 2d_{s,t}d'$.

Cuadro 4.1: Conjuntos de 3 y 4 cadenas en los que la cadena media clásica y cuadrática usando la distancia de edición normalizada difieren

Conjunto de datos	Clásica	$\sum d$	$\sum d^2$	Cuadrática	$\sum d$	$\sum d^2$
{aba, bbc, baa}	baba	2'20	1'94	bba	2'33	1'89
{aba, bbc, baa, caa}	baa	3'17	3'69	baba	3'20	2'94

γ	a	b	c	λ
a	0	2	3	2
b	2	0	2	2
c	3	2	0	2
λ	2	2	2	-

Figura 4.1: Matriz de pesos usada para las pruebas del Cuadro 4.1

Es claro que $d_{s,t}^2 + 2{d'}^2 - 2d_{s,t}d' < d_{s,t}^2$, ya que $d' < d_{s,t}$ y, consecuentemente, $2{d'}^2 - 2d_{s,t}d' < 0$. Por tanto, s o t no alcanzan la distancia acumulada mínima con la definición cuadrática y no pueden escogerse como cadena media.
□

Si aplicamos la definición de cadena media cuadrática al ejemplo previo de $s = ab$ y $t = ba$, vemos ahora que las cadenas que cumplen la condición son a, b, aa, bb, aba y bab, quedando las propias s y t excluídas, tal y como habíamos demostrado.

Aunque inicialmente esta definición se ha propuesto para solventar el problema con dos cadenas, existen ejemplos que aseveran que para conjuntos de más de dos cadenas también se obtienen resultados distintos. En el Cuadro 4.1 se muestran las cadenas medias obtenidas usando la definición clásica y cuadrática, con la distancia de edición normalizada, para los conjuntos de cadenas presentados. La matriz de pesos entre símbolos es la presentada en la Figura 4.1.

La cadena media cuadrática presenta, de todas maneras, el mismo problema computacional que la definición clásica: encontrar la solución exacta es un problema NP-Duro. Esto se debe a que el único cambio que se está haciendo es usar otra medida de distancia, con lo cual el problema es idéntico al original y tiene la misma complejidad. Por tanto, de nuevo es necesario la búsqueda de aproximaciones a esta cadena media cuadrática.

Repasando los algoritmos de aproximación presentados en la Sección 3.1, la incorporación de la definición cuadrática es tan simple como sustituir el cálculo de la distancia acumulada por el cálculo acumulativo de distancias al cuadrado. Esta modificación no altera el coste del algoritmo, pues lo único que hay que hacer es obtener la distancia de la manera habitual y luego elevar ese valor al cuadrado antes de acumularlo.

Ya hemos visto que usar la cadena media cuadrática proporciona resultados

distintos en ciertos conjuntos de datos, y estas diferencias también pueden presentarse a la hora de obtener las distintas aproximaciones. Esto se debe a que la alteración en el proceso de optimización puede variar el resultado final.

Queda ahora comprobar el efecto que tiene en la calidad de los prototipos obtenidos usar esta nueva definición de cadena media. Así, la Sección 4.2 va dedicada a comparar las clasificaciones con los diversos prototipos obtenidos usando ambas definiciones (clásica y cuadrática) para el corpus *Copenhagen*.

4.2. Experimentos comparativos

En esta sección vamos a aplicar ambas definiciones de cadena media, la clásica, dada en (1.21), y la cuadrática, dada en (4.2), para extraer prototipos de diversos agrupamientos y usarlos con clasificadores k-NN. Estos experimentos se realizarán sobre el corpus de cromosomas que se describe en el apartado 2.2.1.

La finalidad de realizar estos experimentos es comparar el efecto que tiene usar una definición u otra en la extracción de prototipos mediante los algoritmos propuestos en la Sección 3.1, ya que al ser métodos aproximados, dicha definición influye en el prototipo extraído independientemente del tamaño del conjunto de cadenas tratado.

En primer lugar, vamos a verificar que las distintas funciones objetivo usadas se optimizan como se espera. Para ello, mostramos en las gráficas de la Figura 4.2 el valor de la suma de distancias acumuladas, normal y cuadrática, para la cadena mediana y las medias aproximadas con inicialización mediana y proceso conjunto, haciendo uso de las definiciones clásica y cuadrática. Como se puede apreciar, las diferencias a nivel de distancia acumulada entre el prototipo clásico y el cuadrático son prácticamente nulas, con lo cual parece concluirse que es indiferente el criterio de optimización empleado con respecto a este índice. Sí que existe diferencia cuando se comparan a nivel de distancia acumulada cuadrática (las dos gráficas inferiores de la Figura 4.2), pero sólo para un número de agrupamientos alto (a partir de 30 agrupamientos). Esto se debe a que es entonces cuando surgen agrupamientos con pocas cadenas, en los cuales la cadena media cuadrática optimiza adecuadamente el criterio de distancia cuadrática, algo que la cadena media clásica no consigue. De todas maneras, las diferencias resultan tan inapreciables que puede esperarse una influencia escasa en la clasificación al usarse una u otra definición.

Por tanto, se realizaron experimentos de clasificación usando las cadenas medias aproximadas obtenidas usando la definición clásica y la cuadrática. Los resultados de clasificación con cadena mediana y las aproximaciones (usando el método conjunto y la cadena mediana como cadena inicial) usando ambas definiciones se presentan en las gráficas de la Figura 4.3 para un número de agrupamientos de 1 a 100 y $k = 1, 6, 12$. Para un número de agrupamientos entre 1 y 10 y $k = 1, 3, 6$ se presentan los resultados en las gráficas de la Figura 4.4.

En las gráficas se puede observar que en ciertos casos generales, y a nivel absoluto, la distancia cuadrática consigue mejores prototipos que la distancia clásica (por ejemplo, para la regla del vecino más cercano, primera gráfica de la

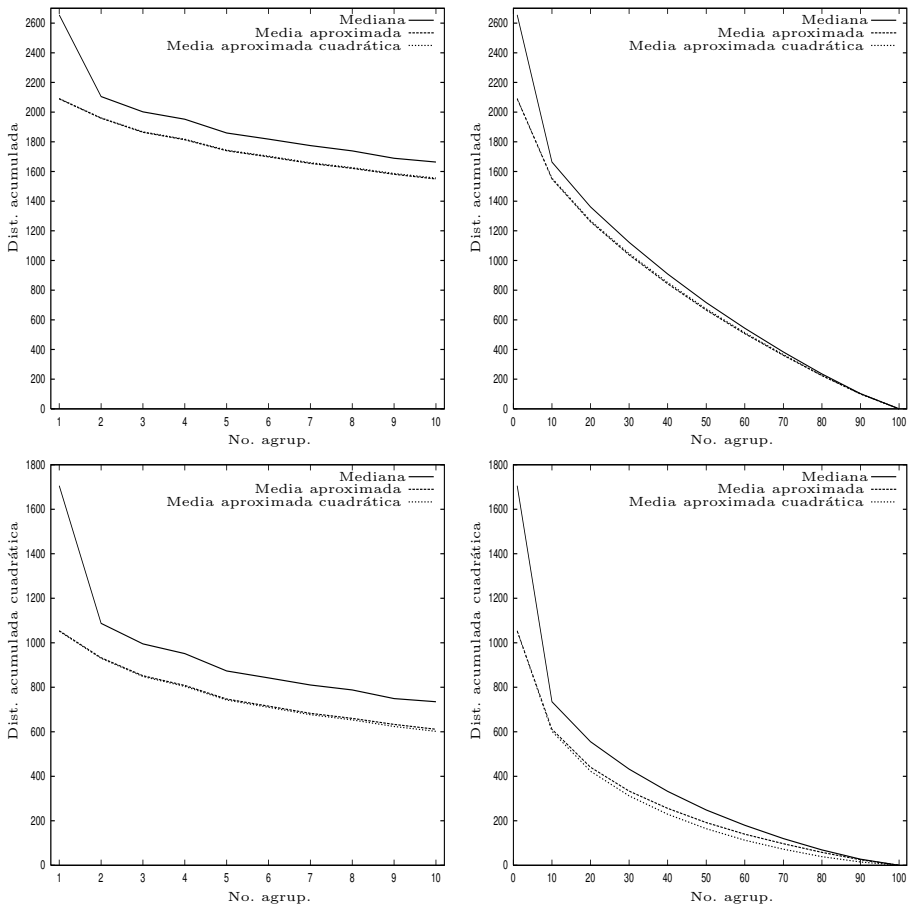


Figura 4.2: Suma de distancias acumuladas (arriba) y distancias acumuladas cuadráticas (abajo) de la cadena mediana y la cadena media aproximada, en sus versiones clásica y cuadrática, con inicialización mediana y método iterativo conjunto, a sus respectivos agrupamientos, para los intervalos de 1 a 10 y de 1 a 100 agrupamientos

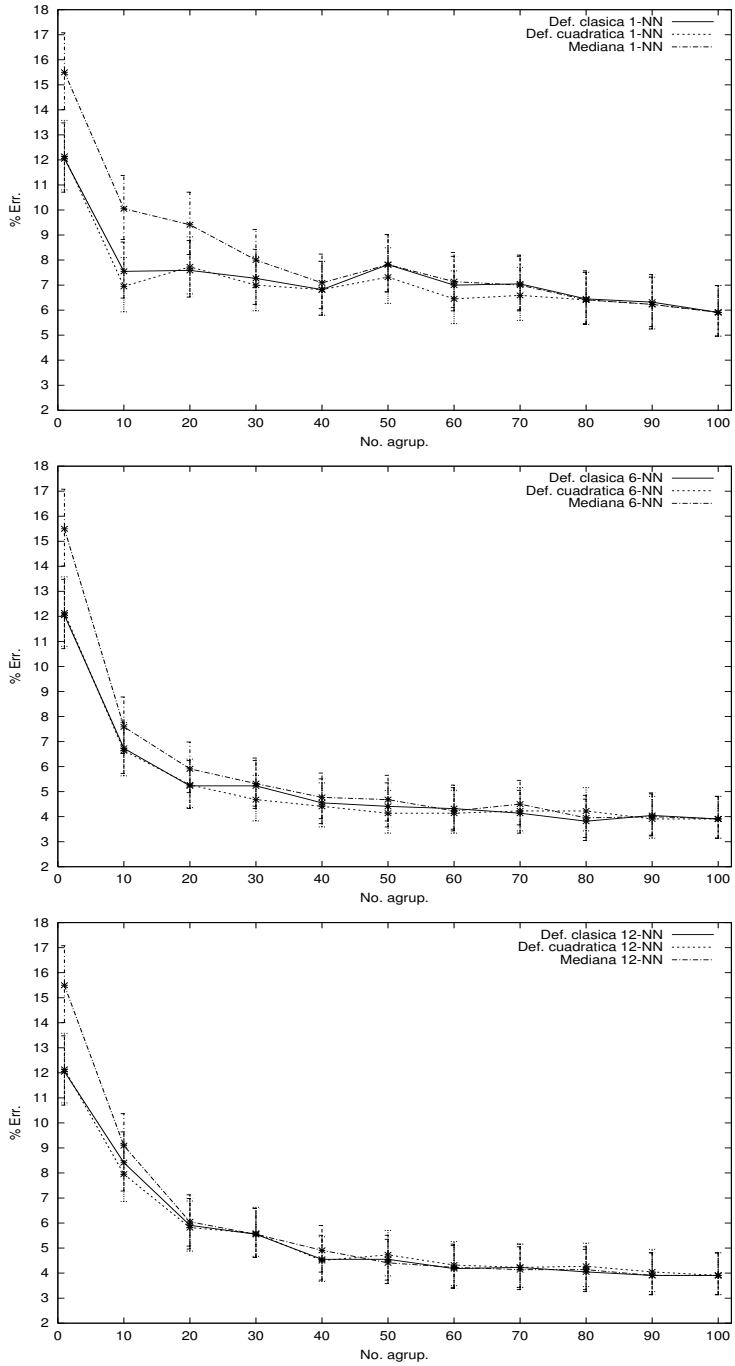


Figura 4.3: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la definición de cadena media usada, usando la cadena mediana como inicialización y el método conjunto, para 1, 6 y 12-NN

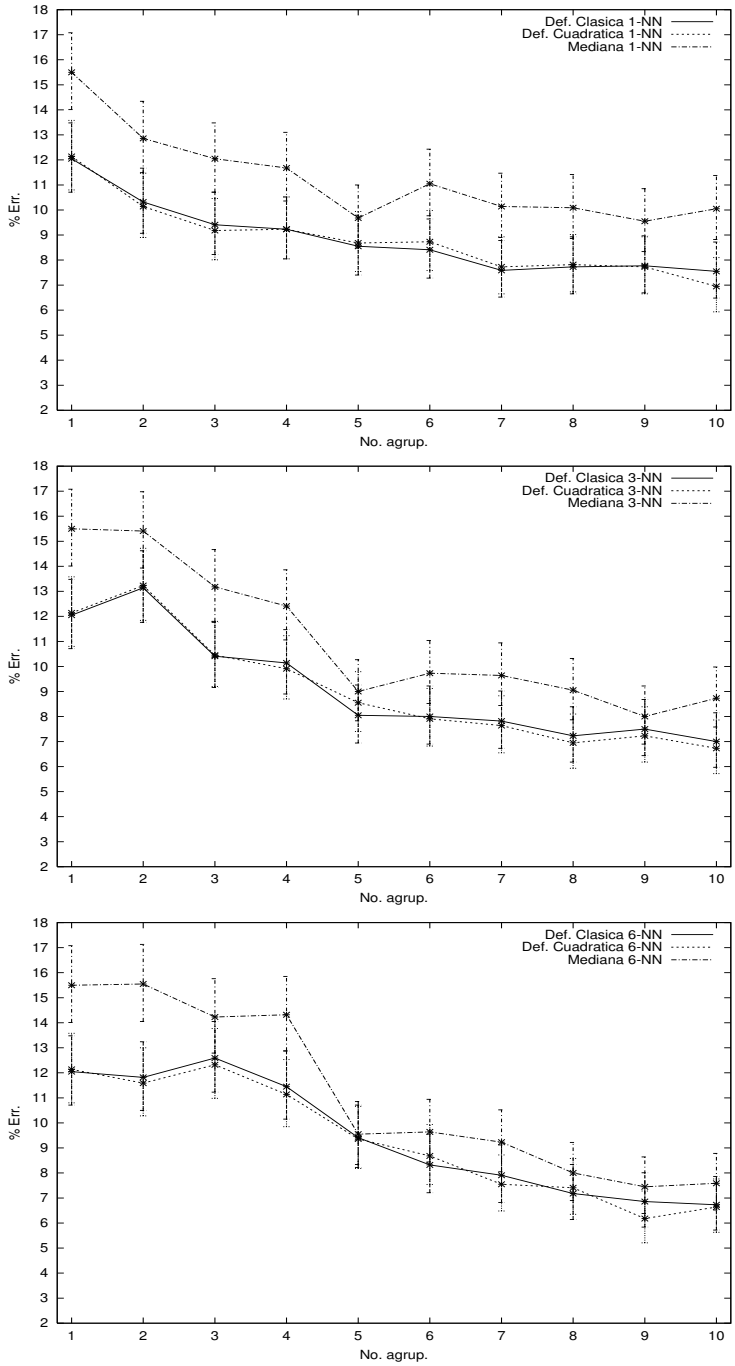


Figura 4.4: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la definición de cadena media usada, usando la cadena mediana como inicialización y el método conjunto, para 1, 3 y 6-NN

Figura 4.3, parece presentar mejor comportamiento al aumentar el número de agrupamientos). Pero en general las diferencias no son realmente significativas, tal y como se esperaba, ya que los intervalos de confianza presentan un grado de solapamiento muy alto (algo que es más acusado aún cuando el número de agrupamientos es más pequeño, como se ve en la primera gráfica de la Figura 4.4).

A medida que aumenta el número de vecinos empleado en el clasificador estas diferencias se van haciendo aún menos significativas, quedando las más destacadas reducidas a zonas más localizadas. Por tanto, no parece que haya diferencias claras entre usar una definición u otra de cadena media.

A fin de verificar estas conclusiones para inicialización voraz o método separado, también se realizaron experimentos comparativos con dichas variantes, cuyos resultados se muestran en el Apéndice A (Figuras A.5 y A.6 para inicialización por mediana y proceso separado, Figuras A.7 y A.8 para inicialización voraz y proceso conjunto y Figuras A.9 y A.10 para inicialización voraz y proceso separado). Los resultados obtenidos resultan semejantes a los discutidos para inicialización por cadena mediana y proceso de optimización conjunto.

Todas estos resultados nos llevan a concluir que, en principio, usar una definición u otra de cadena media no va a tener excesiva influencia en nuestro proceso de aproximación a nivel de clasificación. Es decir, aunque los prototipos que se van a obtener usando una definición u otra serán en muchos casos distintos (el índice a optimizar es diferente en cada caso), en promedio la calidad de los prototipos será semejante. Por tanto, puede concluirse que no es determinante usar una definición u otra de cadena media.

4.3. Resumen

En este capítulo hemos ofrecido una definición alternativa de cadena media, basada en la suma cuadrática, que resuelve algunas características problemáticas de la definición clásica en algunos tipos de agrupamientos. Como dicha definición tiene influencia en el proceso de optimización que se realiza para la obtención de las aproximaciones, se ha realizado una experimentación exhaustiva con el corpus de cromosomas *Copenhagen* a fin de verificar la influencia de esta nueva definición en la obtención de prototipos. Dichos experimentos han confirmado que no existen diferencias significativas entre usar una u otra definición de cadena media en estos procesos.

Capítulo 5

Reducción del coste temporal de las aproximaciones

En el Capítulo 3 hemos descrito métodos para obtener aproximaciones en un tiempo razonable a la cadena media. Aun así, estos métodos continúan siendo costosos, por lo cual en este capítulo se describen técnicas que permiten reducir el coste temporal de dichos métodos a costa de la precisión en la aproximación. La validez de los métodos se prueba mediante experimentación con el corpus descrito en el apartado 2.2.1.

5.1. Técnicas generales de reducción de coste

Existen varias técnicas, aplicables a los algoritmos que usan cálculos de distancia de edición (entre ellos, los descritos en el Capítulo 3), que permiten reducir el coste computacional de hallar una solución, en muchos casos a costa de perder precisión en la misma.

Una de estas técnicas puede emplearse en el cálculo de la distancia de edición (en cualquiera de sus versiones) entre cadenas, que es una parte costosa dentro del algoritmo (orden cuadrático). Es la técnica conocida como búsqueda en haz (o por su término anglosajón *beam search*) [4], y se basa en no explorar exhaustivamente todos los caminos dentro del grafo de Programación Dinámica, sino que se descartan aquellos cuyo peso acumulado sobrepasa un cierto umbral. Esta técnica hace que la distancia calculada no sea la óptima, sino un valor subóptimo superior (pues se pueden eliminar caminos que lleven al valor óptimo). Es una optimización implementable en cualquiera de los algoritmos de perturbación iterativa expuestos en la Sección 3.1, pero no ha sido implementada y ni se ha estudiado su posible influencia en la calidad y tiempo de obtención de los prototipos.

En la línea de la técnica de la búsqueda en haz, otra técnica es la conocida como búsqueda en ventana. En dicha técnica, se define que los caminos válidos son aquellos que transcurren por un subconjunto de aristas del grafo tales que no están alejadas más de una cierta distancia (ventana) del camino más "diagonal" posible en el grafo de Programación Dinámica. Así, el número de caminos a explorar se reduce considerablemente, pero de nuevo se pueden perder soluciones óptimas. Esta técnica es adecuada cuando las cadenas a comparar son bastante semejantes, ya que en ese caso sí es frecuente que el camino óptimo transcurra en las proximidades del camino más "diagonal". Al igual que la búsqueda en haz, se puede aplicar a cualquiera de los algoritmos de optimización presentados aunque tampoco se ha realizado su implementación.

Otra optimización evidente es no hacer el cálculo completo de la distancia de edición acumulada. Es decir, si hasta ahora tenemos que nuestra solución presenta una distancia acumulada de valor *dist*, al llegar la suma de distancias acumuladas para la nueva candidata a un valor superior a *dist* se puede descartar dicha candidata, sin necesidad de hacer el sumatorio completo para las n cadenas de S. De nuevo es una técnica aplicable a cualquiera de los algoritmos presentados y ha sido implementada en la realización de todos los experimentos de esta tesis.

Estas técnicas descritas son de uso general para cualquier algoritmo que implique cálculo de distancias de edición y distancias acumuladas. En las siguientes secciones propondremos, en cambio, técnicas específicas para los algoritmos presentados en el Capítulo 3 y que han sido implementadas y probadas experimentalmente.

5.2. Método de la división

Si observamos de nuevo la complejidad temporal de los métodos iterativos propuestos, que es $O(l^3 \cdot |\Sigma| \cdot n)$, queda claro que el factor más determinante en dicho coste es la longitud de las cadenas de S. Por tanto, toda actuación encaminada a reducir dicha longitud comportará una reducción del coste temporal (aunque no a nivel asintótico) del algoritmo.

Así, surge la idea de dividir las cadenas de S en varias subcadenas de aproximadamente la misma longitud, y luego computar la cadena media aproximada de cada uno de esos subconjuntos de subcadenas. La aplicación de los algoritmos iterativos a este subconjunto será mucho menos costosa temporalmente, pues la longitud de las cadenas se habrá reducido un orden igual al número de divisiones efectuado. La cadena media aproximada para S sería una concatenación de las cadenas medias aproximadas obtenidas para cada división.

Por tanto, este método consiste realmente en un preproceso que toma S y crea D conjuntos de cadenas $S^1, S^2, \ldots, S^D$, formados cada S^i por las subcadenas producto de la división en D subcadenas de cada cadena de S y siguiendo el mismo orden. Por ejemplo, si tenemos $S = \{abaabb, aaba, bbaaa, aababba, ab\}$ y efectuamos la división para $D = 2$, tendríamos que $S^1 = \{aba, aa, bba, aaba, a\}$ y $S^2 = \{abb, ba, aa, bba, b\}$.

Entrada: $S = \{s^1, s^2, \ldots, s^n\}$, $D \in \mathbb{N}$ número de divisiones
Salida: $m \in \Sigma^*$ cadena media aproximada
Inicio
Para $i = 1, \ldots, D$ $S^i = \emptyset$ **FPara** /* Inicialización a vacío */
Para $i = 1, \ldots, n$
  Para $j = 1, \ldots, D$
    $s^{ij} = \text{subcadena}(s^i, j, D)$ /* Hallamos la subcadena */
    $S^j = S^j \cup \{s^{ij}\}$ /* La añadimos al conjunto */
  FPara
FPara
Para $i = 1, \ldots, D$ $m^i = \text{cadena\_media\_aproximada}(S^i)$ **FPara**
Devolver $m = m^1 \cdot m^2 \cdots m^D$

Figura 5.1: Algoritmo del método de la división

Una vez obtenidos los S^i para $i = 1, \ldots, D$, se calcula la cadena media aproximada de cada uno de ellos aplicando alguna de las aproximaciones vistas en la Sección 3.1 (también podría obtenerse la cadena mediana), obteniendo D cadenas $m^1, m^2, \ldots, m^D$, cada una de ellas extraída de $S^1, S^2, \ldots, S^D$. La cadena media aproximada final usando este método sería $m = m^1 \cdot m^2 \cdots m^D$. La Figura 5.1 muestra el algoritmo a seguir [50]. En dicho algoritmo, la llamada a la función "subcadena(s, i, j)" indica que se debe obtener la i-ésima división de un total de j divisiones que queremos efectuar de la cadena s. Asimismo, la llamada a "cadena_media_aproximada(S)" obtiene la cadena media aproximada del conjunto de cadenas S por el método que se decida.

Respecto al coste computacional de dicho algoritmo, la partición en los conjuntos de subcadenas $S^1, \ldots, S^D$ es de orden $O(n \cdot D \cdot l)$, al ser el coste de obtener la subcadena de orden lineal con su longitud. Y la parte más costosa es el bucle de cálculo de las cadenas medias parciales aproximadas, que será $O(l'^3 \cdot |\Sigma| \cdot n \cdot D)$, donde l' sería la nueva longitud máxima entre las cadenas de S^i. Claramente, $l' = \frac{l}{D}$, con lo cual la complejidad asintótica es $O((\frac{l}{D})^3 \cdot |\Sigma| \cdot n \cdot D)$, que simplificando queda finalmente en $O(\frac{l^3}{D^2} \cdot |\Sigma| \cdot n)$. Por tanto, esta optimización no produce una mejora del coste temporal asintótico, pero sí una mejora del coste real, al quedar dividido por el número de divisiones al cuadrado. Nótese que el coste total no es realmente ese, pues dentro del cálculo de la cadena media parcial el número de iteraciones de perturbación también influye, y no está claro cómo puede afectar a dicho número de iteraciones el trabajar con cadenas más cortas. Lo más probable es que al ser cadenas más cortas, ese número de iteraciones disminuya (pues el espacio de búsqueda que se explore será más reducido).

Este método es aplicable a cualquier aproximación a la cadena media, no sólo la dada por los algoritmos iterativos (es decir, se puede aplicar a la cadena mediana y al método constructivo voraz). Por otra parte, hay que señalar que, aunque en la invocación no se incluya, es necesario una cadena inicial para cada conjunto de subcadenas de manera que se puedan empezar a hacer las

perturbaciones sobre ella. Dicha cadena inicial puede ser la cadena mediana o la obtenida por el proceso voraz.

5.3. Método de la optimización local

Dentro de los algoritmos iterativos de obtención de aproximaciones a la cadena media, hemos visto que la parte más costosa es la dedicada a las operaciones de sustitución e inserción, pues es necesario repetir la perturbación en una posición para cada símbolo de Σ. Esto hace que el coste temporal total tenga ese factor $|\Sigma|$, que generalmente es menor que l y, por tanto, tiene una influencia menor. Sin embargo, conseguir eliminar del coste asintótico dicho factor sería una ganancia importante, sobre todo cuando el alfabeto sea grande.

Esta consideración nos lleva a hacer una reflexión sobre el sentido de usar todos los símbolos en la perturbación. Parece razonable que cuando se prueba a sustituir un símbolo no todos los símbolos disponibles del alfabeto tengan sentido en la sustitución; por ejemplo, si la probabilidad de que un símbolo a se confunda por un símbolo b o c es diez veces mayor que la probabilidad de que se confunda con d o e, el cambiar a por b o c dará, con toda probabilidad, una mejora mucho mayor que el cambiar por d o e. Igualmente puede pasar en las inserciones: ciertos símbolos del alfabeto tienen más probabilidades de ser insertados en la posición actual que otros. No sólo se puede tener en cuenta la probabilidad *a priori* de que un símbolo sea sustituído por otro, o sea insertado, sino que también puede depender del contexto: los símbolos adyacentes a la posición actual pueden determinar si es más probable insertar o sustituir determinado símbolo.

Basándose en dicha idea, una modificación de los algoritmos es evidente: limitar las pruebas en sustitución e inserción a un cierto número de símbolos seleccionados. El problema es, por tanto, qué símbolos seleccionar. Evidentemente, esta selección de símbolos puede tener en cuenta factores muy complejos (estudio estadístico de las diferencias y operaciones de edición entre cadenas del conjunto, influencia de la posición de la cadena, etc.), pero lo deseable para conseguir una complejidad temporal adecuada es un proceso rápido y basado en datos poco costosos de obtener.

Siguiendo esta filosofía, una forma evidente es proceder basándose en la matriz de pesos. Se puede asumir, siempre que la matriz de pesos esté bien diseñada, que la probabilidad de que dos símbolos se confundan es inversamente proporcional al peso que tiene asignada dicha sustitución. Respecto a las inserciones no existe un criterio tan claro. Una idea es que, de nuevo, los símbolos con menor peso de inserción tienen mayor probabilidad de insertarse, pero esto nos limitaría siempre a los mismos símbolos sin recurrir a ningún tipo de información contextual. Por tanto, otra idea es suponer que el símbolo previo (también podría considerarse el posterior) a la posición de inserción tiene influencia en qué símbolos tienen mayor probabilidad de insertarse, siguiendo un criterio semejante al de la sustitución (es decir, basándose en la matriz de pesos).

Basándose en esto, se define un conjunto de símbolos candidatos a sustitu-

ción y otro de candidatos a inserción para cada símbolo del alfabeto basándose en un análisis simple de la matriz de pesos. Este análisis consistiría en encontrar los k símbolos con menor peso de sustitución para cada símbolo, lo cual puede realizarse en $O(|\Sigma|^2)$ y es perfectamente aceptable. Los candidatos para sustitución serían esos k símbolos y los de inserción serían el propio símbolo precedente y los k símbolos usados en sustitución para dicho símbolo. A este conjunto de k símbolos lo denominaremos vecindad o localidad, y de ahí viene el nombre de esta técnica (optimización local). En la Figura 5.2 vemos la modificación del algoritmo de perturbación iterativo conjunto para hacer uso de esta técnica. Dicha modificación consiste en aplicar sustituciones e inserciones sólo para los símbolos pertenecientes a la vecindad computada (es decir, en vez de hacerlo $\forall a \in \Sigma$, se hace $\forall a \in v_b$, siendo v_b los símbolos asociados a $b \in \Sigma$). La modificación de la versión separada sería muy semejante.

Respecto a la complejidad computacional, resulta claro que el factor $|\Sigma|$ queda ahora eliminado del coste temporal de dichos algoritmos y en su lugar entra la constante k. Como generalmente se buscan valores de k pequeños (sobre todo con respecto al resto de factores que afectan el coste temporal), también se puede obviar, con lo cual queda un coste temporal final $O(l^3 \cdot n)$. Por tanto, la complejidad asintótica queda reducida en el factor $|\Sigma|$ que buscábamos.

5.4. Experimentos comparativos

En esta sección vamos a comparar los resultados que se obtienen usando los prototipos extraídos aplicando las técnicas de reducción de coste expuestas (división y optimización local) con los de los prototipos de las técnicas sin reducción de coste. La comparación se hará usando de nuevo el corpus de cromosomas descrito en el apartado 2.2.1. Debido a los resultados mostrados en el Capítulo 3, reducimos la experimentación en este caso a los prototipos obtenidos a partir de la inicialización por cadena mediana y el método iterativo conjunto, ya que en el resto de casos el comportamiento es similar.

Las pruebas con el método de la división se hicieron para 2 y 3 divisiones. Con respecto a la optimización local, uno de los factores que hay que determinar es la vecindad de un símbolo, es decir, el conjunto de símbolos que se va a probar en sustituciones e inserciones. En esta experimentación, dicha vecindad se limita a los dos símbolos con menor peso de sustitución respecto al símbolo considerado. Por ejemplo, para este corpus, con la matriz de pesos usada y para el símbolo a, su vecindad vendría constituída por $\{=, b\}$.

Los resultados de clasificación, con sus respectivos intervalos de confianza, se presentan en las gráficas de las Figuras 5.3 y 5.4. La Figura 5.3 presenta resultados para clasificación por vecino más próximo entre 1 y 100 agrupamientos, mientras que las gráficas de la Figura 5.4 presentan el detalle de 1 a 10 agrupamientos y para distinto número de vecinos (1, 3 y 6).

Como era de prever, en líneas generales se verifica que las aproximaciones obtenidas mediante el método de la división resultan de menor calidad a la hora de clasificar en términos absolutos. Sin embargo, el solapamiento entre los

Entrada: $m \in \Sigma^*$ solución inicial, $S = \{s^1, s^2, \ldots, s^n\}$, W matriz de pesos de $\Sigma \times \Sigma$
Salida: $m \in \Sigma^*$ solución final
Inicio
ParaTodo $a \in \Sigma$ $v_a = \text{vecindad}(a, W)$ **FParaTodo**
Hacer
  $d_p = \sum_{i=1}^{n} d(m, s^i)$
  $d_a = d_p$
  Para $i = 1, \ldots, |m|$ /* Para todas las posiciones */
    $m^{sub} = m$
    $sb = \emptyset$
    $d_{sub} = d_a$
    ParaTodo $a \in v_{m_i}$ /* Aplica sust. para los símb. vecinos */
      $m_i^{sub} = a$
      Si $\sum_{j=1}^{n} d(m^{sub}, s^j) < d_{sub}$ **Entonces** /* Hay mejora */
        $sb = a$
        $d_{sub} = \sum_{j=1}^{n} d(m^{sub}, s^j)$
      FSi
    FParaTodo
    Si $sb \neq \emptyset$ **Entonces** $m_i^{sub} = sb$ **Sino** $m^{sub} = m$ **FSi**
    $m^{bor} = m_1 \cdots m_{i-1} m_{i+1} \cdots m_{|m|}$ /* Aplica borrado */
    $m^{ins} = m$
    $sb = \emptyset$
    $d_{ins} = d_a$
    ParaTodo $a \in \{m_i\} \cup v_{m_i}$ /* Aplica ins. para los vecinos */
      $m^{ins} = m_1 \cdots m_{i-1} \cdot a \cdot m_i \cdots m_{|m|}$
      Si $\sum_{j=1}^{n} d(m^{ins}, s^j) < d_{ins}$ **Entonces** /* Hay mejora */
        $sb = a$
        $d_{ins} = \sum_{j=1}^{n} d(m^{ins}, s^j)$
      FSi
    FParaTodo
    Si $sb \neq \emptyset$ **Entonces** $m^{ins} = m_1 \cdots m_{i-1} \cdot sb \cdot m_i \cdots m_{|m|}$
      Sino $m^{ins} = m$
    FSi
    $m = \text{argmin}_{s \in \{m, m^{sub}, m^{bor}, m^{ins}\}} \sum_{j=1}^{n} d(s, s^j)$ /* La mejor */
    $d_a = \sum_{j=1}^{n} d(m, s^j)$
  FPara
Mientras $d_a < d_p$ /* Hasta que no hay mejora global */
Devolver m

Figura 5.2: Algoritmo del método iterativo conjunto aplicando la técnica de optimización local.

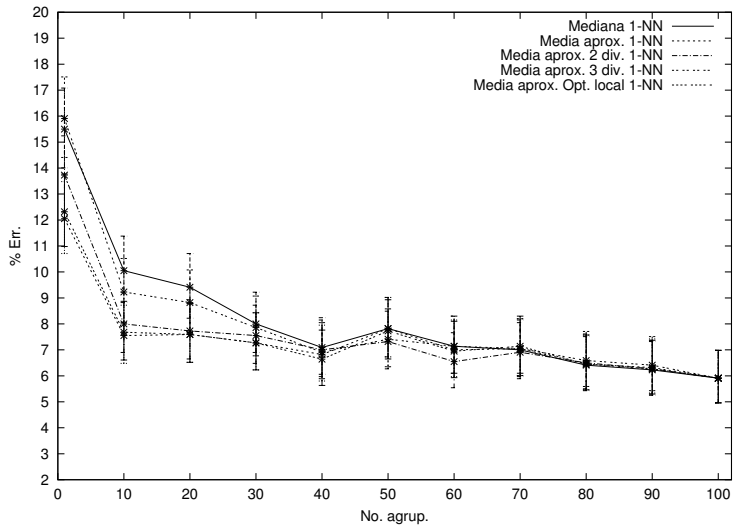


Figura 5.3: Resultados de clasificación usando la cadena mediana, la cadena media aproximada sin optimizar y las cadenas medias aproximadas obtenidas por optimización local y 2 y 3 divisiones, usando la cadena mediana como inicialización y el método conjunto, para 1-NN

intervalos de confianza entre la aproximación no optimizada y la que usa dos divisiones resulta lo bastante alto como para concluir que no hay diferencias significativas entre usar la aproximación sin optimizar y la optimizada mediante dos divisiones. Este grado de solapamiento se reduce ostensiblemente para tres divisiones (siempre para un número reducido de agrupamientos). Existen algunos casos concretos en los que, de manera absoluta, los resultados de la aproximación usando división superan incluso a la aproximación sin optimizar (como puede verse en la Figura 5.3 para 60 agrupamientos). Lo que sí parece evidente es que, salvo excepciones como la citada, aumentar el número de divisiones conduce a una peor aproximación.

En cambio, en estos resultados de clasificación se ve que la diferencia de calidad entre los prototipos obtenidos por optimización local y los obtenidos sin optimizar es prácticamente inapreciable. De hecho, los intervalos de confianza se solapan en su práctica totalidad. Por tanto, se deduce que la técnica de optimización local obtiene prototipos que apenas reducen la calidad de la tarea de clasificación. Así, se demuestra la gran ventaja a nivel de resultados de clasificación de usar optimización local con respecto a la división, pues mientras los intervalos de confianza de los resultados por el método de la división presentan un solapamiento moderado con los resultados no optimizados, en el caso del uso de la optimización local este solapamiento es casi completo. Hay más resultados disponibles en el Apéndice A, Sección A.6, en la Figura A.11.

En el Apéndice B se presentan resultados complementarios obtenidos sobre

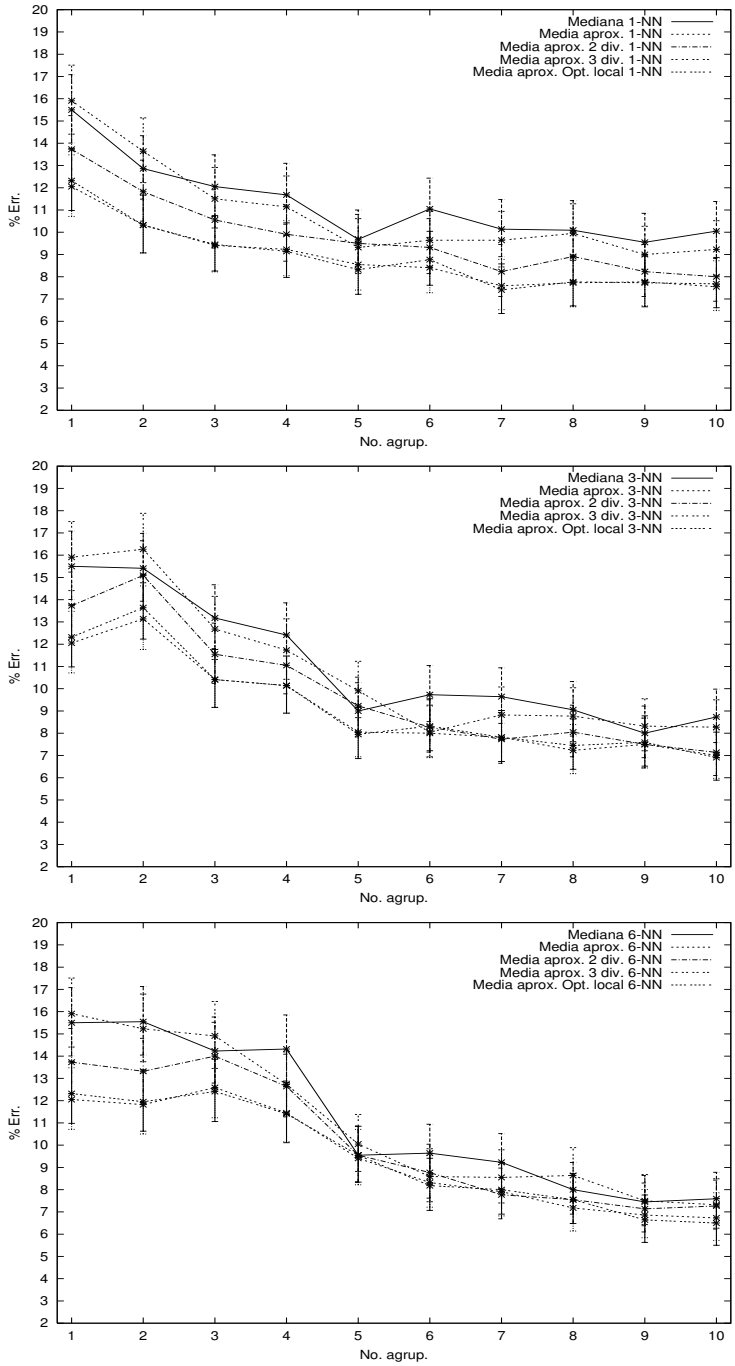


Figura 5.4: Resultados de clasificación usando la cadena mediana, la cadena media aproximada sin optimizar y las cadenas medias aproximadas obtenidas por optimización local y 2 y 3 divisiones, usando la cadena mediana como inicialización y el método conjunto, para 1, 3 y 6-NN

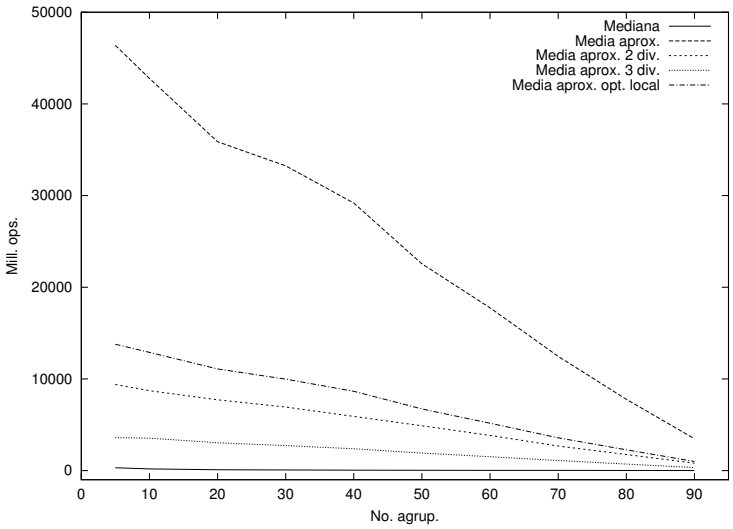


Figura 5.5: Coste temporal de obtención de la cadena mediana, la cadena media aproximada sin optimizar y las cadenas medias aproximadas obtenidas por optimización local y 2 y 3 divisiones, usando la cadena mediana como inicialización y el método conjunto

otro corpus de mayor tamaño usando la optimización local. Dichos resultados confirman las conclusiones obtenidas en este capítulo y muestran nuevas características de la optimización local.

Por último, queda comprobar la eficacia temporal de las optimizaciones. A la hora de obtener una medida del tiempo necesario para obtener cada aproximación, se ha optado por usar el producto de las longitudes de las cadenas comparadas (pues el coste temporal de obtener la distancia de edición normalizada es, en promedio, del orden del producto de las longitudes de las cadenas comparadas). Así, se utiliza la suma de los productos de las longitudes de las cadenas comparadas como medida de coste temporal.

Siguiendo dicho criterio respecto al coste temporal, se presentan los resultados en la gráfica de la Figura 5.5. Los costes temporales para un menor número de agrupamientos no se muestran debido a que problemas de desbordamiento restaban credibilidad a los datos de coste obtenidos.

Como se puede ver, para el método de la división la ganancia temporal es espectacular, bajando prácticamente un orden de magnitud con respecto a la versión no optimizada usando 3 divisiones. Igualmente, el uso de la optimización local reduce en un factor considerable el coste de obtención de la cadena media aproximada. Como el comportamiento en clasificación es casi idéntico al de los prototipos obtenidos usando técnicas no optimizadas, se concluye que esta técnica de optimización es realmente la más adecuada como método de obtención de la cadena media aproximada, aunque su ganancia temporal no sea tan alta

como la obtenida mediante el método de la división.

5.5. Resumen

Este capítulo se ha dedicado a abordar el problema de la reducción del coste temporal de la obtención de las aproximaciones a la cadena media. Para ello, inicialmente se ha hecho un somero repaso de posibles optimizaciones temporales generales implementables para conseguir las aproximaciones con menor coste. Tras ello se describen dos técnicas específicas, llamadas técnica de la división y técnica de la optimización local, que permiten una reducción notable del coste temporal de la obtención de las aproximaciones. Para demostrar la idoneidad de dichas técnicas, se han realizado experimentos comparativos usando el corpus *Copenhagen* que han demostrado la alta ganancia en el coste temporal de las técnicas (especialmente la técnica de la división) y la escasa degradación de los prototipos obtenidos con respecto a tareas de clasificación (especialmente la técnica de optimización local). La técnica de la optimización local se ha revelado así como la mejor alternativa para un cálculo menos costoso de prototipos de calidad.

Capítulo 6

La cadena media exacta

En este capítulo se aborda la posibilidad de obtener la cadena media exacta, hasta ahora descartada por su elevado coste computacional. Para ello, se propone un algoritmo basado en Ramificación y Poda y se demuestran las cotas necesarias para su implementación. Posteriormente, se usa un corpus artificial, con un número reducido de símbolos y longitud limitada a fin de acotar razonablemente el tiempo de obtención, para comparar como prototipos a nivel de clasificación y de distancia acumulada la cadena media exacta con sus diversas aproximaciones.

6.1. Obtención de la cadena media exacta

En esta sección vamos a proponer un método para hallar la cadena media exacta. Este método sigue el esquema general de Ramificación y Poda [27], donde se realiza una exploración dirigida de todo el espacio de búsqueda (en este caso, Σ^*). Este espacio de búsqueda puede representarse mediante un árbol; un árbol es un conjunto de elementos, llamados nodos, relacionados entre sí mediante una relación padre-hijo. En un árbol existe un único nodo sin padre, llamado raíz. A los nodos que no tienen hijos se les denomina hojas, que representan las soluciones dentro del espacio de búsqueda. A los nodos que no son hojas se les denomina nodos internos y representan soluciones parciales. Así pues, la exploración del espacio de búsqueda se haría mediante un recorrido de este árbol.

En la práctica, el recorrido del árbol se va haciendo a medida que se genera. Esta generación se hace a partir de una lista de nodos del árbol pendientes de evaluación, llamada lista de nodos vivos; inicialmente, esta lista contiene la raíz del árbol (lo que sería una solución parcial trivial inicial). A cada paso del algoritmo se va generando dinámicamente el árbol de exploración hasta llegar a una solución (hoja en el árbol), la cuál es evaluada según el valor de la función objetivo que se busca optimizar. Esta generación se hace basándose en el nodo vivo actual, del cuál se generan sus hijos (soluciones parciales derivadas del

Entrada: s solución parcial inicial
Salida: s_{opt} solución óptima
Inicio
$lnv = \{s\}$; $s_{opt} = s$; $v_{opt} = F(s_{opt})$ /* Inicialización */
Mientras $lnv \neq \emptyset$
  Escoger $s_{act} \in lnv$ /* Tomamos el siguiente nodo */
  Si hoja(s_{act}) **Entonces** /* Es solución */
    Si $F(s_{act}) < v_{opt}$ **Entonces** /* Y mejor que la actual */
      $s_{opt} = s_{act}$; $v_{opt} = F(s_{opt})$
      ParaCada $s_{aux} \in lnv$ /* Hacemos la poda */
        Si $g(s_{aux}) > v_{opt}$ **Entonces** $lnv = lnv - s_{aux}$ **FSi**
      FParaCada
    FSi
  Sino /* No es solución */
    hj = GenerarHijos(s_{act}) /* Genera hijos */
    ParaCada $h \in hj$
      /* Si su cota no es peor, lo inserta */
      Si $g(h) < v_{opt}$ **Entonces** Insertar(h,lnv) **FSi**
    FParaCada
    FSi
  $lnv = lnv - s_{act}$
FMientras
Devolver s_{opt}

Figura 6.1: Esquema algorítmico general de Ramificación y Poda para el caso de minimización, donde F es la función objetivo y g la cota

mismo). Sin embargo, esta generación está limitada, pues existe una función, llamada cota, que nos acota el valor de cualquier solución final obtenible a partir de la solución parcial. De esta manera, si la cota del nuevo nodo a generar es peor que la mejor solución obtenida hasta ahora, dicho nodo no se incluye en la lista de nodos vivos. De igual manera, al alcanzar una nueva solución mejor que la previa, se deben eliminar de la lista de nodos vivos todos aquellos cuya cota sea peor que la nueva solución.

El esquema algorítmico general de Ramificación y Poda se puede ver en la Figura 6.1. Como se puede ver, inicialmente la lista de nodos vivos la forma la solución inicial, y a cada iteración se extrae un nodo de dicha lista. Si ese nodo es una hoja (solución final), se comprueba si es mejor que la mejor solución final actual. Si se da ese caso, se toma como nueva mejor solución y se realiza la poda. Si el nodo no es una hoja, se generan sus hijos y se insertan en la lista de nodos vivos (siempre y cuando su cota sea mejor que el valor de la solución actual).

Particularizando para el caso que nos ocupa, sea S el conjunto de cadenas sobre el cual pretendemos hallar la cadena media m_S. La función objetivo a

Entrada: s solución parcial inicial
Salida: s_{opt} solución óptima
Inicio
$lnv = \{s\}$; $s_{opt} = s$; $v_{opt} = F(s_{opt})$ /* Inicialización */
Mientras $lnv \neq \emptyset$
  Escoger $s_{act} \in lnv$ /* Tomamos siguiente nodo */
  hj = GenerarHijos(s_{act}) /* Generamos hijos */
  ParaCada $h \in hj$
    /* Si su cota no es peor, lo inserta */
    Si $g(h) < v_{opt}$ **Entonces** Insertar(h,lnv) **FSi**
  FParaCada
  /* Si es mejor que la solución actual, lo tomamos */
  Si $F(s_{act}) < v_{opt}$ **Entonces**
    $s_{opt} = s_{act}$; $v_{opt} = F(s_{opt})$
    ParaCada $s_{aux} \in lnv$ /* Hacemos la poda */
      Si $g(s_{aux}) > v_{opt}$ **Entonces** $lnv = lnv - s_{aux}$ **FSi**
    FParaCada
  FSi
  $lnv = lnv - s_{act}$
FMientras
Devolver s_{opt}

Figura 6.2: Esquema algorítmico de Ramificación y Poda particular para el caso en el que los nodos internos se consideran soluciones. Se muestra para el caso de minimización, siendo F la función objetivo y g la cota

minimizar es:

$$F(s) = \sum_{t \in S} d(s,t) \tag{6.1}$$

donde d es la medida de disimilitud usada (distancia de edición, distancia de edición normalizada,...) y $s = s_1 s_2 \dots s_i$ es la cadena considerada, perteneciente al espacio de búsqueda Σ^*. Las soluciones parciales, al igual que las finales, son cadenas sobre Σ (en nuestro caso una solución parcial sería un prefijo de la posible cadena media). Como las soluciones parciales también pertenecen al espacio de búsqueda, cualquier nodo que se genere se puede considerar solución final (es decir, cualquier cadena generada se puede tomar como solución actual del problema hasta encontrar una mejor). Esto nos hace particularizar el algoritmo de Ramificación y Poda para nuestro caso, adoptando el esquema que se presenta en la Figura 6.2. En este caso, cada nodo que se toma de la lista de nodos vivos siempre es solución final, y como tal hay que hacer las comprobaciones para quedarse con la nueva mejor solución y hacer la poda (si se diera el caso) en cada paso. Pero también dicho nodo es solución parcial y se deben generar e introducir adecuadamente sus hijos.

Respecto a la parte de Ramificación, la generación de los nodos hijos se realiza añadiendo a la solución parcial en exploración todos los símbolos de Σ (de manera que cada nodo genera $|\Sigma|$ nodos hijos). Respecto a la parte de Poda, para cada nodo se puede calcular el valor de una cota $g(s)$ [27], el cual determina si el nodo será podado (se excluirá de la lista de nodos vivos y no generará sus hijos), ya que indica el valor mínimo de cualquier solución que descienda del mismo.

La poda se realiza en dos momentos distintos dentro del proceso de generación de nodos, tal y como se ve en el esquema presentado en la Figura 6.2:

1. Cuando el nuevo nodo generado presenta una cota superior a la mejor solución actual, no se incluye en la lista de nodos vivos (se poda él mismo y todos sus posibles hijos quedan implícitamente podados).
2. Cuando el nuevo nodo generado presenta un valor de la función objetivo inferior al de la mejor solución encontrada hasta ahora, se eliminan de la lista de nodos vivos todos aquellos cuya cota asociada sea superior al valor de la función objetivo de la nueva solución.

El problema real de este algoritmo, como en la mayoría de las aplicaciones de Ramificación y Poda, es el hallar la función de cota g adecuada, es decir, una función que sea cota (es decir, que dé una estimación optimista de la mejor solución que puede hallarse a partir de una solución parcial) pero que además sea una "buena" cota (lo suficientemente ajustada a la función objetivo). En nuestro caso, la función de cota tiene además la particularidad de variar según cuál sea la medida de disimilitud que se use en la función objetivo (ya que a distintas funciones objetivo, distintas cotas). Por tanto, nuestro siguiente objetivo es determinar las funciones de cota para las posibles funciones objetivo a satisfacer.

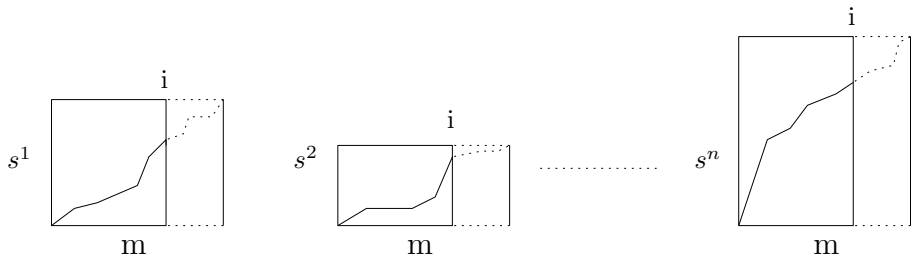


Figura 6.3: Secuencias de edición para la cadena media m y $s^1, s^2, \ldots, s^n$

6.2. Cota para la distancia de edición

Sea un alfabeto Σ; sea su monoide libre generado Σ^*. Sea $S = \{s^1, s^2, \ldots, s^n\} \subset \Sigma^*$ el conjunto de cadenas sobre el cual pretendemos encontrar la cadena media, siendo l^k la longitud de s^k y siendo $s^k = s_1^k s_2^k \ldots s_{l^k}^k$ para $k = 1, 2, \ldots, n$.

Sea $s = s_1 s_2 \ldots s_i$ la cadena candidata actual. Sea $D_s(k, j) = d(s_1^k s_2^k \ldots s_j^k, s)$ para $j = 0, \ldots, l^k$, siendo d la distancia de edición no normalizada (es decir, $D_s(k, j)$ computa la distancia entre s y el prefijo de longitud j de s^k).

Proposición 3 *$g_1(s) = \sum_{k=1}^{n} \text{mín}_{0 \le j \le l^k} D_s(k, j)$ es una cota inferior del valor de la función objetivo (6.1) cuando d es la distancia de edición no normalizada.*

Demostración: Supongamos que $m = m_1 m_2 \ldots m_l$, de longitud l, es la cadena media real para S, siendo d la distancia de edición no normalizada. Consideramos el prefijo de longitud $i \le l$ de m, $m_{pref_i} = m_1 m_2 \ldots m_i$, y el sufijo restante $m_{suf_i} = m_{i+1} \ldots m_l$. Para cada $s^k \in S$, está definida la secuencia de edición $p(k) = p_1, p_2, \ldots, p_{km}$ que indica las operaciones de edición entre m y s^k. Esta secuencia puede representarse como la concatenación de dos secuencias de edición $p_i^{pref}(k) = p_1, p_2, \ldots, p_j$ y $p_i^{suf}(k) = p_{j+1}, \ldots, p_{km}$, de manera que $p_i^{pref}(k)$ representa la secuencia de edición correspondiente a m_{pref_i} y $p_i^{suf}(k)$ la correspondiente a m_{suf_i}.

La distancia final de s^k a m viene dada por la suma de los pesos de $p_i^{pref}(k)$ y de $p_i^{suf}(k)$, tal y como se puede ver en la Figura 6.3. Por tanto,

$$F(m) = \sum_{k=1}^{n} d(m, s^k) = \sum_{k=1}^{n} W(p_i^{pref}(k)) + W(p_i^{suf}(k)) \tag{6.2}$$

para $i = 1, \ldots, l$, donde $W(p)$ representa el peso de una secuencia de edición p.

Claramente, la secuencia de edición $p_i^{suf}(k)$ tiene un peso mayor o igual a 0, por lo cual $\sum_{k=1}^{n} W(p_i^{pref}(k)) \le F(m)$, y es una función que sería cota. Sin embargo, al desconocer la cadena media desconocemos el valor de $p_i^{pref}(k)$ para cualquier i, y, por tanto, desconocemos el valor de su peso.

Sin embargo, sí que es claro que $W(p_i^{pref}(k))$ es igual a $D_{m_{pref_i}}(k,j)$ para algún j entre 1 y l^k. Por tanto, $\sum_{k=1}^{n} D_{m_{pref_i}}(k,j_k) \leq F(m)$ para el valor adecuado de j_k. Por tanto, al ser $\min_{j=1,\ldots,l^k} D_{m_{pref_i}}(k,j) \leq D_{m_{pref_i}}(k,j_k)$, tendremos que:

$$g_1(m_{pref_i}) = \sum_{k=1}^{n} \min_{j=1,\ldots,l^k} D_{m_{pref_i}}(k,j) \leq \sum_{k=1}^{n} D_{m_{pref_i}}(k,j_k) \leq F(m) \qquad (6.3)$$

donde recordemos que g_1 es la función de cota estudiada y m_{pref_i} es el prefijo hasta la posición i de la cadena media exacta m.

Por tanto, cualquier prefijo de m de longitud i tiene como cota inferior g_1, tal y como queríamos demostrar.
□

6.3. Cota alternativa para la distancia de edición

La cota definida en la Sección 6.2 no es la mejor cota posible si se conoce la longitud de la cadena media, ya que en ese caso se puede hallar una cota que, al menos, será tan optimista como la definida previamente.

Sean dos cadenas s y t sobre Σ, de manera que la longitud de s es l_s y la de t es l_t. Sea p la secuencia de edición óptima entre s y t, cuyo peso nos da la distancia de edición (ponderada o no) entre s y t. Dicha secuencia se puede representar como el camino en el grafo de edición de $(l_s+1)\cdot(l_t+1)$ nodos que indica cómo se van emparejando los símbolos de s y t, como se puede ver en la Figura 6.4. Recordemos que dicho grafo está basado en el algoritmo de Programación Dinámica que resuelve el problema del cálculo de la distancia de edición [80].

Proposición 4 *Para cualquier columna i del grafo de edición, $0 \leq i \leq l_s$, el prefijo de la secuencia de edición óptima p que llega hasta dicha columna i acaba en una fila j tal que* $\max\{0, l_t - (l_s - i)\} \leq j \leq l_t$.

Demostración: Por inducción: la posición (i,j) del grafo de edición sólo puede ser alcanzable, debido a que únicamente usamos las tres operaciones de edición básicas, desde las posiciones $(i-1,j)$ por borrado, $(i,j-1)$ por inserción y $(i-1,j-1)$ por sustitución.

Como base de inducción, demostramos que para el punto final de la secuencia de edición óptima entre s y t, es decir, (l_s,l_t), la subsecuencia óptima que acaba en l_s-1 está entre $l_t-(l_s-(l_s-1)) = l_t-1$ y l_t. Partiendo de (l_s,l_t), éste punto sólo es alcanzable desde (l_s-1,l_t), (l_s,l_t-1) y (l_s-1,l_t-1); como estamos interesados en la subsecuencia que acaba en l_s-1, basta con quedarnos con los puntos (l_s-1,l_t) y (l_s-1,l_t-1). Ninguna de las segundas coordenadas de estos

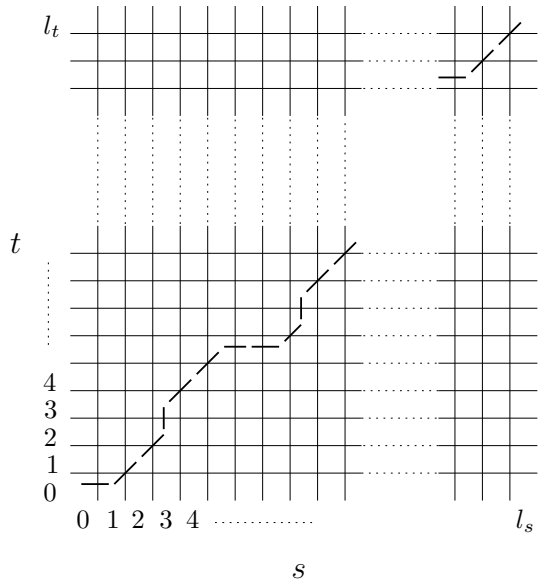


Figura 6.4: Grafo de Programación Dinámica para calcular $d(s,t)$.

dos puntos supera a l_t y tampoco ninguna de esas dos segundas coordenadas es inferior a $l_t - 1$, por lo cual queda demostrado.

Supongamos como hipótesis de inducción que para una columna i cualquiera se cumple que el camino óptimo pasa por una fila j tal que $\text{máx}\{0, l_t-(l_s-i)\} \leq j \leq l_t$.

El paso de inducción debe desarrollarse para $i-1$. Hasta cualquier posición (i,j) por la que pueda pasar el camino óptimo (suponiendo $j \geq 1$), se puede llegar a partir de las posiciones $(i-1,j)$, $(i,j-1)$ o $(i-1,j-1)$. Como sólo buscamos para la columna $i-1$, nos quedamos con los puntos $(i-1,j)$ y $(i-1,j-1)$. Teníamos que $\text{máx}\{0, l_t-(l_s-i)\} \leq j \leq l_t$, y se debe cumplir para $i-1$ que para el nuevo punto $j' \in \{j, j-1\}$ se dé $\text{máx}\{0, l_t-(l_s-i+1)\} \leq j' \leq l_t$. Si $j' = j$, se cumple directamente por la hipótesis de inducción (pues en todo caso hemos decrementado el límite inferior o lo hemos dejado a 0); si $j' = j-1$, al haber decrementado el límite inferior en uno, también se cumple la desigualdad por la hipótesis de inducción. En el caso de $j = 0$, como sólo es alcanzable desde $(i-1,j)$, se cumple lo dicho trivialmente.
□

Este hecho es aplicable a la cota que obtuvimos en la Sección 6.2. En dicha cota, considerábamos el valor mínimo de $D_s(k,j)$ para cualquier valor de fila j posible. Sin embargo, si conocemos la longitud de la cadena media l (o una cota máxima de dicha longitud), se puede reducir el rango del valor de j a las filas que nos indican la Proposición 4. Por tanto:

Proposición 5 $g_2(s) = \sum_{k=1}^{n} \min_{\max\{0,l^k-(l-i)\}\leq j\leq l^k} D_s(k,j)$ *es una cota inferior del valor de la función objetivo (6.1) cuando d es la distancia de edición no normalizada y l es (una cota superior de) la longitud de m.*

Demostración: La demostración se reduce a la demostración de las Proposiciones 3 y 4. Con la Proposición 3 demostrábamos que la suma para los valores mínimos de los $D_s(k,j)$ era cota inferior de la función objetivo, pero gracias a la Proposición 4 sabemos que sólo ciertas secuencias de edición son susceptibles de ser la óptima, con lo cual podemos reducir la búsqueda de los mínimos de los $D_s(k,j)$ a los j por los que puede pasar esa secuencia de edición óptima. □

Claramente, se cumplirá que $g_2(s) \geq g_1(s)$, al ser el conjunto de valores para g_2 subconjunto del de g_1, con lo cual dicha cota es más satisfactoria. El problema es que para la utilización de esta cota g_2 es necesario conocer la longitud de la cadena media o un valor máximo de la misma, a fin de garantizar que no se ignoren posibles secuencias válidas en una cierta columna del grafo de edición.

6.4. Cota para la distancia de edición normalizada

Las cotas propuestas g_1 y g_2 no son utilizables para el caso de la distancia de edición normalizada. Esto se debe a la no monotonicidad del camino óptimo para la distancia de edición normalizada (ya comentada en la Sección 2.1), característica que sí presenta la distancia de edición clásica y en la que se basan g_1 y g_2. Así pues, la obtención de la cota para la distancia de edición normalizada debe basarse en la definición de la misma.

Sea un alfabeto Σ; sea su monoide libre generado Σ^*. Sea $S = \{s^1, s^2, \ldots, s^n\} \subset \Sigma^*$ el conjunto de cadenas sobre el cual pretendemos encontrar la cadena media para la distancia de edición normalizada, siendo l^k la longitud de s^k y siendo $s^k = s_1^k s_2^k \ldots s_{l^k}^k$ para $k = 1, 2, \ldots, n$.

Sea $s = s_1 s_2 \ldots s_i$ la cadena candidata actual. Sea $D_s(k,j) = d(s_1^k s_2^k \ldots s_j^k, s)$ para $j = 0, \ldots, l^k$, siendo d la distancia de edición no normalizada. Sea l (una cota de) la longitud de la cadena media.

Proposición 6 $g_3(s) = \sum_{k=1}^{n} \frac{1}{l+l^k} \min_{\max\{0,l^k-(l-i)\}\leq j\leq l^k} D_s(k,j)$ *es una cota inferior del valor de la función objetivo (6.1) cuando d es la distancia de edición normalizada.*

Demostración: Basada en la propia definición de la distancia de edición normalizada y en la demostración de la Proposición 5.

En primer lugar, la distancia de edición normalizada entre dos cadenas s y t se define por $d_N(s,t) = \frac{W(p_N)}{L(p_N)}$, donde p_N es la secuencia de edición óptima entre s y t para la distancia de edición normalizada, $W(p_N)$ indica el peso de dicha

secuencia de edición y $L(p_N)$ su longitud (número de operaciones de edición de la misma) [53].

Como hemos visto en el ejemplo previo, p_N (secuencia de edición para la distancia normalizada) y p (para la no normalizada) pueden ser distintas, pero lo que sí es evidente es que $d(s,t) = W(p)$ (por la propia definición de la distancia de edición no normalizada) y por tanto $W(p)$ es el valor mínimo para cualquier secuencia de edición posible entre s y t. Por tanto, al ser p_N otra secuencia de edición entre s y t, posiblemente distinta de p, tendremos que $d(s,t) = W(p) \leq W(p_N)$. Por tanto, se dará que $d_N(s,t) \geq \frac{d(s,t)}{L(p_N)}$.

Igualmente, está demostrado en [53] que $L(p_N)$ está acotado, de manera que si la longitud de s es l_s y la de t es l_t, $\max\{l_s, l_t\} \leq L(p_N) \leq l_s + l_t$. Por tanto, nos queda que $d_N(s,t) \geq \frac{d(s,t)}{L(p_N)} \geq \frac{d(s,t)}{l_s+l_t}$.

Si aplicamos este hecho a la cadena media, tendremos que para cualquier $s^k \in S$ se da que $d_N(s^k, m) \geq \frac{1}{l^k+l} \cdot d(s^k, m)$. Si aplicamos la cota g_2 para $d(s^k, m)$, tendremos que para cualquier cadena candidata s de longitud i se dará que:

$$d_N(s^k, s) \geq \frac{1}{l^k + l} g_2(s) = \frac{1}{l^k + l} \min_{\max\{0, l^k - (l-i)\} \leq j \leq l^k} D_s(k, j) \qquad (6.4)$$

y por tanto la suma para todas las cadenas también será cota inferior, tal y como queríamos demostrar.
□

El problema de la cota g_3, al igual que el de g_2, es la obtención de la longitud o una cota superior de la longitud de la cadena media de S. Sin este factor, las cotas carecen de aplicabilidad, y es por tanto el siguiente punto a resolver.

6.5. Cota de la longitud de la cadena media

La longitud de la cadena media de un conjunto de cadenas S dependerá de la estructura de las propias cadenas de S, por lo cuál tendrá que venir determinada por las características de S.

Sea Σ un alfabeto; sea $S = \{s^1, s^2, \ldots, s^n\}$ el conjunto de cadenas sobre el cual pretendemos obtener la cadena media, $S \subset \Sigma^*$. Definimos por s_a^k el número de símbolos a presentes en la cadena s^k para todo $a \in \Sigma$ y $k = 1, \ldots, n$. Definimos $M_a = \max_{k=1,\ldots,n} s_a^k$ para todo $a \in \Sigma$.

Conjetura 1 *La longitud l de la cadena media m para S cumple que $l \leq \sum_{a \in \Sigma} M_a$.*

Para la Conjetura 1 no se ha encontrado, por el momento, demostración. Existen sin embargo una serie de razones intuitivas que permiten confiar que, con un grado de confianza bastante alto, es cierto. Estas razones intuitivas se basan en que la cadena media m debe cumplir que sea la que mejor se ajuste,

en promedio, al conjunto de cadenas en los grafos de Programación Dinámica usados para el cálculo de la distancia de edición normalizada.

El mejor ajuste respecto a una cadena sería que se dieran siempre sustituciones correctas en el grafo de edición. Evidentemente, para eso hay que tener el mismo número de símbolos de la cadena y en el mismo orden. Si luego se buscan a partir de esa cadena ajustes a otra cadena, sin que se deshagan las sustituciones correctas, será necesario insertar una serie de símbolos en determinadas posiciones para conseguir sustituciones correctas; el resto de símbolos sobrantes son inserciones (o borrados, según se considere) respecto a la cadena a la que se intenta ajustar.

Por tanto, para que la cadena media pueda ajustarse a todas las cadenas del conjunto de esta manera debería tener tantos símbolos $a \in \Sigma$ como aquella cadena de S que presente el máximo número de símbolos a. Incluír más símbolos a sólo estaría contribuyendo a más inserciones (o borrados) dentro del grafo de edición (ese número de símbolos sobrantes no puede hacer sustitución correcta en ninguna cadena de S).

Este razonamiento es el que nos lleva a considerar que la cadena media tiene esa cota de longitud máxima. Sin embargo, la falta de una prueba formal nos obliga a buscar mecanismos que nos permitan comprobar si la solución obtenida usando la cota dada por g_3 con esta longitud máxima es aceptable. Estos mecanismos de comprobación van a consistir en hallar cotas inferiores de la distancia acumulada de la cadena media exacta para el conjunto de cadenas S y en hacer pruebas empíricas que nos determinen si, en la práctica, se cumple dicha conjetura.

6.6. Cotas de la distancia acumulada de la cadena media exacta

En esta sección buscamos encontrar cotas que nos digan el valor mínimo que puede tener la distancia de edición acumulada de una cadena de Σ^* a un conjunto de cadenas $S \subset \Sigma^*$. Al ser para cualquier cadena de Σ^*, esto hace que dicha cota sea a su vez cota de la distancia acumulada de la cadena media de S a las cadenas de S. El obtener dichas cotas nos permitirá verificar el grado de desviación respecto al óptimo de nuestras aproximaciones y de la solución exacta, con lo cual se puede estimar la calidad de la aproximaciones frente a la solución exacta (por la variación de dicha desviación).

La obtención de dicha cota se puede basar en propiedades de la métrica o de la propia cadena media. En los siguientes puntos, describiremos dos cotas basadas en la desigualdad triangular y en las propiedades que debe cumplir la cadena media.

6.6.1. Cota obtenida por desigualdad triangular directa

Recordando la definición de cadena media m_S de un conjunto de cadenas $S = \{s^1, s^2, \ldots, s^n\} \subset \Sigma^*$, teníamos que era:

$$m_S = \underset{s \in \Sigma^*}{\operatorname{argmin}} \sum_{i=1}^{n} d(s, s^i)$$

donde d es la medida de distancia escogida.

Siguiendo esta definición, está claro que la distancia acumulada que presenta m_S al conjunto de cadenas S es:

$$\sum_{i=1}^{n} d(m_S, s^i) = d(m_S, s^1) + d(m_S, s^2) + \ldots + d(m_S, s^n) \tag{6.5}$$

Si tenemos que d es una distancia, es decir, entre otras propiedades cumple la desigualdad triangular, es evidente que $d(m_S, s^i) + d(m_S, s^j) \geq d(s^i, s^j)$. Aplicando este hecho sobre la fórmula dada en la Ecuación (6.5), y suponiendo que el número de cadenas n es par, tendremos que:

$$\sum_{i=1}^{n} d(m_S, s^i) = d(m_S, s^1) + \ldots + d(m_S, s^n) \geq d(s^1, s^2) + d(s^3, s^4) + \ldots + d(s^{n-1}, s^n)$$

Si n fuera impar, no supone ningún problema, pues simplemente:

$$\sum_{i=1}^{n} d(m_S, s^i) = d(m_S, s^1) + \ldots + d(m_S, s^n) \geq d(s^1, s^2) + \ldots + d(s^{n-2}, s^{n-1}) + d(m_S, s^n)$$

$$\geq d(s^1, s^2) + \ldots + d(s^{n-2}, s^{n-1})$$

Realmente, esta cota que hemos escogido se basa en sumar las distancias entre las cadenas de S dos a dos, de manera que no se repita ninguna de las cadenas. A estas posibles combinaciones las llamaremos emparejamientos. Por tanto, a pesar de que hemos expresado la cota mediante un emparejamiento concreto, se puede expresar mediante un emparejamiento general de S, lo cual podemos expresar mediante:

$$\sum_{i=1}^{n} d(m_S, s^i) \geq \sum_{j=1}^{\lfloor n/2 \rfloor} d(s^{e1[j]}, s^{e2[j]}) \tag{6.6}$$

donde $e1$ y $e2$ son dos conjuntos de índices sobre el rango $[1 \ldots n]$, disjuntos entre sí y que no admiten valores repetidos, que indican los emparejamientos.

Para obtener la cota más ajustada, lo que nos interesa es obtener el emparejamiento tal que nos dé la suma de distancias más elevada. La obtención de dicho emparejamiento es un problema combinatorio de magnitud bastante elevada (para n cadenas, del orden $n!$). Por tanto, para simplificar la elección del emparejamiento se ha optado por implementar una estrategia voraz la cual

	s^1	s^2	s^3	s^4
s^1	0	1	2	3
s^2	1	0	1	3
s^3	2	1	0	2
s^4	3	3	2	0

Figura 6.5: Distancias entre las cadenas del conjunto $S = \{s^1, s^2, s^3, s^4\}$

escoge el emparejamiento tomando las cadenas aún no emparejadas y emparejando aquellas dos que presentan entre sí una mayor distancia.

Evidentemente, esta estrategia es subóptima con respecto a la obtención de la cota más ajustada, ya que la forma de escoger las cadenas que forman el emparejamiento no permite corregir decisiones tomadas previamente que nos llevan a valores de distancia acumulada inferiores. Como contraejemplo, supongamos un conjunto formado por cuatro cadenas, $S = \{s^1, s^2, s^3, s^4\}$, que presentan entre ellas las distancias dadas en la Figura 6.5; se puede comprobar que, efectivamente, en S se cumple la desigualdad triangular. Sin embargo, al escoger las dos primeras cadenas a emparejar en S, podemos tomar el emparejamiento (s^1, s^4) inicialmente (aunque (s^2, s^4) también es posible), y eso nos lleva a obtener el emparejamiento total $\{(s^1, s^4), (s^2, s^3)\}$ con una distancia acumulada de 4. Sin embargo, si hubiéramos escogido (s^2, s^4), hubiéramos llegado al emparejamiento $\{(s^2, s^4), (s^1, s^3)\}$, que presenta una distancia acumulada de 5. Por tanto, queda claro que el orden en que se escogen los emparejamientos (aunque presenten la misma distancia uno a uno) resulta determinante para obtener la cota más ajustada, con lo cual la solución propuesta es subóptima.

La implementación realizada para encontrar la cota propuesta crea la matriz de distancias cruzadas y la utiliza, en primer lugar, para verificar que se cumple la desigualdad triangular, y después para crear los emparejamientos. Para emparejar, se escogen las dos cadenas con mayor distancia entre sí (en caso de empate, es la primera que se encuentra siguiendo el orden de enumeración habitual [1,2],[1,3],. . . ,[1,n],[2,3],. . .), y se eliminan para las posteriores selecciones, añadiendo su distancia al acumulado que da el valor final de la cota. El proceso acaba cuando quedan una o ninguna cadena por emparejar.

Evidentemente, esta cota es poco restrictiva, pero nos puede ayudar a hacernos una idea de lo alejadas que están del óptimo teórico las soluciones obtenidas por nuestro proceso de Ramificación y Poda. Sin embargo, una cota más ajustada, obtenida a partir de tener en cuenta otras posibles restricciones, es deseable, ya que nos daría una idea más realista de lo que se aleja la solución obtenida del posible valor óptimo de la distancia acumulada.

6.6.2. Cota obtenida por programación lineal

El problema de obtener la distancia acumulada de la cadena media m_S al conjunto de cadenas $S = \{s^1, s^2, \ldots, s^n\}$ podría expresarse como el problema

de minimizar la suma de distancias de m_S a las cadenas de S, es decir:

$$\min_{m_S \in \Sigma^*} \sum_{i=1}^{n} d(m_S, s^i) = \min_{m_S \in \Sigma^*} d(m_S, s^1) + d(m_S, s^2) + \ldots + d(m_S, s^n) \quad (6.7)$$

Naturalmente, debido a las características de la medida de distancia d, y más en concreto debido a la desigualdad triangular, se dan las siguientes restricciones para minimizar la distancia acumulada:

$$\left.\begin{array}{l} d(m_S, s^i) + d(m_S, s^j) \geq d(s^i, s^j) \\ d(m_S, s^i) + d(s^i, s^j) \geq d(m_S, s^j) \\ d(m_S, s^j) + d(s^i, s^j) \geq d(m_S, s^i) \end{array}\right\} \forall i, j \in \{1, 2, \ldots, n\}, i \neq j \quad (6.8)$$

Además, otra restricción viene dada por la no negatividad de la medida de distancia, es decir:

$$d(m_S, s^i) \geq 0 \quad \forall i \in \{1, 2, \ldots, n\}$$

Con estas restricciones, y teniendo como función objetivo la Ecuación (6.7), el resultado es que nuestro problema es lo que se conoce como un programa lineal [61]. Como tal, este programa lineal (y por tanto la cota) es resoluble por las técnicas clásicas de optimización usadas en Investigación Operativa, como por ejemplo el popular algoritmo *simplex* [61]. Además, está demostrado que la cota que se obtiene resolviendo este problema lineal es la óptima [31]. Es decir, partiendo del conjunto de las distancias entre las cadenas de S no es posible obtener una cota más ajustada a la distancia acumulada de la cadena media de S.

En el programa lineal a resolver, las variables son las distancias $d(m_S, s^i)$, y las restricciones son las indicadas en (6.8). Para resolverlo se hizo una implementación sencilla del algoritmo simplex, usando el método de la M [61] para obtener la solución admisible inicial.

Esta cota se puede comparar con la cota obtenida usando la Ecuación (6.6), que aunque sea menos ajustada puede resultar lo suficientemente buena para comprobar que nuestras soluciones al problema de la cadena media exacta se acercan en un cierto intervalo de confianza al posible óptimo teórico. Evidentemente, usar esta última cota obtenida por programación lineal resultará más discriminante, pero es interesante comprobar en qué grado son capaces de discriminar una y otra.

6.7. Experimentos comparativos

En esta sección vamos a describir los experimentos y resultados obtenidos usando como prototipo de clasificación la cadena media exacta frente a sus diversas aproximaciones. Debido a lo costoso del proceso de obtención de la cadena media exacta, se ha determinado usar un corpus de cadenas artificial, con un número reducido de símbolos y clases y una longitud moderada de las cadenas, que pasamos a describir en el apartado 6.7.1.

Cuadro 6.1: Corpus *abecede*: características que definen cada clase (longitud de la cadena y frecuencia de aparición de símbolos)

Clase	Longitud cadena		Frecs. de los símbolos			
	Media	Desv. típica	a	b	c	d
C1	5	1	0'5	0'2	0'2	0'1
C2	7	1	0'1	0'3	0'5	0'1
C3	4	2	0'4	0'4	0'1	0'1
C4	8	2	0'2	0'2	0'1	0'5

Cuadro 6.2: Características del corpus *abecede*

Número de clases	4
Número de objetos	400
Tamaño del alfabeto	4
Longitud de las cadenas (mínima-máxima)	1-13

6.7.1. Corpus *abecede*

El corpus *abecede* es un corpus artificial creado con el fin de probar la obtención de la cadena media exacta. Este corpus se constituye en un total de 400 cadenas sobre el alfabeto $\Sigma = \{a, b, c, d\}$; estas 400 cadenas se subdividen en cuatro clases distintas, caracterizadas por la longitud de las cadenas y la frecuencia de aparición de los símbolos. En el Cuadro 6.1 se indican las características de cada una de las clases del corpus.

Para la creación del corpus en realidad se fijaron previamente las características de cada una de las clases y posteriormente se generaron las cien cadenas correspondientes a cada una de ellas. El proceso consistió en elegir en primer lugar la longitud de la cadena usando una distribución normal sobre un generador de números aleatorios descrito en [43], con la media y desviación típica correspondiente a la clase y evitando longitudes negativas o nulas. Posteriormente, para cada una de las posiciones de la cadena se elegía el símbolo, usando la generación de números aleatorios estándar del lenguaje C para determinarlo y siempre teniendo en cuenta la probabilidad de aparición de los símbolos para dicha clase. El corpus completo obtenido se puede ver en el Apéndice C y un resumen de sus características en el Cuadro 6.2.

Respecto al desarrollo de los experimentos, quedan por determinar ciertos factores. El primero de ellos es la matriz de pesos con la que se va a calcular la distancia entre las cadenas; dicha matriz se presenta en la Figura 6.6. La medida de distancia a usar será, como en el resto de la experimentación realizada, la distancia de edición normalizada. Luego también se debe determinar el número de prototipos a extraer por clase, es decir, el número de agrupamientos en los

γ	a	b	c	d	λ
a	0	1	2	3	3
a	1	0	1	2	2
a	2	1	0	1	2
a	3	2	1	0	3
λ	3	2	2	3	-

Figura 6.6: Matriz de pesos usada con los experimentos del corpus *abecede*

cuales dividir el corpus. Se optó por una división en cinco agrupamientos distintos (que corresponden a cada una de las columnas presentadas en el Apéndice C) de manera arbitraria (es decir, no se aplicó ningún criterio específico para agrupar muestras con características comunes).

Los experimentos de clasificación se hicieron mediante validación cruzada aprovechando la división por agrupamientos realizada. Es decir, de los cinco agrupamientos, cuatro se usan para la extracción de los diversos prototipos y uno de ellos como conjunto a clasificar. Esto se hace para cada uno de los cinco agrupamientos, y el error de clasificación se promedia para estos cinco resultados de clasificación. En el apartado 6.7.2 se muestran los resultados de la experimentación llevada a cabo.

6.7.2. Comparación entre la cadena media exacta y aproximada

Para la experimentación partimos de un resultado básico de clasificación basado en usar todos los datos de cuatro agrupamientos como prototipos y los datos del agrupamiento restante como datos de prueba, para promediar el error de clasificación. Este experimento previo nos da el error mínimo al que se puede tender con nuestros experimentos de clasificación, y se mostrará como punto óptimo en las gráficas de resultados.

Para cada uno de los agrupamientos de cada una de las clases se obtuvieron diversos prototipos: cadena mediana, ocho variantes de cadena media aproximada (cambiando entre la definición clásica o la cuadrática, la inicialización por mediana o voraz y el proceso conjunto o separado) y la cadena media exacta. En los Cuadros 6.3 a 6.5 se presentan para cada agrupamiento y cada tipo de cadena los prototipos obtenidos (en el Cuadro 6.3 para inicialización por mediana, en el Cuadro 6.4 para inicialización voraz y en el Cuadro 6.5 para la mediana). Las cadenas que están encerradas entre paréntesis corresponden a soluciones parciales, obtenidas tras la generación de alrededor de cuatro millones de nodos (es decir, el proceso no acabó y en ese estado obtuvo esa mejor solución). La distancia acumulada de cada prototipo al conjunto de cadenas del cual se ha obtenido se presenta en los Cuadros 6.6 (inicialización mediana) y 6.7 (inicialización voraz), junto con el incremento relativo promedio de distancia acumulada (última fila). También se muestra la distancia de edición normaliza-

Cuadro 6.3: Cadenas obtenidas para la media aproximada con inicialización por mediana y proceso conjunto y separado, usando ambas definiciones de cadena media (clásica y cuadrática)

Clase-Agrup.	Mediana Conj.	Mediana Sep.	Mediana Conj. Cuadr.	Mediana Sep. Cuadr.	Media Exacta
C1-1	bacab	bacab	cacab	cacab	bacab
C1-2	caaca	caaca	baca	baca	caaca
C1-3	acbbb	acbbb	abbcb	abbcb	babcb
C1-4	cabac	caab	cabac	cabac	cabac
C1-5	acaac	acaac	acaac	acaac	acaac
C2-1	cccbccb	cccbccb	cccbccb	cccbccb	cccbcbc
C2-2	cbcccbc	cbcccbc	cbcbcbc	cbcbcbc	(cbcccbc)
C2-3	cbbccbc	cbbccbc	bcbccbc	bcbccbc	bcbccbc
C2-4	cbcbcc	cbcbcc	cbcbccc	cbcbccc	cbcbcc
C2-5	cbccccb	cbccccb	cbccccb	cbccccb	cbccccb
C3-1	babb	babb	babb	babb	babb
C3-2	bbbac	bbbac	babab	babab	bbbac
C3-3	abcbb	abcbb	abcbb	abcbb	bcabb
C3-4	bab	bab	bba	bba	bab
C3-5	bbcbb	bbcbb	bacbb	bacbb	bbcbb
C4-1	bdbdbdbd	bdbdbdbd	bdbdacdbc	bdbdbdbd	(dbdbdbdc)
C4-2	bddbdcbcdb	bddbdcbcdb	cdbddbcdb	bcdbdcbcdb	(bddbdcbcdb)
C4-3	dacbdbdcb	cbdcbbdcb	dbbbdddb	dbbbdddb	(dacbcdbdb)
C4-4	ddcdbdc	ddcdbdc	ddcbdbdc	ddccbdd	(ddcdbdc)
C4-5	bdbddacbd	bddbdacbd	bcdddacbd	bcdddacbd	(bdbddacbd)

da entre las distintas cadenas medias aproximadas y la media exacta obtenida en los Cuadros 6.8 y 6.9.

Como puede verse en dichas tablas, en la mayor parte de los casos la media exacta coincide con algunas de las aproximaciones realizadas. En otros casos no es así, pero la diferencia en la distancia acumulada es realmente despreciable respecto a la mejor aproximación. Por tanto, de esto parece concluirse que las diferencias a nivel de clasificación serán realmente mínimas o nulas.

Antes de pasar a los experimentos de clasificación, vamos a pasar a comparar la desviación de la distancia acumulada de la cadena media obtenida respecto a las cotas descritas en la Sección 6.6, a fin de observar la distancia que las separa del óptimo teórico. Para ello se obtuvieron las cotas por desigualdad triangular y por programación lineal para cada uno de los agrupamientos. Los resultados y las desviaciones promedias respecto a las cotas para la mediana y las medias exactas obtenidas se presentan en el Cuadro 6.10. Aquí se ve que la desviación de la mediana respecto a los límites teóricos propuestos es un 5 % superior a la desviación de la media, la cual está en torno al 15 % de desviación con respecto a la cota más ajustada (la obtenida por programación lineal), lo que

Cuadro 6.4: Cadenas obtenidas para la media aproximada con inicialización voraz y proceso conjunto y separado, usando ambas definiciones de cadena media (clásica y cuadrática)

Clase-Agrup.	Voraz Conj.	Voraz Sep.	Voraz Conj. Cuadr.	Voraz Sep. Cuadr.	Media Exacta
C1-1	baca	baca	baca	baca	bacab
C1-2	caaca	caaca	caaca	bcaca	caaca
C1-3	babcb	acbbb	babcb	babcb	babcb
C1-4	caab	caab	cbaac	bcaac	cabac
C1-5	acaa	acaa	acaa	acaa	acaac
C2-1	cccbccb	cccbccb	cccbccb	cccbccb	cccbcbc
C2-2	bcbccbc	bcbccbc	ccbccbc	ccbccbc	(cbcccbc)
C2-3	cbbccbc	cbbccbc	bbcccbc	bcbccbc	bcbccbc
C2-4	cbcbcc	cbcbcc	cbcbccc	cbcbccc	cbcbcc
C2-5	cbccccb	cbccccb	cbccccb	cbccccb	cbccccb
C3-1	bcab	bcab	bcab	bcab	babb
C3-2	bbba	bbba	babbc	babbc	bbbac
C3-3	bcabb	bcabb	bcabb	bbacb	bcabb
C3-4	bab	bab	bacb	bacb	bab
C3-5	bcbbb	bbbcb	cbbbb	bcbbb	bbcbb
C4-1	bdbdbdbd	bdbdbdbd	bdbdacdbc	bdbdbdbd	(dbdbdbdc)
C4-2	bdbddbdbc	bddbdcbcdb	bdbddbdbc	bdbddbdbc	(bddbdcbcdb)
C4-3	cbdbcdbdcb	cbdbcdbdcb	dacbdbdcb	cbdbcdbdcb	(dacbcdbdb)
C4-4	ddcdbdc	ddcdbdc	ddcbdbdc	ddcbdbdc	(ddcdbdc)
C4-5	bdbdbdcbc	bdbddabcd	bdbdacbdb	cbdbdacdac	(bdbddacbd)

Cuadro 6.5: Cadenas obtenidas para la mediana y la media exacta

Clase-Agrup.	Mediana	Media exacta
C1-1	aacab	bacab
C1-2	baca	caaca
C1-3	abbba	babcb
C1-4	caac	cabac
C1-5	aabab	acaac
C2-1	cbcccb	cccbcbc
C2-2	bbcccbc	(cbcccbc)
C2-3	cbbccbc	bcbccbc
C2-4	bbcccc	cbcbcc
C2-5	cbccbcb	cbccccb
C3-1	babb	babb
C3-2	babab	bbbac
C3-3	abcbb	bcabb
C3-4	bba	bab
C3-5	bacbb	bbcbb
C4-1	dbdddbd	(dbdbdbdc)
C4-2	bddddbddb	(bddbdcbcdb)
C4-3	dbbbdddb	(dacbcdbdb)
C4-4	ddddbdd	(ddcdbdc)
C4-5	bcdddacbd	(bdbddacbd)

Cuadro 6.6: Distancias acumuladas a cada agrupamiento de la cadena prototipo obtenida (mediana, medias aproximadas con inicialización mediana y media exacta) y desviación promedio de distancia acumulada

Clase-Agrup.	Mediana	Mediana Conjunto	Mediana Separado	Mediana Cj. Cuad.	Mediana Sep. Cuad.	Media Exacta
C1-1	17'73	16'91	16'91	17'14	17'14	16'91
C1-2	19'23	18'55	18'55	19'23	19'23	18'55
C1-3	19'00	18'63	18'63	18'50	18'50	18'10
C1-4	19'19	18'95	19'15	18'95	18'95	18'95
C1-5	17'50	16'16	16'16	16'16	16'16	16'16
C2-1	13'55	12'79	12'79	12'79	12'79	12'59
C2-2	13'56	13'35	13'35	13'48	13'48	13'35
C2-3	13'41	13'41	13'41	13'16	13'16	13'16
C2-4	12'69	12'20	12'20	12'36	12'36	12'20
C2-5	12'47	11'96	11'96	11'96	11'96	11'96
C3-1	20'56	20'56	20'56	20'56	20'56	20'56
C3-2	22'70	21'75	21'75	22'70	23'70	21'75
C3-3	23'40	23'40	23'40	23'40	23'40	22'83
C3-4	23'16	21'83	21'83	23'16	23'16	21'83
C3-5	23'46	23'00	23'00	23'46	23'46	23'00
C4-1	19'32	18'45	18'45	18'84	18'45	18'44
C4-2	21'04	19'74	19'74	19'80	19'97	19'74
C4-3	21'52	19'72	19'98	21'52	21'52	19'60
C4-4	17'77	17'20	17'20	17'45	17'68	17'20
C4-5	20'85	20'05	20'18	20'85	20'85	20'05
Prom.	4'3 %	0'5 %	0'6 %	2'2 %	2'4 %	-

Cuadro 6.7: Distancias acumuladas a cada agrupamiento de la cadena prototipo obtenida (mediana, medias aproximadas con inicialización voraz y media exacta) y desviación promedio de distancia acumulada

Clase-Agrup.	Mediana	Voraz Conjunto	Voraz Separado	Voraz Cj. Cuad.	Voraz Sep. Cuad.	Media Exacta
C1-1	17'73	17'12	17'12	17'12	17'12	16'91
C1-2	19'23	18'55	18'55	18'55	19'07	18'55
C1-3	19'00	18'10	18'63	18'10	18'10	18'10
C1-4	19'19	19'15	19'15	19'26	19'40	18'95
C1-5	17'50	16'37	16'37	16'37	16'37	16'16
C2-1	13'55	12'79	12'79	12'79	12'79	12'59
C2-2	13'56	13'48	13'48	13'71	13'71	13'35
C2-3	13'41	13'41	13'41	13'63	13'16	13'16
C2-4	12'69	12'20	12'20	12'36	12'36	12'20
C2-5	12'47	11'96	11'96	11'96	11'96	11'96
C3-1	20'56	21'03	21'03	21'03	21'03	20'56
C3-2	22'70	23'33	23'33	22'05	22'05	21'75
C3-3	23'40	22'83	22'83	22'83	23'25	22'83
C3-4	23'16	21'83	21'83	22'90	22'90	21'83
C3-5	23'46	23'60	23'03	23'77	23'60	23'00
C4-1	19'32	18'45	18'45	18'84	18'45	18'44
C4-2	21'04	20'04	19'74	20'04	20'04	19'74
C4-3	21'52	19'62	19'62	19'72	19'62	19'60
C4-4	17'77	17'20	17'20	17'45	17'45	17'20
C4-5	20'85	20'86	20'18	20'76	21'10	20'05
Prom.	4'3 %	1'3 %	1'1 %	1'7 %	1'7 %	-

Cuadro 6.8: Distancias de edición normalizada entre las diversas cadenas medias aproximadas obtenidas de cada agrupamiento (mediana y medias aproximadas con inicialización mediana) respecto a la media exacta obtenida y distancia promedio

Clase-Agrup.	Mediana	Mediana Conjunto	Mediana Separado	Mediana Cj. Cuad.	Mediana Sep. Cuad.
C1-1	0'20	0'00	0'40	0'20	0'20
C1-2	0'60	0'00	0'00	0'60	0'60
C1-3	0'80	0'80	0'80	0'40	0'40
C1-4	0'40	0'00	0'60	0'00	0'00
C1-5	0'80	0'00	0'00	0'00	0'00
C2-1	0'57	0'29	0'29	0'29	0'29
C2-2	0'14	0'00	0'00	0'14	0'14
C2-3	0'29	0'29	0'29	0'00	0'00
C2-4	0'33	0'00	0'00	0'29	0'29
C2-5	0'14	0'00	0'00	0'00	0'00
C3-1	0'00	0'00	0'00	0'00	0'00
C3-2	0'40	0'00	0'00	0'40	0'40
C3-3	0'80	0'80	0'80	0'80	0'80
C3-4	0'67	0'00	0'00	0'67	0'67
C3-5	0'20	0'00	0'00	0'20	0'20
C4-1	0'50	0'44	0'44	0'80	0'44
C4-2	0'40	0'00	0'00	0'40	0'10
C4-3	0'56	0'40	0'90	0'56	0'56
C4-4	0'29	0'00	0'00	0'25	0'29
C4-5	0'33	0'00	0'40	0'33	0'33
Prom.	0'42	0'15	0'25	0'32	0'29

Cuadro 6.9: Distancias de edición normalizada entre las diversas cadenas medias aproximadas obtenidas de cada agrupamiento (mediana y medias aproximadas con inicialización voraz) respecto a la media exacta obtenida y distancia promedio

Clase-Agrup.	Mediana	Voraz Conjunto	Voraz Separado	Voraz Cj. Cuad.	Voraz Sep. Cuad.
C1-1	0'20	0'40	0'40	0'40	0'40
C1-2	0'60	0'00	0'00	0'00	0'60
C1-3	0'80	0'00	0'80	0'00	0'00
C1-4	0'40	0'60	0'60	0'40	0'67
C1-5	0'80	0'40	0'40	0'40	0'40
C2-1	0'57	0'29	0'29	0'29	0'29
C2-2	0'14	0'43	0'43	0'29	0'29
C2-3	0'29	0'29	0'29	0'29	0'00
C2-4	0'33	0'00	0'00	0'29	0'29
C2-5	0'14	0'00	0'00	0'00	0'00
C3-1	0'00	0'75	0'75	0'75	0'75
C3-2	0'40	0'40	0'40	0'40	0'40
C3-3	0'80	0'00	0'00	0'00	0'40
C3-4	0'67	0'00	0'00	0'50	0'50
C3-5	0'20	0'40	0'40	0'40	0'40
C4-1	0'50	0'44	0'44	0'80	0'44
C4-2	0'40	0'73	0'00	0'73	0'73
C4-3	0'56	0'50	0'50	0'40	0'50
C4-4	0'29	0'00	0'00	0'25	0'25
C4-5	0'33	0'60	0'22	0'50	0'91
Prom.	0'42	0'31	0'30	0'35	0'41

Cuadro 6.10: Desviación promedio de las distancias acumuladas de la cadena mediana y la media exacta con respecto a las cotas por programación lineal y desigualdad triangular

	Desviación cota programación lineal	Desviación cota desigualdad triangular
Mediana	20'8 %	26'8 %
Media exacta	15'8 %	21'8 %

puede considerarse un incremento notable. Además, en dicha experimentación se constató que las diferencias entre la cota por programación lineal y la basada en desigualdad triangular no son muy altas (en promedio se diferencian apenas en un 6 %), lo que nos permite concluir que la cota basada en desigualdad triangular es una buena alternativa a la otra para hacer comprobaciones rápidas.

Como no parecen obtenerse confirmaciones concluyentes del uso de las cotas propuestas, se recurrió finalmente a hacer una comprobación empírica para ciertos conjuntos de datos reducidos, a fin de comprobar que las soluciones obtenidas fueran correctas y confirmar, de manera experimental, que las cotas usadas en el proceso (y en particular la cota de la longitud presentada en la Conjetura 1) funcionan en la práctica. En este caso, debido a sus características, se han escogido los conjuntos C1.3 y C3.1, cuyas cotas de longitud son respectivamente 9 y 10, y cuyas cadenas medias obtenidas son respectivamente *babcb* y *babb*. Para ello se realizó una generación de todas las cadenas existentes con el alfabeto $\Sigma = \{a, b, c, d\}$ hasta una longitud de nueve símbolos inclusive y se calculó la distancia acumulada media del conjunto total de cadenas de una cierta longitud a los conjuntos C1.3 y C3.1.

En la Figura 6.7 se ve la progresión de dicha distancia media al conjunto de datos correspondiente, así como la distancia acumulada de las cadenas de menor y mayor distancia acumulada. Como se puede ver, en ambos casos se alcanza el mínimo para las longitudes correspondientes a las medias exactas obtenidas (longitud 5 para C1.3 y longitud 4 para C3.1), y a partir de ahí presenta un crecimiento continuado (tanto en el caso medio como en el mínimo), no dando en ningún caso una cadena mejor para esas longitudes que presentan el mínimo. Además, esas longitudes resultan todas inferiores a la establecida como cota por la Conjetura 1, con lo cual parece demostrarse empíricamente que dicha conjetura es correcta.

En cuanto a los resultados de clasificación usando dichos prototipos, se presentan en las gráficas de la Figura 6.8. En ellas en principio no se observa una diferencia sustancial al variar el número de vecinos del clasificador. Sí que es destacable el hecho de que las medias que usan la inicialización voraz presenten un mejor comportamiento que las que usan la mediana. Estos hechos se deben con toda probabilidad a la naturaleza artificial del corpus, que hace que su distribución de probabilidad no sea precisamente natural. Así, el aumento

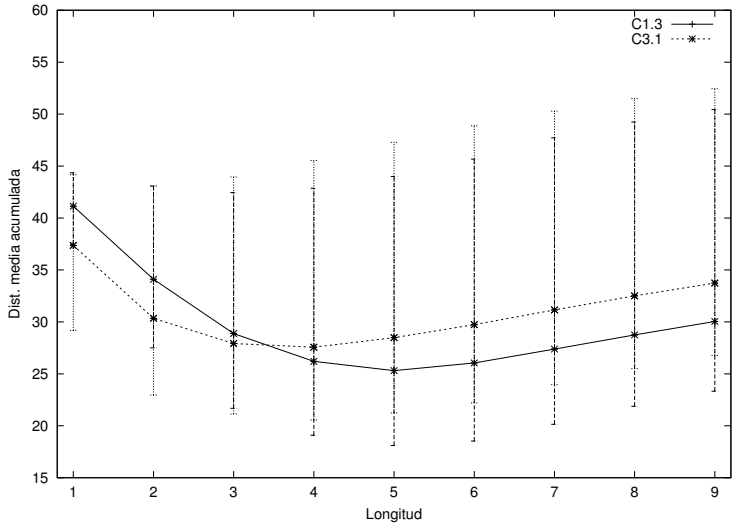


Figura 6.7: Distancia acumulada promedio de todas las cadenas de una cierta longitud sobre $\Sigma = \{a, b, c, d\}$ a los conjuntos de datos C1.3 y C3.1 y distancias de las cadenas de máxima y mínima distancia acumulada para dicha longitud

del número de vecinos no resulta tan beneficioso (pues no hay una distribución "natural" de los datos) y la mediana tampoco es, por tanto, un buen punto de partida para el proceso de optimización (pues tampoco sería un prototipo natural).

Respecto a la comparación con la media exacta, los resultados de clasificación se muestran en la Figura 6.9. De nuevo se observan comportamientos un tanto extraños, achacables seguramente a la naturaleza artificial del corpus y a la arbitrariedad de los agrupamientos, como es el caso de que usando vecino más próximo el peor resultado se dé usando la cadena media exacta. Sin embargo, sí que es apreciable que a medida que aumenta el número de vecinos la calidad de clasificación con la media exacta se va incrementando, aunque las diferencias con las cadenas medias aproximadas resultan poco significativas (inferiores a 2 puntos en cualquier caso con el mejor resultado de las aproximadas). Esto nos lleva a concluir que las aproximaciones a la cadena media propuestas son, en general, buenas aproximaciones para tareas de clasificación con respecto a la cadena media exacta.

6.7.3. Experimentos complementarios con un corpus no sintético

Debido a la naturaleza artificial del corpus *abecede*, se buscó confirmar las conclusiones obtenidas con un corpus no sintético. En este caso, el corpus escogido es el *Copenhagen* descrito en el apartado 2.2.1. Debido a que la longitud de

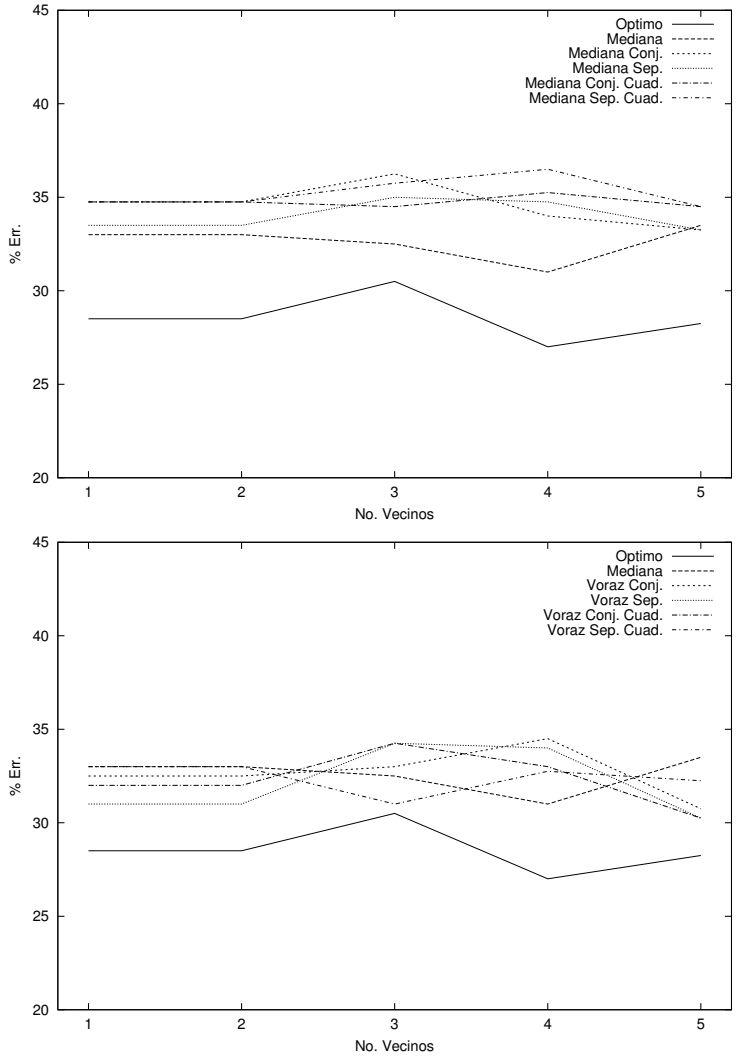


Figura 6.8: Resultados de clasificación para el corpus *abecede* con la cadena mediana y las cadenas medias aproximadas por inicialización por mediana (arriba) y por inicialización voraz (abajo). Se muestra también el óptimo (usando todas las cadenas restantes de la clase como prototipos).

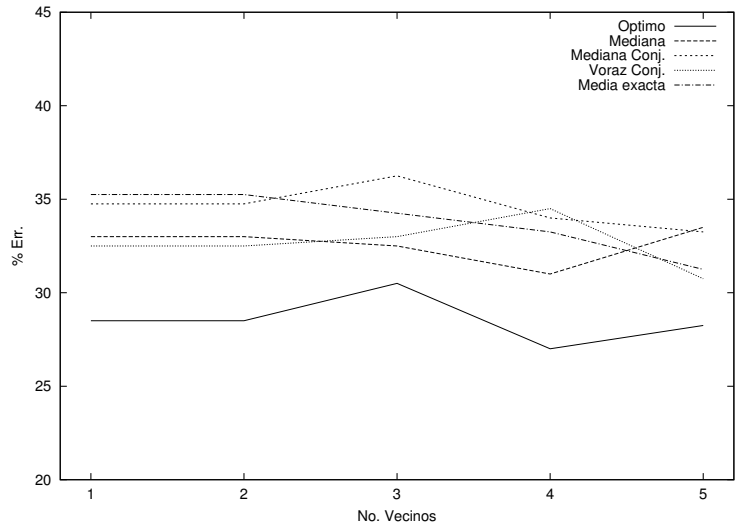


Figura 6.9: Resultados de clasificación para el corpus *abecede* con la cadena mediana, las cadenas medias aproximadas por inicialización mediana y voraz (usando el proceso conjunto y la definición clásica), y la cadena media exacta. Se muestra también el óptimo.

sus cadenas resulta excesiva para el tamaño del alfabeto, se realizó una recodificación de las cadenas para reducir la longitud de las mismas en un orden de magnitud. Esta recodificación se basó en sumar las diferencias representadas por segmentos de diez símbolos consecutivos, codificando el resultado con el símbolo que representa dicha diferencia final.

Con dicha recodificación se llevó a cabo el cómputo de todas las aproximaciones propuestas y de la cadena media exacta para tres clases (la 20, 21 y 22), que presentan las cadenas más cortas (longitudes de 3 y 4 símbolos). La distancia acumulada de cada uno de los prototipos computados con respecto al conjunto de entrenamiento se presenta en el Cuadro 6.11.

Como puede observarse, la desviación de distancia acumulada que presentan las diversas aproximaciones (incluyendo la mediana) son mínimas respecto a la media exacta calculada. Es destacable también que la media aproximada con inicialización por cadena mediana y proceso conjunto dé los mismos resultados que la media exacta, lo cual confirma que es la aproximación más adecuada.

6.8. Resumen

En este capítulo hemos examinado la forma de obtener la cadena media exacta mediante un proceso basado en Ramificación y Poda. Se han propuesto las diversas cotas aplicables, quedando por demostrar la conjetura sobre la longitud

Cuadro 6.11: Distancias acumuladas para las clases 20, 21 y 22 del corpus *Copenhagen* de cadenas reducidas para la cadena mediana, medias aproximadas con el método conjunto y media exacta

Prototipo Datos	Mediana	Media aproximada inic. mediana proc. conjunto	Media aproximada inic. voraz proc. conjunto	Media exacta
C20.A	105'0	104'0	107'3	104'0
C21.A	139'3	138'7	138'7	138'7
C22.A	81'4	81'4	81'4	81'4
C20.B	102'0	102'0	102'0	102'0
C21.B	151'3	151'3	151'3	151'3
C22.B	78'1	78'1	78'1	78'1

máxima de la cadena media para la distancia de edición normalizada. Para poder estudiar la fiabilidad de los resultados posteriores se han presentado las cotas de la distancia acumulada de la cadena media a fin de comprobar la desviación entre los resultados obtenidos y dichas cotas optimistas. También se ha realizado una exploración exhaustiva de un conjunto más reducido para verificar empíricamente la validez de las conjeturas. La experimentación sobre la obtención de la cadena media se ha efectuado sobre el corpus artificial *abecede*, a fin de obtener resultados en un tiempo razonable, y en una versión alterada del corpus *Copenhagen*. Los resultados tanto a nivel de distancia acumulada como a nivel de clasificación nos muestran que las diferencias entre la media exacta y las aproximaciones propuestas no son significativas.

Capítulo 7

Agrupamientos usando la cadena media

Este capítulo va a estudiar la aplicación de las aproximaciones a la cadena media en la construcción de agrupamientos de datos. Se efectuará una comparación entre agrupamientos obtenidos usando la cadena mediana y la media aproximada. Esta comparación se efectuará a nivel de resultados de clasificación, extrayendo diversos prototipos (cadena mediana y cadenas medias aproximadas) de los agrupamientos obtenidos usando cada una de las posibles combinaciones.

7.1. El método k-medias generalizado

En el apartado 1.4.3 hacíamos una descripción de los métodos basados en sumas de cuadrados más usuales para la obtención de agrupamientos: el k-medias y el k-medianas. En el caso de tratar con cadenas, el método k-medias resulta inviable debido a que el cálculo de la media implica, para cadenas, el cálculo de la cadena media (que es NP-Duro [11]). En cambio, la aplicación del método k-medianas únicamente implica el cálculo en cada iteración de la cadena mediana, lo cual es abordable en la práctica. En dicha línea, se han realizado trabajos previos [32].

Una alternativa clara al proceso de las k-medianas sería usar como nuevos representantes las cadenas medias aproximadas de cada agrupamiento, obteniendo dichas aproximaciones mediante los algoritmos presentados en el Capítulo 3. A este nuevo proceso se le denomina k-medias generalizado [49]. Básicamente, se trata de partir de nuevo de k muestras iniciales que generan los k agrupamientos correspondientes. Tras esta inicialización, el proceso calcula la cadena media aproximada de cada agrupamiento y obtiene así el nuevo conjunto de representantes. El proceso se repite sobre el nuevo conjunto de representantes hasta que los agrupamientos no cambian de una iteración a otra. El algoritmo del proceso k-medias generalizado se muestra en la Figura 7.1.

En este proceso hay que tener en cuenta un detalle. Como se vio en el

Entrada: $S = \{s^1, s^2, \ldots, s^n\}$ conjunto de cadenas, $R^0 = \{r^1, r^2, \ldots, r^k\}$ conjunto inicial de k representantes
Salida: $R = \{r^1, r^2, \ldots, r^k\}$ conjunto final de k representantes
Inicio
$R = R^0$
Para $i = 1$ **Hasta** k /* Hacemos agrupamientos iniciales */
$C_i = \{s \in S : i = \text{argmin}_{j=1,\ldots,k}\, d(s, r^j)\}$
Fpara
Hacer
$cambio = falso$
Para $i = 1$ **Hasta** k /* Calcular nuevos representantes */
$r^i = \text{cadena\_media\_aproximada}(C_i)$
FPara
Para $i = 1$ **Hasta** k /* Calcular nuevos agrupamientos */
$C'_i = \{s \in S : i = \text{argmin}_{j=1,\ldots,k}\, d(s, r^j)\}$
Si $C_i \neq C'_i$ **Entonces**
$cambio = cierto$ /* Cambian agrupamientos */
FSi
$C_i = C'_i$
Fpara
Mientras $cambio = cierto$ /* Hasta que los agrup. no cambian */
Devolver R

Figura 7.1: Algoritmo k-medias generalizado

Capítulo 3, los algoritmos de obtención de la cadena media aproximada necesitan una cadena inicial sobre la que realizar el proceso de perturbación. Además, dicho proceso no garantizaba que obtuviéramos un óptimo global, pero sí que la nueva cadena presenta una distancia acumulada con respecto al resto cadenas igual o menor que la cadena inicial.

Así, para calcular un nuevo representante necesitamos de una cierta cadena inicial. Si esta cadena inicial no es adecuada, podemos obtener un nuevo representante (cadena media aproximada) que presente mayor distancia acumulada a las cadenas del agrupamiento que el representante previo. Para evitar este riesgo, resulta necesario que la cadena inicial escogida para el proceso sea el representante actual. A esta opción la denominaremos k-medias generalizado guiado por representante.

Una alternativa plausible es usar la cadena mediana del agrupamiento como inicialización, pero esto sólo debe hacerse cuando dicha cadena mediana presenta menor distancia acumulada a las cadenas del agrupamiento que el representante actual. A esta opción la denominaremos k-medias generalizado guiado por mediana.

Al igual que k-medias y k-medianas, k-medias generalizado también necesita un conjunto de k muestras iniciales para que actúen como representantes. Los métodos de inicialización propuestos en el apartado 1.4.4 son también aplicables. En particular, en nuestro caso tomaremos la inicialización por maxmin con la que se ha trabajado en los capítulos previos. Además, en la Sección 7.2, proponemos una modificación del algoritmo maxmin para obtener nuevas inicializaciones.

7.2. Inicialización con maxmin modificado

El esquema básico del algoritmo maxmin para cadenas, presentado en la Figura 1.2 de Capítulo 1 puede adaptarse de manera sencilla para que el conjunto de representantes vaya variando de manera dinámica a cada iteración. En el esquema presentado en la Figura 1.2, este conjunto de representantes acaba siendo un subconjunto del conjunto de muestras, y además se va construyendo de manera monótona (es decir, $S^{t-1} \subset S^t$). Este hecho puede modificarse con cierta facilidad, haciendo que antes de añadir el nuevo representante se haga el recálculo de los representantes actuales.

Así, el esquema básico del algoritmo queda modificado en cada iteración, ya que previamente a elegir la muestra más alejada a los representantes, se hace un recálculo de este conjunto de representantes. La idea básica de este recálculo es usar algún método de reorganización de los agrupamientos (k-medianas o k-medias generalizado) para obtener el conjunto de representantes de la siguiente iteración. Con ese nuevo conjunto de representantes es con el que se determinará la muestra a añadir en el siguiente paso del maxmin. Esta modificación del algoritmo maxmin se presenta en la Figura 7.2. La llamada a "reagrupar" en dicho algoritmo representa el uso de k-medianas o de k-medias generalizado.

Si en esta reorganización utilizamos el método k-medianas, el conjunto de representantes en cada nuevo paso será el conjunto de cadenas medianas de los

Entrada: $T \subseteq \Sigma^*$ conjunto de cadenas, k número de agrupamientos a obtener
Salida: $S \subseteq \Sigma^*$ conjunto de k representantes de los agrupamientos
Inicio
$S = \emptyset$
$s =$ muestra_arbitr(T)
Para $t = 1, \ldots, k$ /* Hasta alcanzar el número de agrupamientos */
  $S = S \cup \{s\}$ /* Añadimos la seleccionada anteriormente */
  $S =$ reagrupar(T, S) /* Paso de reagrupamiento previo */
  $maxmin = 0$
  ParaTodo $u \in T - S$ /* Para todas las muestras no tomadas */
    $duv = \infty$
    ParaTodo $v \in S$
      /* Se toma la distancia al representante más cercano */
      Si $d(u, v) < duv$ **Entonces** $duv = d(u, v)$ **FSi**
    FParaTodo
    Si $duv > maxmin$ **Entonces** /* Si está más lejos, se toma */
      $s = u$
      $maxmin = duv$
    FSi
  FParaTodo
FPara
Devolver S

Figura 7.2: Esquema del algoritmo maxmin modificado con la inclusión de reagrupamientos en cada iteración

agrupamientos del paso previo. En este caso, tendremos que el conjunto final S^k de k muestras cumple $S^k \subseteq T$, pero su construcción no se habrá hecho de manera monótona (es decir, S^{t-1} no es necesariamente un subconjunto de S^t).

En el caso de que la reorganización se haga aplicando el k-medias generalizado, el conjunto final de representantes S obtenido no es necesariamente un subconjunto de T, al igual que la cadena media aproximada de un conjunto de cadenas no pertenecía necesariamente a dicho conjunto. Al igual que en la reorganización por k-medianas, la construcción tampoco es monótona en este caso. También hay que destacar que las dos variantes del k-medias generalizado (guiada por representante o guiada por mediana) son aplicables en el proceso.

Una vez obtenidos los k representantes mediante el algoritmo maxmin modificado, queda por realizar la aplicación de k-medianas o k-medias generalizado a los agrupamientos que generan esos k representantes iniciales, tal y como se ha descrito en la Sección 7.1.

En la Sección 7.3 se van a describir los experimentos hechos sobre el corpus *Copenhagen* descrito en el apartado 2.2.1 para verificar el efecto de las distintas formas de obtener los agrupamientos (teniendo en cuenta tanto el proceso de inicialización como la aplicación posterior de k-medianas y k-medias generalizado) y la influencia que tienen respecto a la calidad de los prototipos obtenibles a partir de dichos agrupamientos. Dicha calidad se comprobará mediante los resultados obtenidos a nivel de clasificación, especialmente comparando con el agrupamiento basado en k-medianas usado en el resto de experimentos realizados con este corpus.

7.3. Experimentos comparativos

En esta sección vamos a describir los experimentos hechos sobre el corpus *Copenhagen* descrito en el apartado 2.2.1 para verificar el efecto de las distintas formas de obtener los agrupamientos y la influencia que tienen respecto a la calidad de los prototipos obtenibles a partir de dichos agrupamientos. Los resultados van a estar en función del tipo de agrupamientos obtenidos, del número de dichos agrupamientos y del prototipo obtenido de cada uno de ellos, y básicamente se busca comparar los agrupamientos obtenidos mediante k-medianas con los obtenidos mediante k-medias generalizado. También se busca verificar la diferencia entre usar como inicialización el algoritmo maxmin clásico (Figura 1.2) y el modificado (Figura 7.2).

Los tipos de agrupamientos obtenidos se pueden subdividir en diversos tipos, según la inicialización aplicada y el posterior proceso de reagrupamiento. Las posibles inicializaciones son:

- Maxmin clásico (el usado en el resto de experimentos presentados previamente sobre el corpus *Copenhagen*)
- Maxmin modificado usando k-medianas
- Maxmin modificado usando k-medias generalizado guiado por representante y usando el método conjunto

- Maxmin modificado usando k-medias generalizado guiado por representante y usando el método separado
- Maxmin modificado usando k-medias generalizado guiado por mediana y usando el método conjunto
- Maxmin modificado usando k-medias generalizado guiado por mediana y usando el método separado

Los posteriores métodos de reagrupamiento a usar dependen también del tipo de inicialización usada, ya que dada una cierta inicialización hay métodos que resultan equivalentes. Por ejemplo, si se inicializa con maxmin modificado usando k-medianas, los representantes finales son las medianas de los agrupamientos. Por tanto, no hay diferencia entre usar posteriormente k-medias generalizado guiado por mediana o por representante (pues son lo mismo).

Tampoco se pretende realizar una experimentación exhaustiva de todas las combinaciones posibles, ya que estas son demasiadas. Las combinaciones efectuadas, junto con un código que las identifica, se presentan en el Cuadro 7.1.

A la vista de los procesos de obtención de agrupamiento propuestos, existen una serie de comentarios a realizar que se exponen en la siguiente lista:

- Para las combinaciones 2 y 3 se ha elegido el guiado por mediana ya que los representantes obtenidos de los agrupamientos son arbitrarios y su calidad como cadena inicial es dudosa.
- En la combinación 4, el reagrupamiento no causa ningún efecto (pues la última iteración de maxmin modificado nos da un conjunto de agrupamientos estables por k-medianas).
- En las combinaciones 5 y 6, es indiferente usar guiado por representante o por mediana (pues los representantes dados por la inicialización son las medianas de los agrupamientos).
- En las combinaciones 8 y 9 se ha usado el método conjunto por coherencia con la inicialización; además, la combinación 8 no provoca cambios en los agrupamientos (pues no puede mejorar los representantes actuales que ya han sido optimizados en la inicialización).
- En las combinaciones 11 y 12 se usa el método separado por coherencia con la inicialización; la combinación 11 tampoco provoca cambios en los agrupamientos.
- En las combinaciones 14 y 16 es indiferente usar k-medias generalizado guiado por mediana o por representante, ya que los representantes dados por la inicialización son la mediana o una cadena mejor que esta; por tanto, aunque se use el guiado por mediana, el representante dado por la inicialización va a ser siempre igual o mejor. El método de optimización es coherente con la inicialización (conjunto en la 14 y separado en la 16).

Cuadro 7.1: Combinaciones utilizadas para la obtención de los agrupamientos

Inicialización	Reagrupamiento	Id.
maxmin clásico	k-medianas	1
	k-medias generalizado guiado por mediana método conjunto	2
	k-medias generalizado guiado por mediana método separado	3
maxmin modificado usando k-medianas	k-medianas	4
	k-medias generalizado guiado por mediana/representante método conjunto	5
	k-medias generalizado guiado por mediana/representante método separado	6
maxmin modificado usando k-medias generalizado guiado por representante método conjunto	k-medianas	7
	k-medias generalizado guiado por representante método conjunto	8
	k-medias generalizado guiado por mediana método conjunto	9
maxmin modificado usando k-medias generalizado guiado por representante método separado	k-medianas	10
	k-medias generalizado guiado por representante método separado	11
	k-medias generalizado guiado por mediana método separado	12
maxmin modificado usando k-medias generalizado guiado por mediana método conjunto	k-medianas	13
	k-medias generalizado guiado por mediana/representante método conjunto	14
maxmin modificado usando k-medias generalizado guiado por mediana método separado	k-medianas	15
	k-medias generalizado guiado por mediana/representante método separado	16

Tras determinar las distintas formas de obtener los agrupamientos, hay que determinar cuántos agrupamientos se van a obtener. A fin de comparar los resultados de los experimentos con los presentados en el Capítulo 3, se ha usado el mismo rango de agrupamientos, es decir, $k = 1, 2, \ldots, 9, 10, 20, \ldots, 100$. Hay que destacar que para $k = 1$ los resultados con cualquier inicialización son equivalentes, pues tenemos un solo agrupamiento (todas las muestras de entrenamiento de la clase). De la misma manera, para $k = 100$ también se tienen los mismos agrupamientos (cada muestra es un agrupamiento en sí).

Finalmente, para los agrupamientos obtenidos se deben extraer los k prototipos a usar en la clasificación. Los prototipos que se extrajeron fueron la cadena mediana y las medias aproximadas por los dos métodos de perturbación (separado y conjunto). Dichos prototipos se usaron en los experimentos de clasificación de la misma forma que se ha hecho en el resto de capítulos con este corpus, usando un clasificador k-NN con $k = 1, 2, \ldots, 15$ (k es el número de vecinos, y no tiene relación en este caso con el número de agrupamientos).

En los siguientes apartados vamos a verificar la influencia de los agrupamientos en la extracción de los prototipos, dedicando un apartado a cada tipo de prototipo extraído (mediana, media aproximada por perturbación conjunta y media aproximada por perturbación separada).

7.3.1. Resultados usando la cadena mediana

En este apartado mostramos los resultados de clasificación usando la cadena mediana de cada uno de los agrupamientos obtenidos como prototipos. Estos resultados se comparan con los obtenidos en el Capítulo 3 para el conjunto de agrupamientos obtenido usando el algoritmo maxmin no modificado.

En las gráficas de la Figura 7.3 podemos ver el efecto de usar k-medianas y k-medias generalizado guiado por mediana sobre los agrupamientos originales, para clasificadores por 1, 6 y 12 vecinos y en el intervalo de 1 a 100 agrupamientos. Aunque no es demasiado apreciable, existe una ligera mejora para un número pequeño de agrupamientos (entre 10 y 20) cuando se usa el reagrupamiento por k-medias generalizado.

En las gráficas de la Figura 7.4 se muestran los resultados para 1, 6 y 12 vecinos en el caso de inicialización con maxmin modificado por k-medianas y reagrupamiento por k-medias generalizado, de nuevo en el intervalo de 1 a 100 agrupamientos. Aquí sí que es apreciable una amplia mejora con respecto a los agrupamientos originales en el intervalo de 10 a 30 agrupamientos, mejora que se va reduciendo a medida que se aumenta el número de vecinos del clasificador.

Por último, en las gráficas de la Figura 7.5 se muestran los resultados para los agrupamientos de 1 a 100 usando un clasificador por vecino más cercano para el resto de combinaciones de inicialización y reagrupamiento (las combinaciones de la 7 a la 16 del Cuadro 7.1, incluyendo siempre la 4 y la original para comparar). El comportamiento de estas combinaciones es muy irregular, aunque en general se comportan peor que la inicialización por maxmin modificado con k-medianas y reagrupamiento k-medianas (combinación 4) y que incluso los agrupamientos originales.

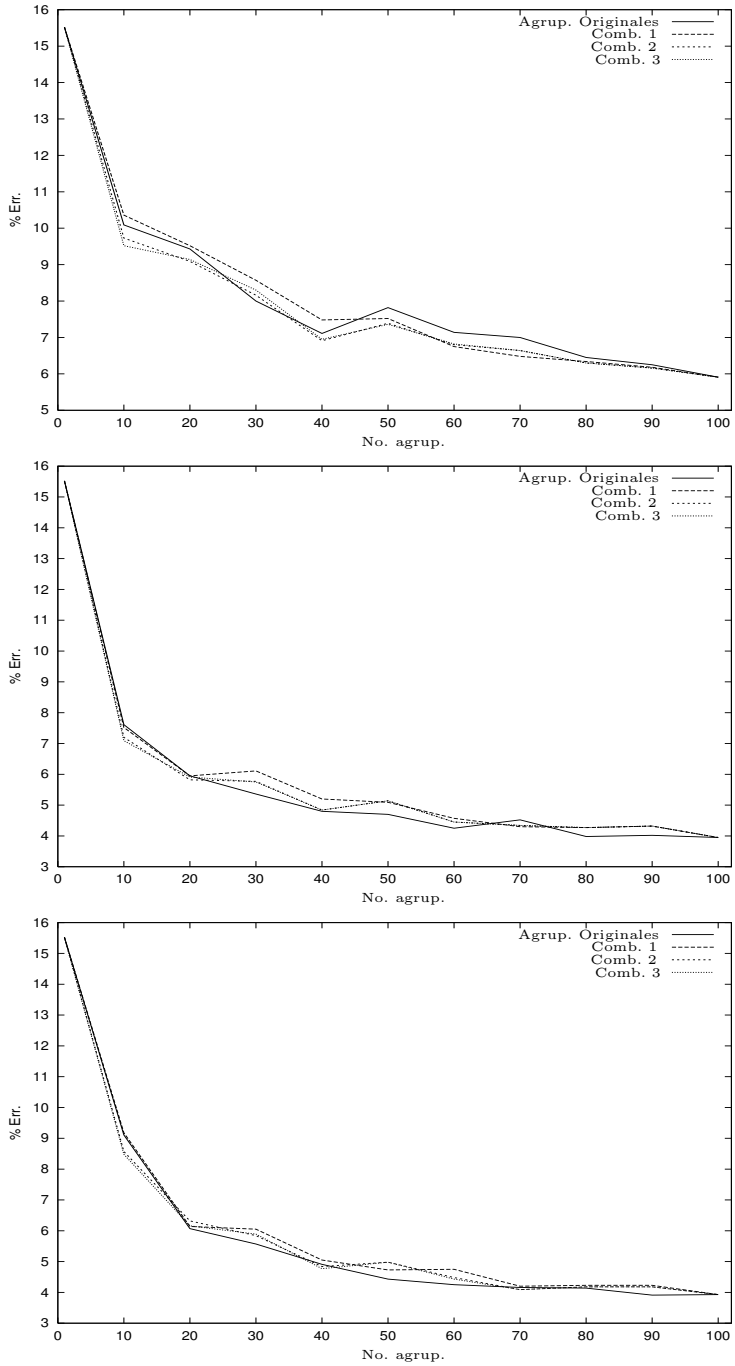


Figura 7.3: Resultados de clasificación para 1, 6 y 12-NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización por maxmin clásico y reagrupamiento por k-medianas y k-medias generalizado (combinaciones 1, 2 y 3). Se muestran los resultados con los agrupamientos originales para comparar.

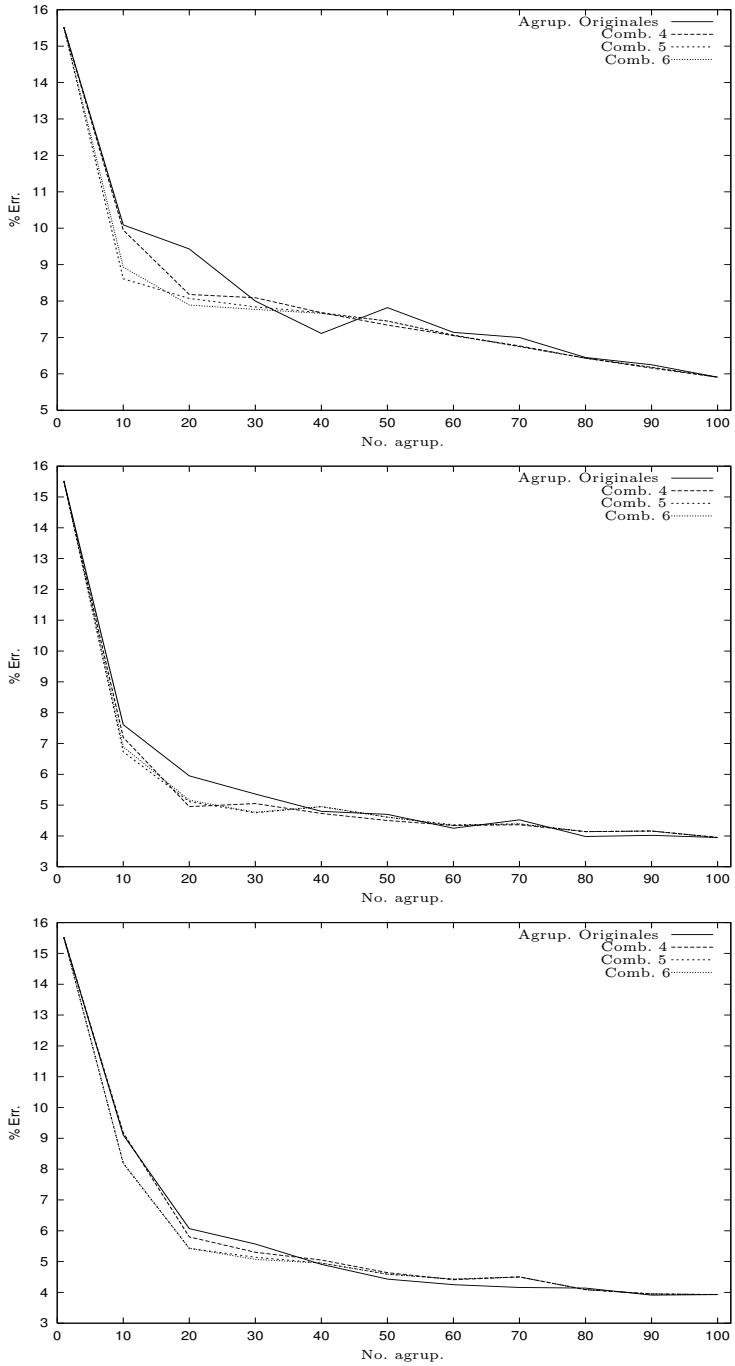


Figura 7.4: Resultados de clasificación para 1, 6 y 12-NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización de maxmin modificado usando k-medianas y el reagrupamiento por k-medianas y k-medias generalizado (combinaciones 4, 5 y 6). Se muestran los resultados con los agrupamientos originales para comparar.

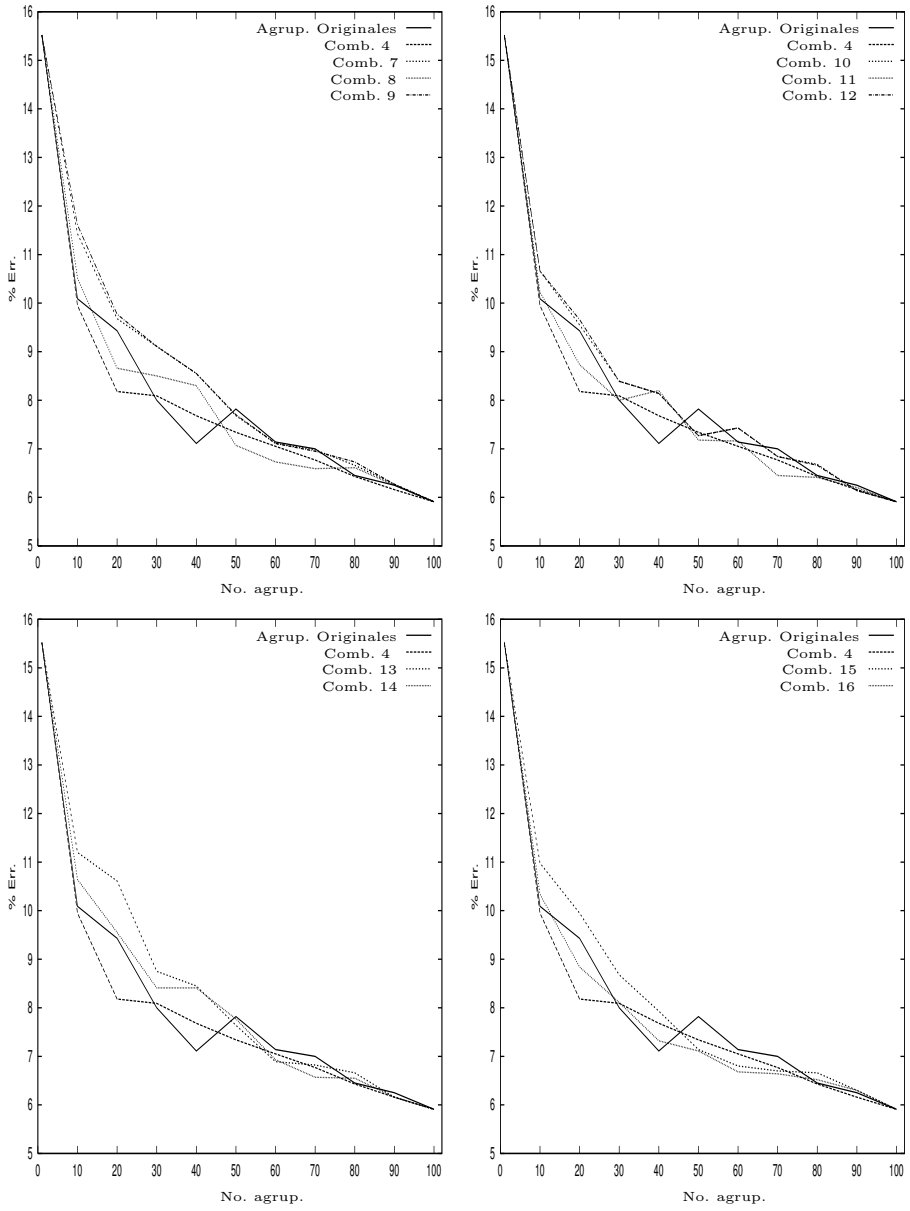


Figura 7.5: Resultados de clasificación para 1NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

De estas gráficas se deduce que los métodos que consiguen mejores agrupamientos para obtener como prototipos cadenas medianas son aquellos que usan la inicialización maxmin modificada con el método k-medianas, independientemente del método de reagrupamiento (k-medianas o k-medias generalizado) que se use después, siendo extensivo para diverso número de vecinos (gráficas de la Figura 7.4). También mejoran resultados sobre los agrupamientos originales el reagrupamiento por k-medias generalizado (Figura 7.3), y puntualmente otras combinaciones (tal y como se puede ver en las gráficas de la Sección A.7, Figuras A.12 y A.13).

En todos los casos, esta mejora se da para un número de agrupamientos reducido (menor o igual que 30 agrupamientos), y va reduciéndose a medida que se usan más vecinos para la clasificación. Así pues, para confirmar la ventaja de los métodos propuestos para un número reducido de agrupamientos, se hicieron experimentos de clasificación en el intervalo de 1 a 10 agrupamientos.

En las gráficas de la Figura 7.6 se muestran los resultados en ese intervalo para 1, 3 y 6 vecinos, usando la inicialización por maxmin clásico y los reagrupamientos por k-medianas y k-medias generalizado. En este caso sí que resulta ostensible la mejora de resultados que otorga hacer el reagrupamiento usando k-medias generalizado.

En las gráficas de la Figura 7.7 se muestran, también para el intervalo de 1 a 10 agrupamientos y para 1, 3 y 6 vecinos, los resultados con inicialización por maxmin modificado con k-medianas y reagrupamientos por k-medianas y k-medias generalizado. Se percibe que el reagrupamiento por k-medianas provoca peores resultados incluso que el agrupamiento original, mientras que k-medias generalizado sigue siendo, en general, una mejor opción.

Por último, en las gráficas de las Figuras 7.8 presentamos los resultados obtenidos para el intervalo de 1 a 10 agrupamientos con un clasificador 1NN de las combinaciones 7 a 16, donde de nuevo se aprecia el comportamiento irregular y, generalmente, de peor calidad, de dichas combinaciones.

Con estas gráficas se puede confirmar que únicamente mejoran los resultados, en líneas generales, para un número pequeño de agrupamientos, los métodos que usan inicialización por maxmin clásico o por maxmin modificado con k-medianas, usando posteriormente k-medias generalizado en el reagrupamiento. Hay que destacar que existen casos particulares en los que el agrupamiento usado en el resto de experimentos funciona mejor (puede verse claramente el mínimo para cinco agrupamientos en todas las gráficas, ventaja que se propaga cuando se usa k-medias generalizado sobre dicho conjunto de agrupamientos).

También existen otros agrupamientos que otorgan resultados equiparables al original (los que usan k-medias generalizado en la inicialización y k-medias generalizado guiado por representante en el paso de reagrupamiento, es decir, las combinaciones 8, 11, 14 y 16). Otro hecho destacable es la robustez ante la escasez de prototipos del agrupamiento original, de los que usan k-medias generalizado a partir de éste, y de los obtenidos usando k-medianas en inicialización y k-medias generalizado en el reagrupamiento: puede verse que usando un número de vecinos bastante mayor que el número de prototipos por clase extraídos (por ejemplo, seis vecinos con dos prototipos por clase), en todos los

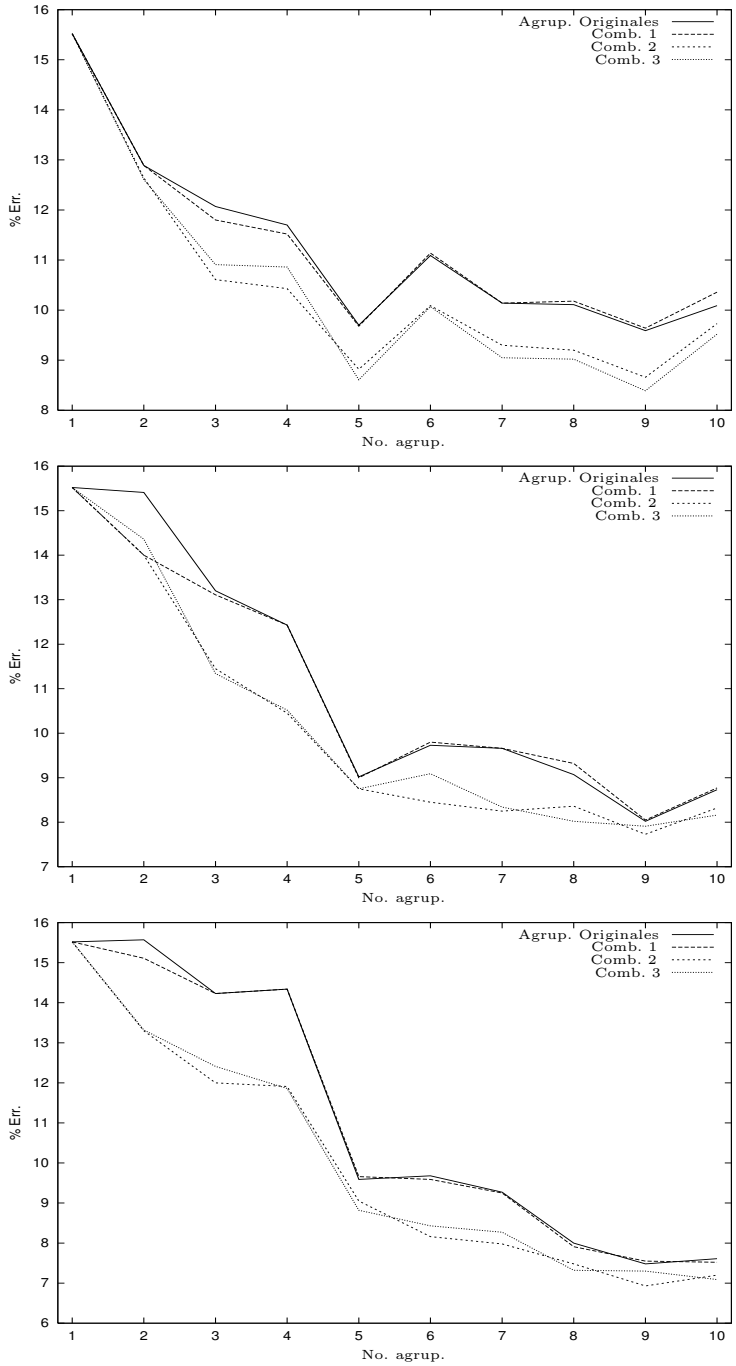


Figura 7.6: Resultados de clasificación para 1, 3 y 6-NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización por maxmin clásico y reagrupamiento por k-medianas y k-medias generalizado (combinaciones 1, 2 y 3). Se muestran los resultados con los agrupamientos originales para comparar.

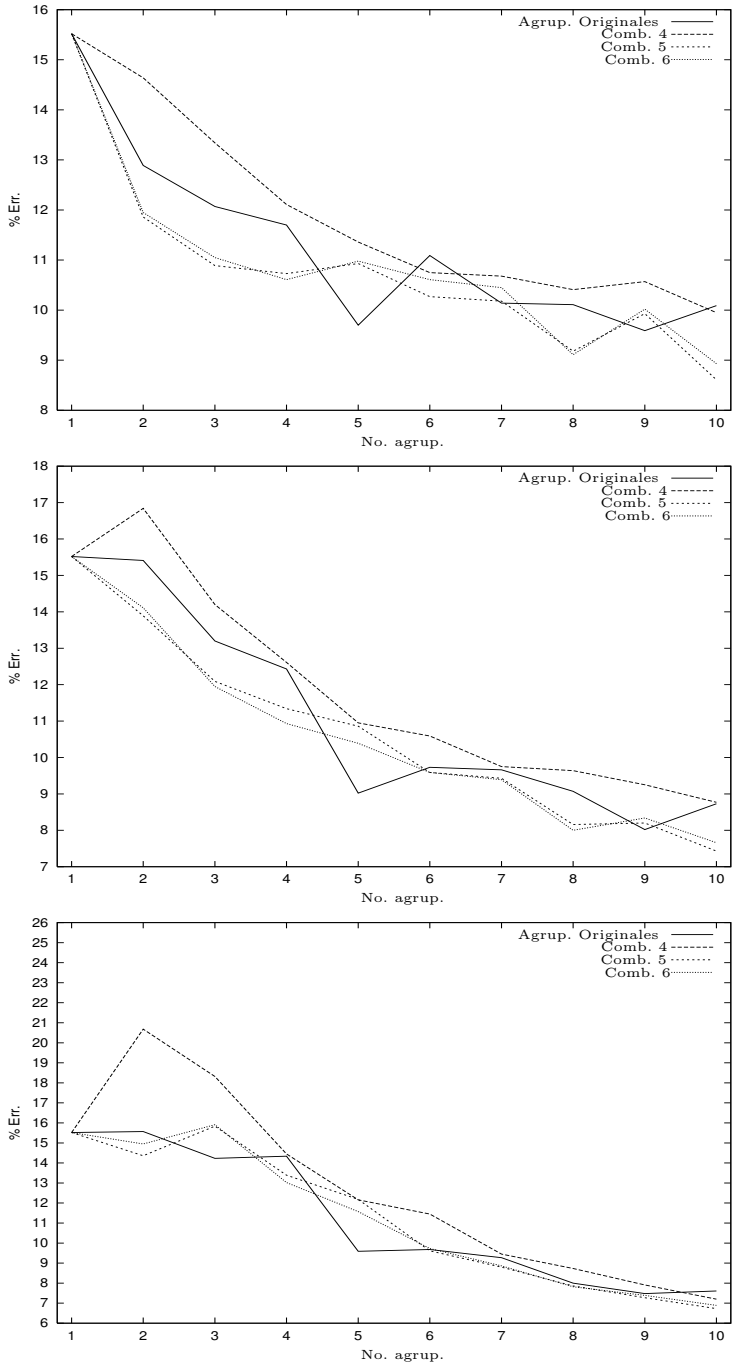


Figura 7.7: Resultados de clasificación para 1, 3 y 6-NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización de maxmin modificado usando k-medianas y el reagrupamiento por k-medianas y k-medias generalizado (combinaciones 4, 5 y 6). Se muestran los resultados con los agrupamientos originales para comparar.

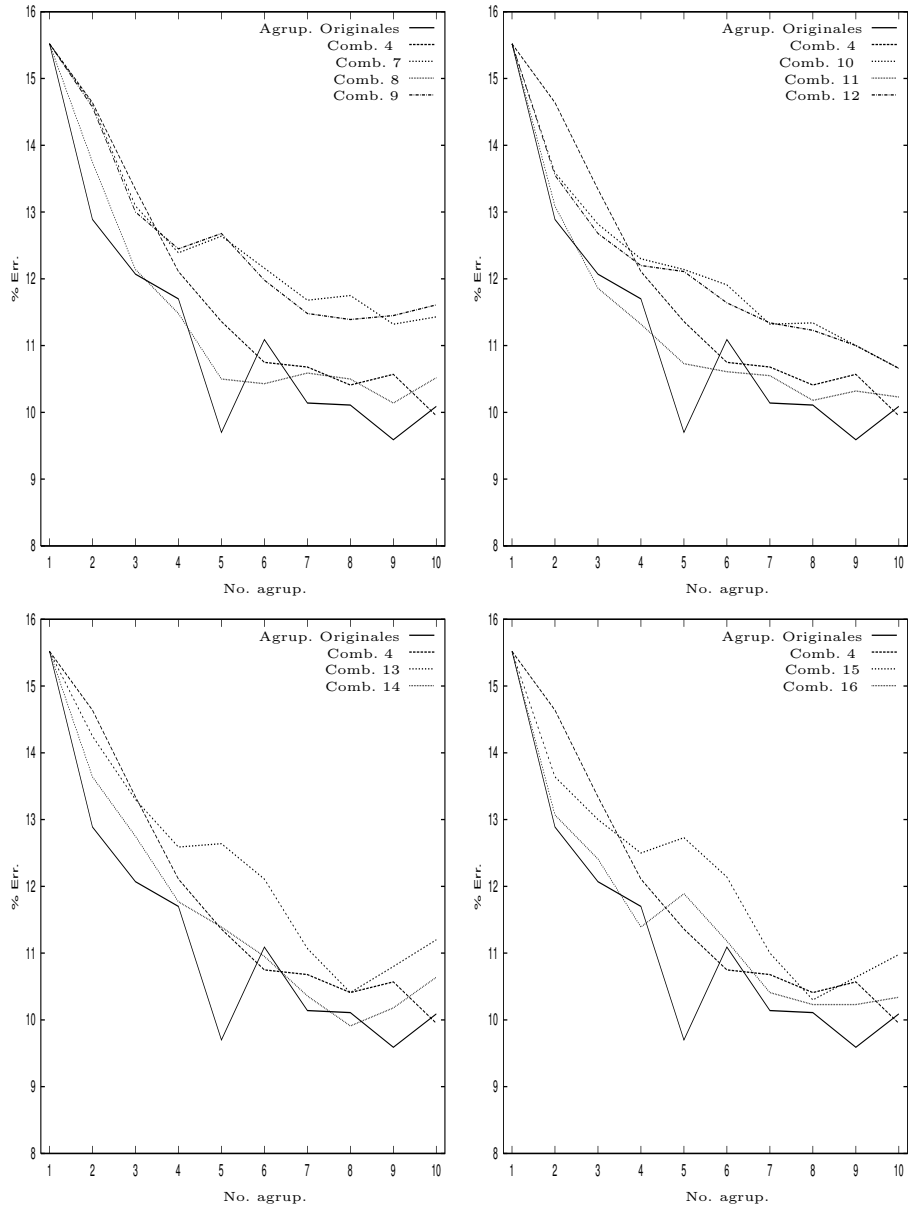


Figura 7.8: Resultados de clasificación para 1NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

casos excepto para dicha situación el error de clasificación se dispara (como se puede apreciar en la Figura 7.7 y en las que se muestran en la Sección A.7, Figuras A.14 y A.15).

Así pues, las conclusiones que parecen desprenderse de estos resultados son que de los diversos métodos, los que proporcionan un mejor resultado son los que usan maxmin clásico en la inicialización y k-medias generalizado (independientemente de usar la perturbación separada o conjunta), y que en todo caso estas mejoras se dan para un número de agrupamientos reducido y un número de hasta seis vecinos en el clasificador. También es destacable el buen comportamiento de las combinaciones que usan maxmin modificado con k-medianas en el proceso de inicialización.

Naturalmente, estos resultados se refieren únicamente a la extracción de la cadena mediana como prototipo y hay que confirmar con las cadenas medias aproximadas la bondad de los agrupamientos obtenidos para la adecuada extracción de prototipos.

7.3.2. Resultados usando la cadena media aproximada por perturbación conjunta

En este apartado se ofrecen los resultados de extraer de los agrupamientos la cadena media aproximada usando el método conjunto (el presentado en la Figura 3.6). De nuevo los resultados se compararán con los obtenidos a partir de los agrupamientos usados en las experimentaciones previas.

En la Figura 7.9 tenemos los resultados de usar k-medianas y k-medias generalizado guiado por mediana, extrayendo la cadena media aproximada por método conjunto. Se presentan los resultados para clasificadores por 1, 6 y 12 vecinos más cercanos en el intervalo de 1 a 100 agrupamientos. En este caso, la mejora dada por aplicar k-medias generalizado es realmente mínima con respecto a los agrupamientos originales. Sólo en el caso de clasificación por 1NN y un número alto de agrupamientos (a partir de 50) se percibe una mejora clara respecto a los agrupamientos originales, sea cual sea el proceso de reagrupamiento usado.

En cambio, si observamos las gráficas de la Figura 7.10 (resultados para 1, 6 y 12 vecinos en el intervalo 1-100 con inicialización por maxmin modificado con k-medianas), sí que es clara la mejora para un número pequeño de agrupamientos (intervalo de 10 a 30 agrupamientos). Los tres métodos de reagrupamiento parecen comportarse de manera semejante en dicho intervalo; sólo para la clasificación por vecino más cercano se observa una notable ventaja del k-medias generalizado respecto al k-medianas.

En cuanto al resto de posibles combinaciones, en las gráficas de la Figura 7.11 se muestran los resultados obtenidos para un clasificador por vecino más cercano en el intervalo de 1 a 100 agrupamientos. De nuevo se pone de manifiesto el comportamiento irregular de estas combinaciones y su peor comportamiento, en general, que la alternativa que usa maxmin modificado con k-medianas en la inicialización y reagrupamiento por k-medianas.

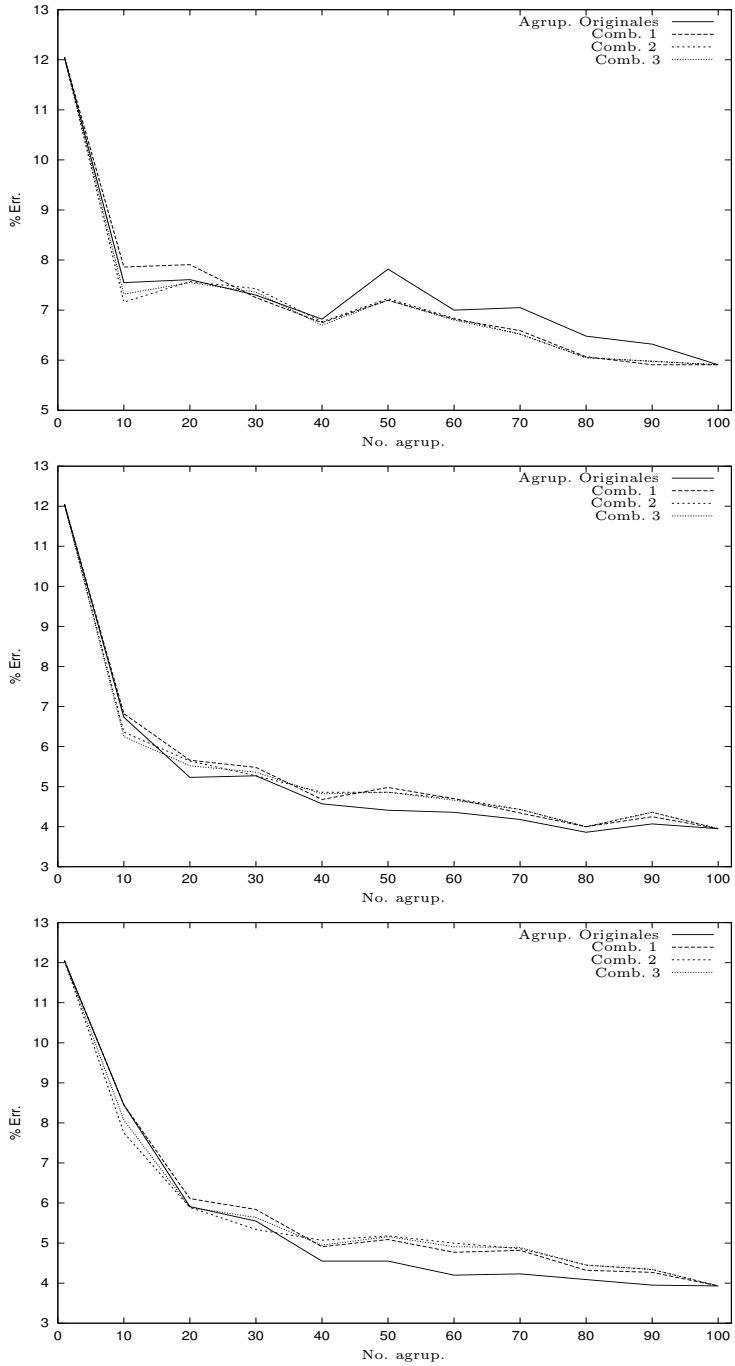


Figura 7.9: Resultados de clasificación para 1, 6 y 12-NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización por maxmin clásico y reagrupamiento por k-medianas y k-medias generalizado (combinaciones 1, 2 y 3). Se muestran los resultados con los agrupamientos originales para comparar.

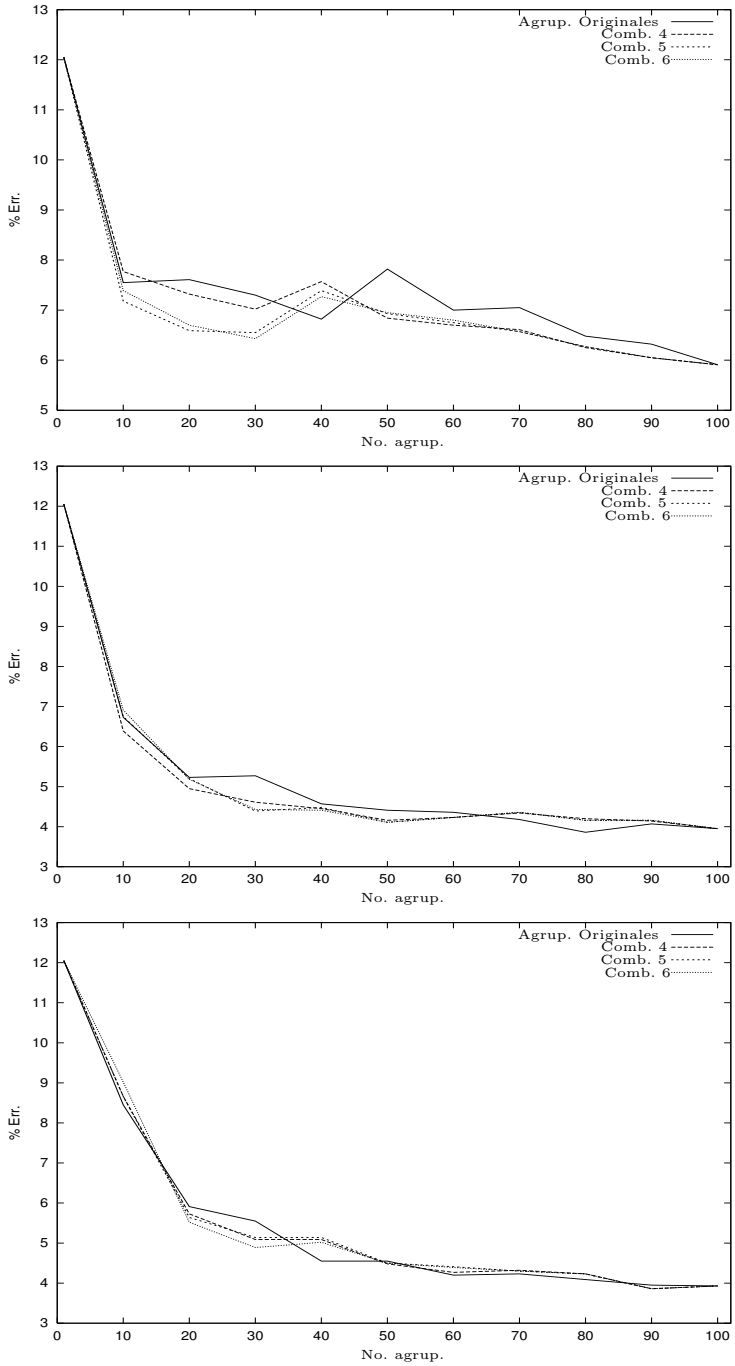


Figura 7.10: Resultados de clasificación para 1, 6 y 12-NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización de maxmin modificado usando k-medianas y el reagrupamiento por k-medianas y k-medias generalizado (combinaciones 4, 5 y 6). Se muestran los resultados con los agrupamientos originales para comparar.

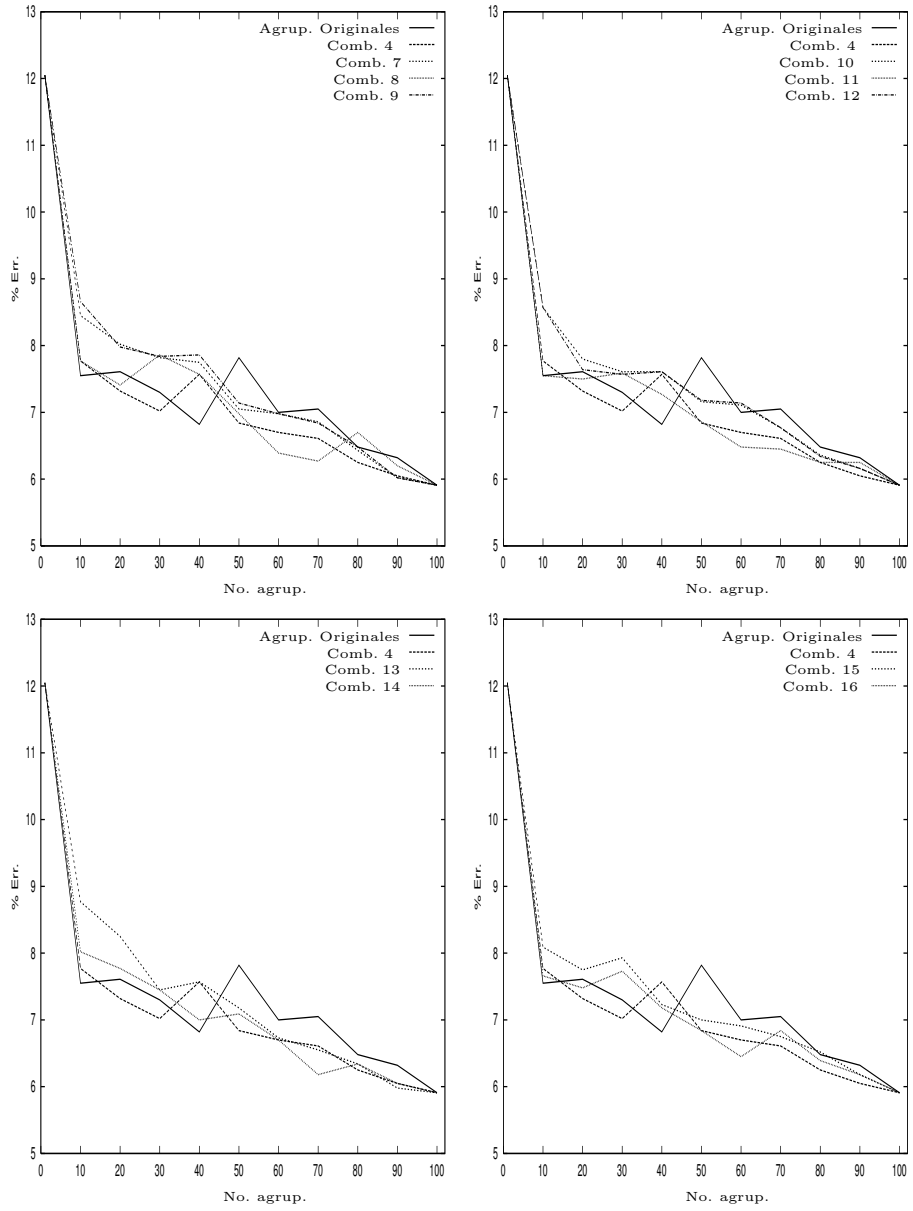


Figura 7.11: Resultados de clasificación para 1NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

De estos resultados se concluye que, en el caso de extraer cadena media aproximada como prototipo (usando en este caso el método conjunto), la mejor alternativa para realizar los agrupamientos es usar inicialización por maxmin modificado con k-medianas y reagrupamiento posterior usando k-medias generalizado (independientemente del método usado, separado o conjunto). Esto parece lo más adecuado para cualquier número de vecinos usado en el clasificador y un número reducido de agrupamientos (inferior a 30). En cualquier caso, van reduciéndose las diferencias a medida que se incrementa el número de vecinos del clasificador (como se puede apreciar también en las gráficas de la Sección A.8, de las Figuras A.16 y A.17).

De nuevo se llevaron a cabo los experimentos de clasificación para un número reducido de agrupamientos con el fin de verificar la influencia de los métodos usados en dicha situación. Los resultados obtenidos con inicialización por maxmin clásico y k-medianas y k-medias generalizado en el posterior reagrupamiento, para el intervalo de 1 a 10 agrupamientos y para 1, 3 y 6 vecinos, se presentan en las gráficas de la Figura 7.12. En dichos resultados se aprecia que, al igual que pasaba con la cadena mediana, se experimenta una mejora si los agrupamientos se obtienen usando el reagrupamiento por k-medias generalizado, aunque en este caso la mejora es mucho menos notable (al igual que pasaba en este mismo caso para el intervalo de 1 a 100 agrupamientos).

En cambio, en los resultados obtenidos usando inicialización por maxmin con k-medianas y reagrupamiento por k-medias generalizado (presentados en la Figura 7.13, de nuevo para 1, 3 y 6 vecinos), vemos como desaparece la ventaja de los prototipos obtenidos a partir de esos agrupamientos con respecto a los agrupamientos originales. Más aún, la robustez que teníamos cuando se extraía la cadena mediana en esta situación (Figura 7.7) se pierde al extraerse la cadena media aproximada (es decir, ahora utilizar más vecinos que prototipos hay por clase sí que degrada los resultados, como se puede ver en el caso de 6NN).

De nuevo los resultados obtenidos para el resto de combinaciones (gráficas de la Figura 7.14) para clasificadores por vecino más próximo muestran la ineficacia de las mismas también en esta situación.

Viendo los resultados aportados, resulta evidente que sólo los métodos basados en inicialización por maxmin clásico o por maxmin modificado con k-medianas, y con posterior reagrupamiento por k-medias generalizado (en cualquiera de sus dos variantes) dan resultados comparables (o incluso mejores) a los de agrupamientos originales. En el resto de técnicas, el error de clasificación resulta mayor, como se puede ver en la Sección A.8 (Figuras A.18 y A.19).

También es destacable la pérdida de la robustez de los prototipos que teníamos en el caso de extraer cadena mediana: al extraer la cadena media aproximada, si el número de prototipos por clase es menor que el número de vecinos empleado en el clasificador, el error de clasificación se dispara ostensiblemente (como se ve con claridad en la última gráfica de la Figura 7.13 y en las de la Figura A.19 que usan un 6NN). La única situación en la que los resultados no se degradan más que los agrupamientos originales (en líneas generales) es la que usa la inicialización por maxmin clásico y reagrupamiento por k-medias generalizado (Figura 7.12).

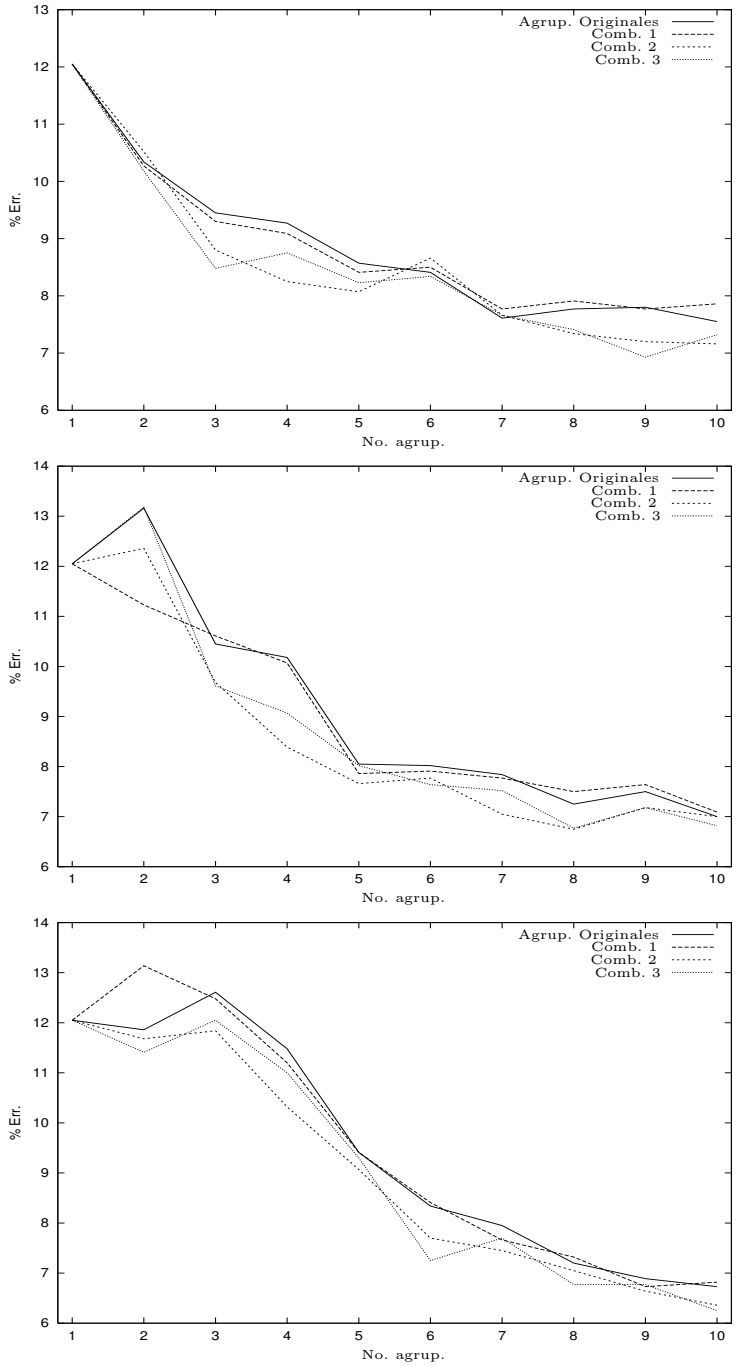


Figura 7.12: Resultados de clasificación para 1, 3 y 6-NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización por maxmin clásico y reagrupamiento por k-medianas y k-medias generalizado (combinaciones 1, 2 y 3). Se muestran los resultados con los agrupamientos originales para comparar.

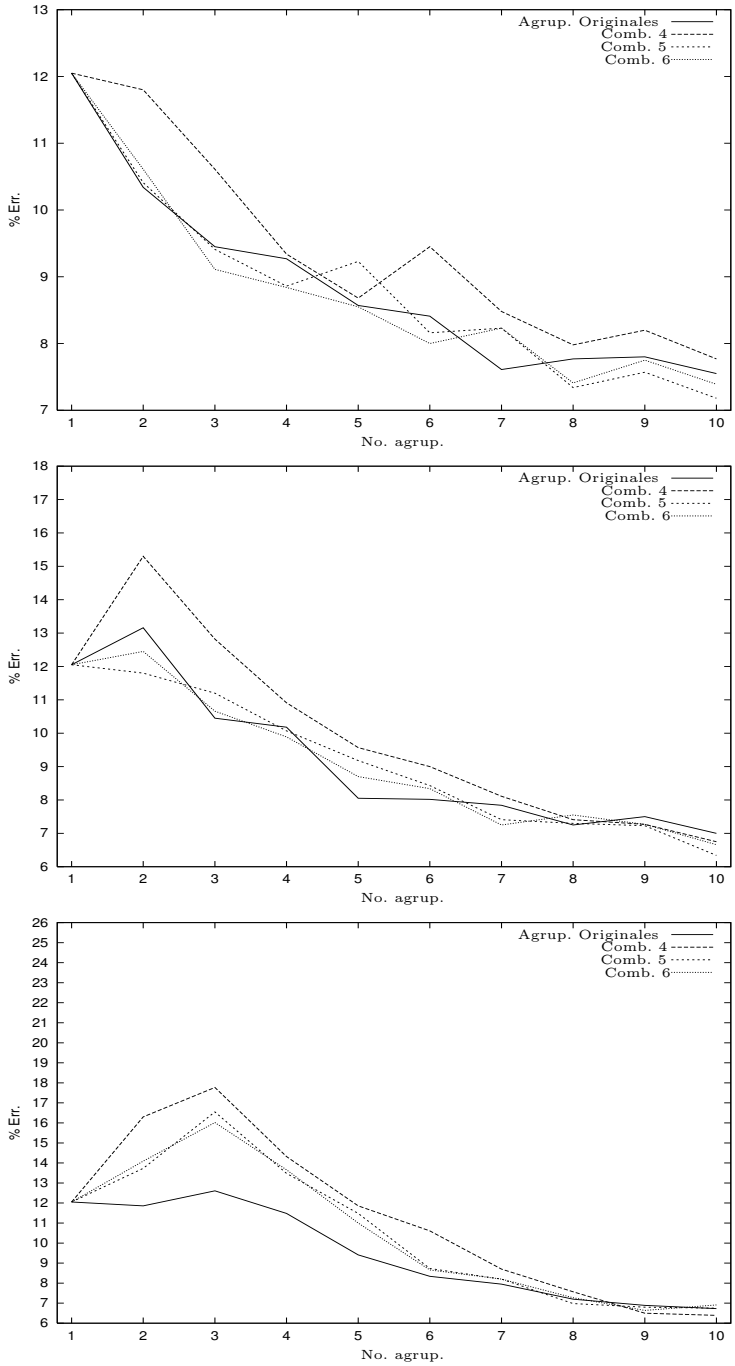


Figura 7.13: Resultados de clasificación para 1, 3 y 6-NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización de maxmin modificado usando k-medianas y el reagrupamiento por k-medianas y k-medias generalizado (combinaciones 4, 5 y 6). Se muestran los resultados con los agrupamientos originales para comparar.

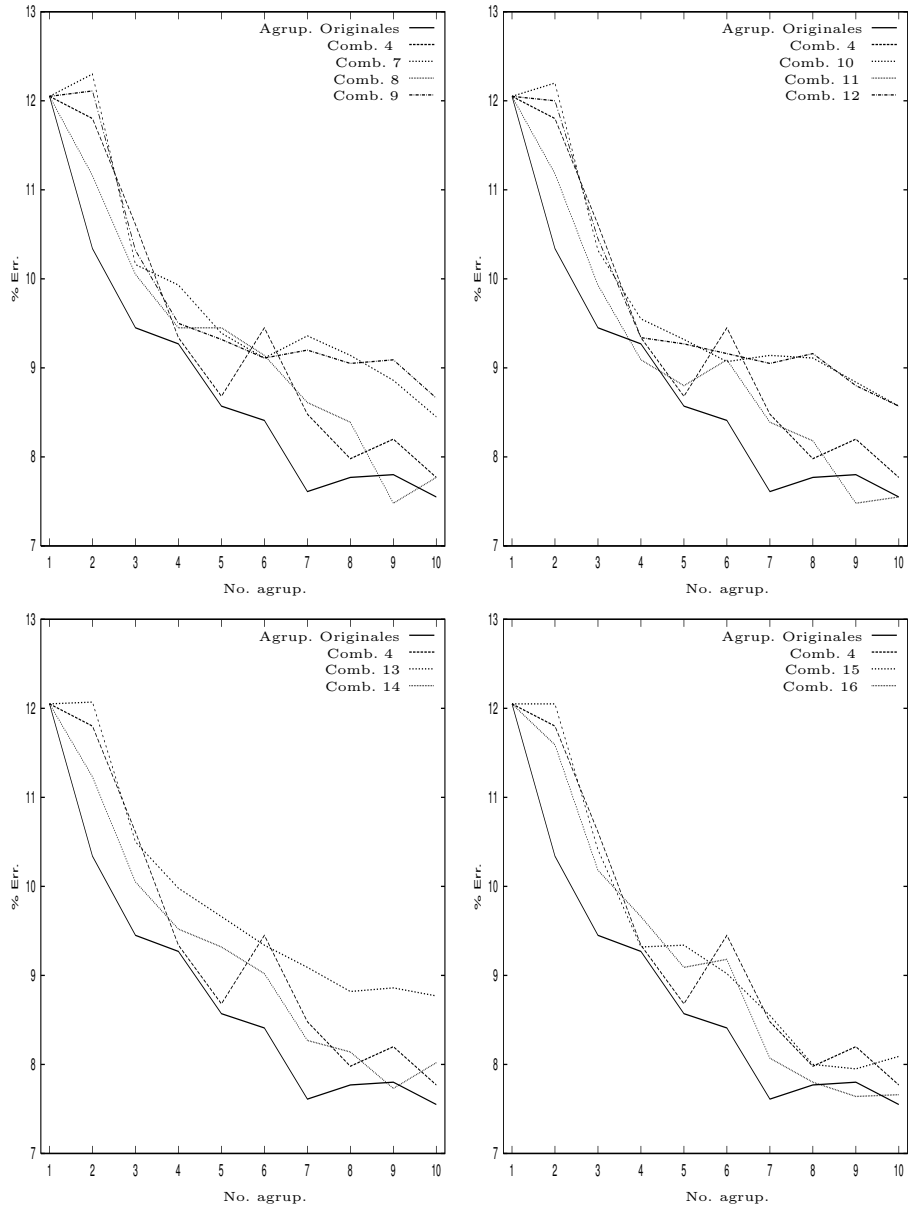


Figura 7.14: Resultados de clasificación para 1NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

De todos estos resultados parece deducirse que de nuevo los mejores métodos para obtener los agrupamientos son los basados en inicialización por k-medianas con el maxmin modificado, y posterior reagrupamiento con el método k-medias generalizado. Sin embargo, si esto parecía una regla general al usar la cadena mediana como prototipo (o al menos asumible para cualquier número de agrupamientos por debajo de un cierto número), en el caso de obtener la media aproximada por el método conjunto vemos que este rango en el que se produce la mejora queda definido entre un mínimo de 10 agrupamientos y un máximo de 40, dependiendo también este rango del número de vecinos del clasificador en cualquier caso.

En el caso de utilizar inicialización por maxmin clásico y reagrupamiento por k-medias generalizado no se da este fenómeno de degradación, pero a cambio las mejoras producidas en otras situaciones no son tan altas. De nuevo también se da el fenómeno de que las diferencias entre las clasificaciones obtenidas a partir de los distintos agrupamientos van reduciéndose al aumentar el número de vecinos del clasificador.

Por último, en el siguiente apartado vamos a mostrar los resultados usando el método de perturbación separado para obtener la cadena media aproximada, tratando de verificar si de nuevo no existen diferencias significativas entre usar un método u otro o si las diferentes formas de obtener los agrupamientos sí influyen en la calidad de hacerlo.

7.3.3. Resultados usando la cadena media aproximada por perturbación separada

En este apartado ofrecemos los resultados obtenidos al extraer como prototipo la cadena media aproximada usando el método perturbativo separado (el presentado en las Figuras 3.3 y 3.4). Se mostrarán los resultados tanto para los nuevos agrupamientos obtenidos como para los agrupamientos usados en los experimentos de los Capítulos 3, 4 y 5.

Presentamos en la Figura 7.15 los resultados para 1, 6 y 12 vecinos, con agrupamientos de 1 a 100, de los agrupamientos obtenidos con inicialización por maxmin clásico y reagrupamiento por k-medianas y k-medias generalizado. Igualmente, en la Figura 7.16 se presentan los resultados en las mismas condiciones (1, 6 y 12 vecinos, de 1 a 100 agrupamientos) para inicialización por maxmin modificado con k-medianas y reagrupamiento por k-medianas y k-medias generalizado. Por último, en la Figura 7.17 se muestran los resultados para los agrupamientos desde 1 hasta 100 y un vecino más próximo para el resto de combinaciones.

Los resultados obtenidos son muy semejantes a los que se obtuvieron con el método conjunto. Por tanto, las conclusiones que se pueden extraer son las mismas: las mejores inicializaciones son maxmin clásico y, sobre todo, maxmin modificado con k-medianas, con posterior reagrupamiento usando k-medias generalizado y las ventajas que obtienen dichos métodos se van reduciendo a medida que aumenta el número de vecinos. El incremento del número de vecinos presenta en este caso diferencias algo menos acusadas (en todos los casos, como

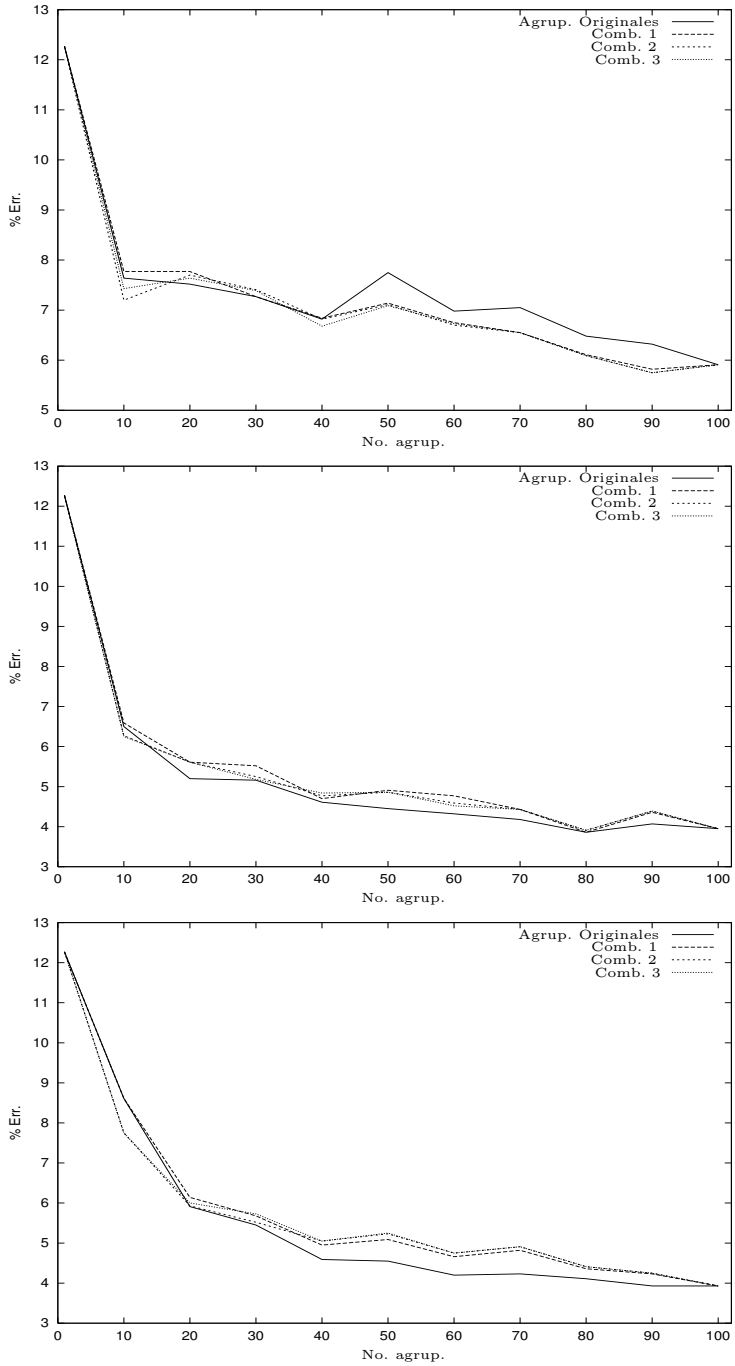


Figura 7.15: Resultados de clasificación para 1, 6 y 12-NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización por maxmin clásico y reagrupamiento por k-medianas y k-medias generalizado (combinaciones 1, 2 y 3). Se muestran los resultados con los agrupamientos originales para comparar.

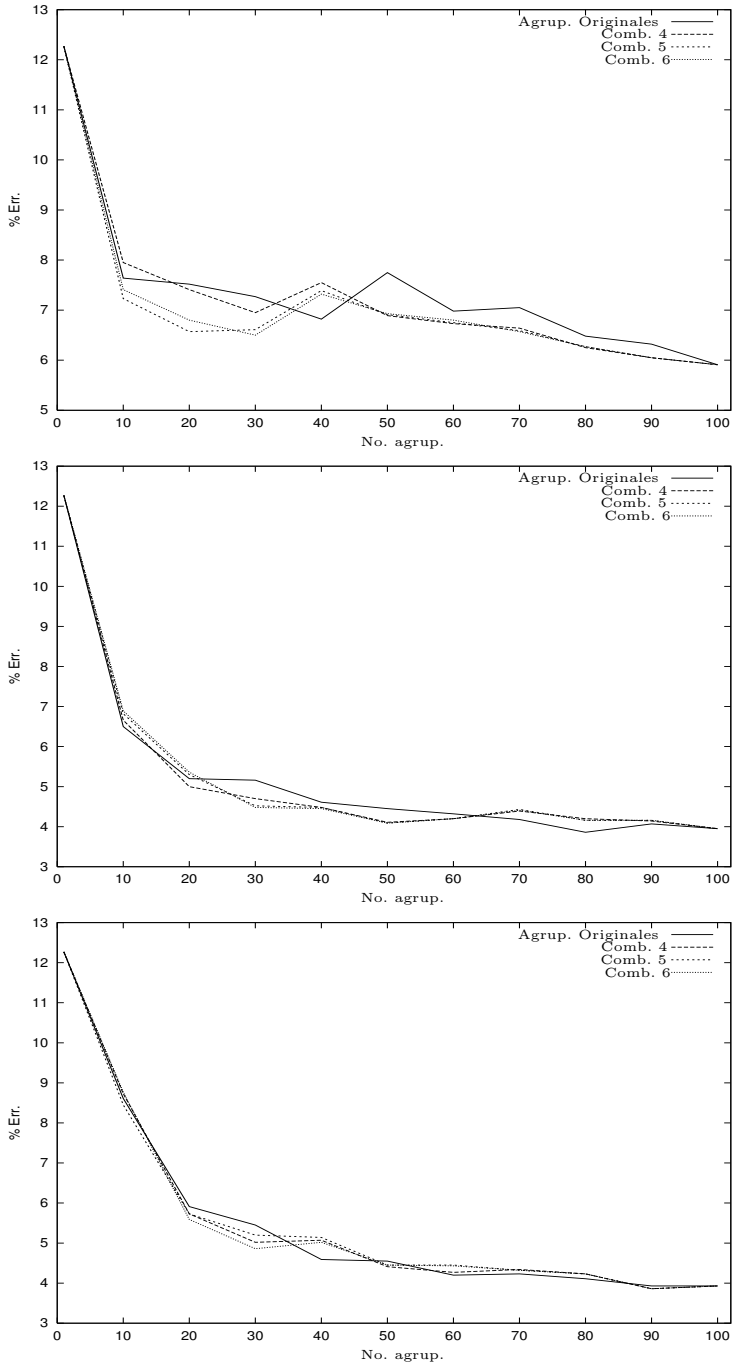


Figura 7.16: Resultados de clasificación para 1, 6 y 12-NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización de maxmin modificado usando k-medianas y el reagrupamiento por k-medianas y k-medias generalizado (combinaciones 4, 5 y 6). Se muestran los resultados con los agrupamientos originales para comparar.

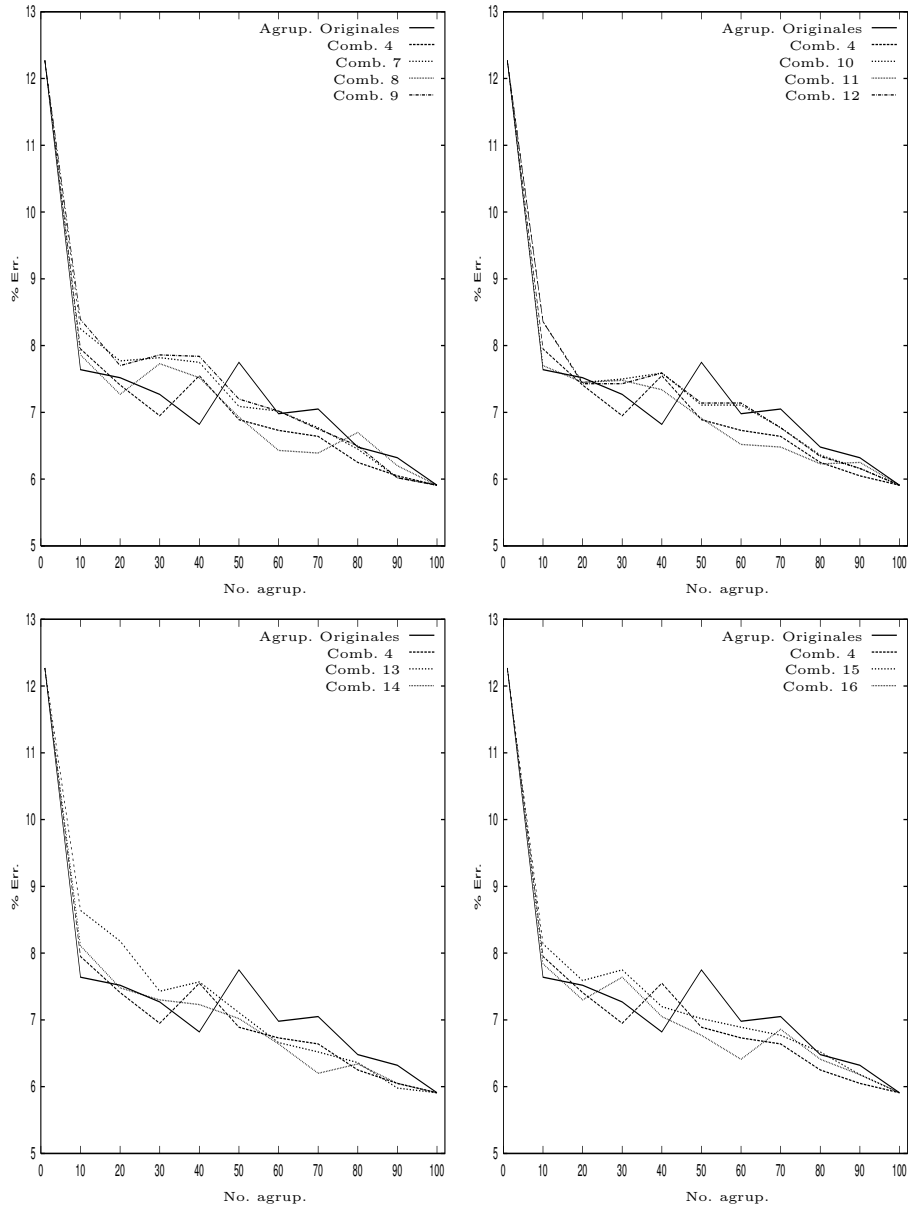


Figura 7.17: Resultados de clasificación para 1NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

se observa en las gráficas de la Sección A.8, Figuras A.16 y A.17) que en los resultados dados por el método conjunto.

Igualmente se realizaron experimentos de clasificación para un número reducido de agrupamientos. En las gráficas de la Figura 7.18 se presentan los resultados para inicialización por maxmin clásico y reagrupamiento k-medianas y k-medias generalizado para 1, 3 y 6 vecinos y en el intervalo de 1 a 10 agrupamientos. En la Figura 7.19 tenemos los resultados en dichas condiciones para inicialización por maxmin modificado con k-medianas y reagrupamiento por k-medianas y k-medias generalizado. Por último, las gráficas de la Figura 7.20 muestran los resultados para un clasificador por vecino más cercano del resto de combinaciones del Cuadro 7.1.

Se observa de nuevo un comportamiento muy semejante al que se da usando el método conjunto. De nuevo, los únicos métodos de agrupamientos que mejoran resultados son los que inicializan con maxmin clásico y reagrupan con k-medias generalizado. Otorgan resultados comparables los que usan la inicialización por maxmin modificado con k-medianas y el reagrupamiento por k-medias generalizado. En el resto de métodos el error de clasificación aumenta de forma aún más acusada que en el caso de utilizar el método conjunto en la extracción de la cadena media aproximada (gráficas de la Sección A.9, de las Figuras A.22 y A.23).

De nuevo también hay que destacar que la mejora producida por el método que usa inicialización k-medianas y reagrupamiento k-medias generalizado queda reducida al rango de 10 a 40 agrupamientos, al igual que al usar el método conjunto. Por tanto, parece que la equivalencia de los métodos separado y conjunto a nivel de calidad de los prototipos extraídos queda confirmada en estos experimentos.

Otras circunstancias que también se observaban en los experimentos realizados con la cadena media aproximada usando el método conjunto vuelven a aparecer en estos resultados. La primera de ellas es la falta de robustez de los prototipos cuando el número de vecinos del clasificador supera al número de prototipos por clase, y la segunda es la disminución de diferencias entre los errores de clasificación entre los diversos tipos de agrupamiento a medida que se incrementa en número de vecinos usados en el clasificador.

Por tanto, en líneas generales se puede concluir que de los métodos de obtención de agrupamientos propuestos, únicamente ofrecen ventajas los que se basan en inicialización por maxmin clásico y por maxmin modificado con k-medianas y la posterior reorganización por k-medias generalizado, sin haber diferencias significativas entre usar el método separado o conjunto en este último paso. Además, estas ventajas sólo se manifiestan en un rango del número de agrupamientos posibles, que en el caso de corpus *Copenhagen* es de 10 a 40 agrupamientos, rango que se extiende para cualquier número de agrupamientos inferior a 40 si el prototipo seleccionado es la cadena mediana. A partir de 40 agrupamientos, los resultados con estos agrupamientos resultan similares a los obtenidos con los agrupamientos usados en los experimentos previos.

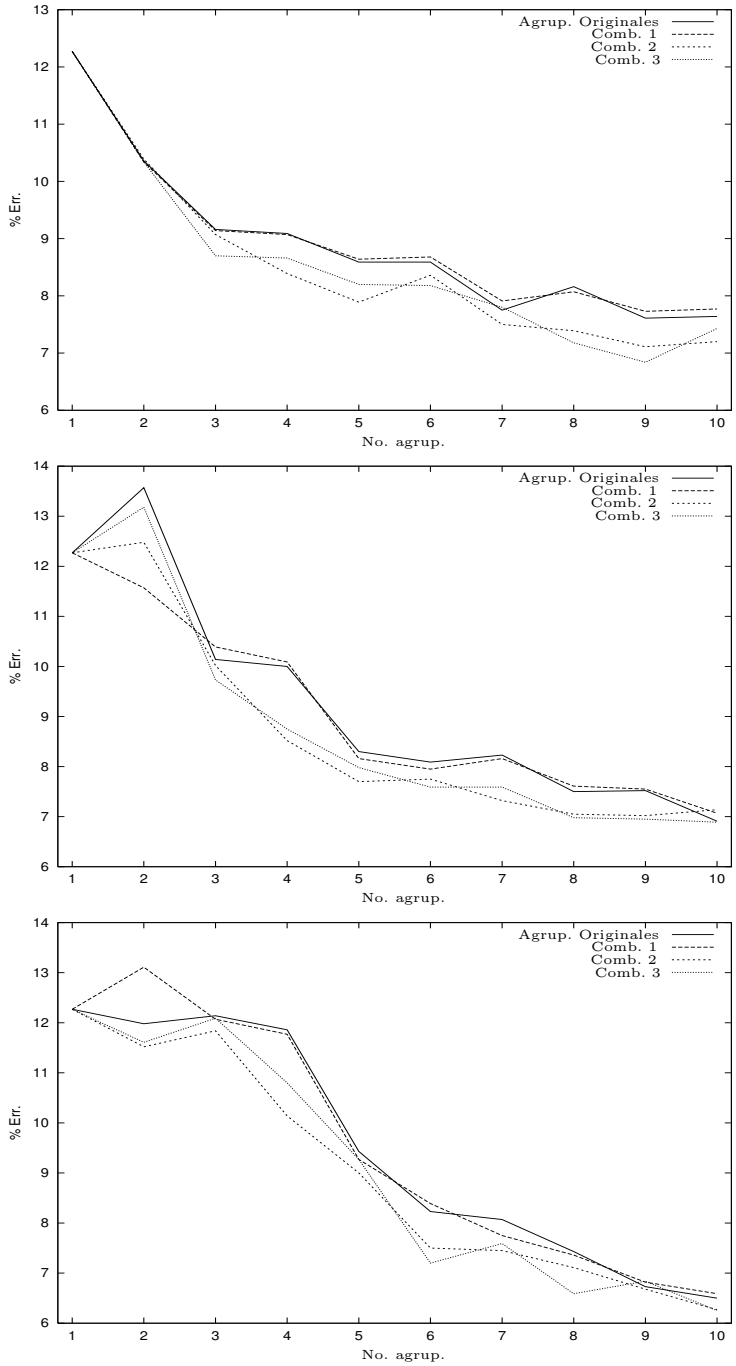


Figura 7.18: Resultados de clasificación para 1, 3 y 6-NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización por maxmin clásico y reagrupamiento por k-medianas y k-medias generalizado (combinaciones 1, 2 y 3). Se muestran los resultados con los agrupamientos originales para comparar.

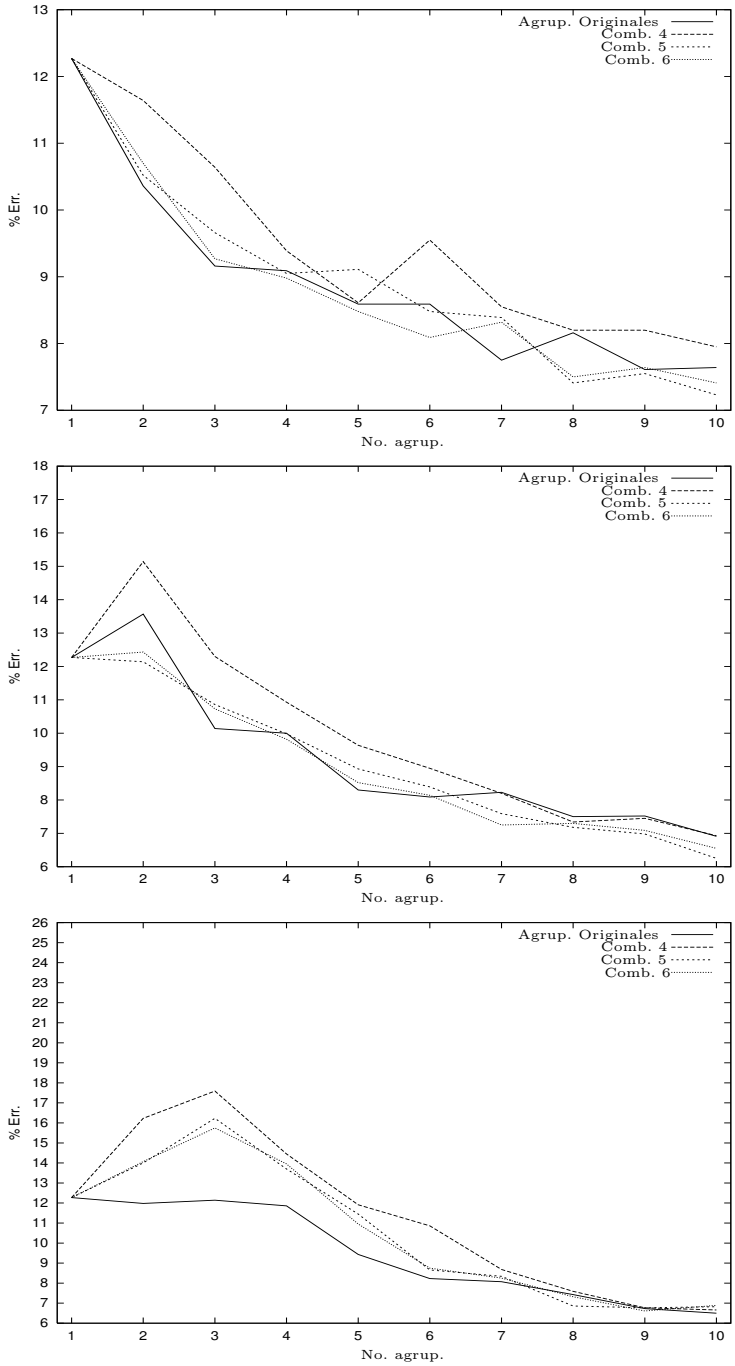


Figura 7.19: Resultados de clasificación para 1, 3 y 6-NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización de maxmin modificado usando k-medianas y el reagrupamiento por k-medianas y k-medias generalizado (combinaciones 4, 5 y 6). Se muestran los resultados con los agrupamientos originales para comparar.

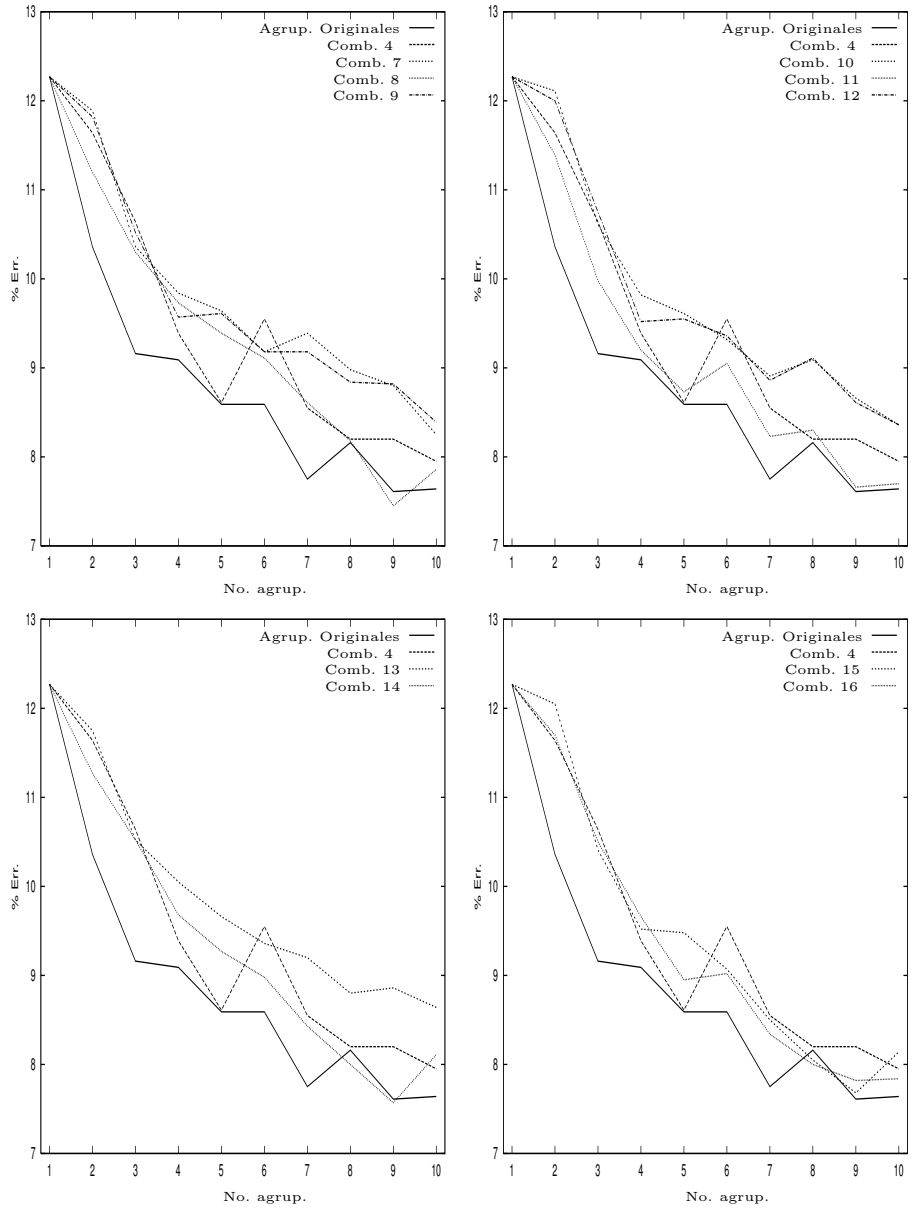


Figura 7.20: Resultados de clasificación para 1NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

7.4. Resumen

En este capítulo hemos abordado la posibilidad de usar las aproximaciones a la cadena media en los métodos de obtención de agrupamientos. Inspirándonos en el método k-medianas, hemos propuesto el método k-medias generalizado, el cual usa las aproximaciones a la cadena media en el proceso de reagrupamiento. Todos estos métodos requieren de un conjunto inicial de k muestras para poder aplicarse, conjunto que puede obtenerse mediante diversos métodos, de entre los cuales se ha escogido el método maxmin, usado en el resto de experimentos realizados. Además, hacemos una nueva propuesta de inicialización consistente en una modificación del maxmin que incluye reagrupamiento (por k-medianas o k-medias generalizado) en cada iteración del algoritmo. Se han explorado muy diversas combinaciones de inicialización y reagrupamiento y se han obtenido agrupamientos por cada una de ellas para el corpus *Copenhagen*. Los experimentos de clasificación han demostrado que las combinaciones que inicializan usando maxmin clásico o maxmin modificado con k-medianas y reagrupan con k-medias generalizado presentan mejor comportamiento para un cierto rango del número de agrupamientos que el resto de alternativas y que los agrupamientos obtenidos usando simplemente maxmin clásico.

Capítulo 8

La cadena media cíclica

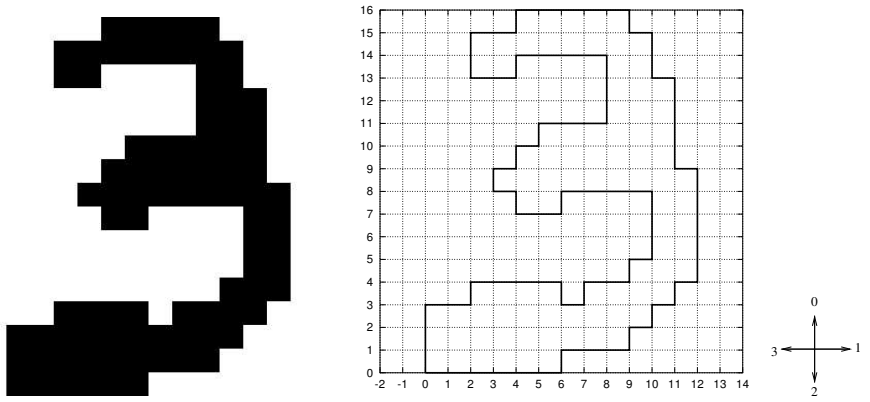
En este capítulo se aborda la extensión del concepto de cadena media a los conjuntos de cadenas cíclicas, es decir, a aquellas que representan objetos (generalmente bidimensionales) de manera invariante a la rotación que estos presenten. Se comprobará experimentalmente la ventaja de usar las aproximaciones a la cadena media frente a la cadena mediana como prototipos en clasificación para cadenas cíclicas.

8.1. Cadenas cíclicas y sus medidas de distancia

Una de las formas habituales de representar un objeto mediante una cadena es hacerlo determinando su cadena de contorno [16]. En estos casos, de la imagen del objeto se toma un punto inicial en el borde del mismo y se va rodeando el objeto por desplazamientos píxel a píxel. Cada uno de estos desplazamientos se hace en una cierta dirección previamente definida, de manera que se aproxime lo más posible al contorno real de la imagen. Cada una de estas direcciones tiene asociado un símbolo, y el recorrido de la imagen completa (acabando en el píxel inicial) genera la cadena asociada a la secuencia de direcciones que ha sido empleada. Un ejemplo de código de contorno de 4 direcciones, junto con la cadena que resulta, se presenta en la Figura 8.1.

Este proceso presenta la característica de que la cadena obtenida depende del píxel inicial que se haya escogido para el recorrido. Evidentemente, si los inicios son distintos, las cadenas pueden ser distintas. Por tanto, tenemos el problema de que el mismo objeto se puede representar por distintas cadenas dependiendo del inicio del recorrido del contorno.

Una primera solución es determinar de manera arbitraria un píxel inicial genérico para cualquier imagen. Opciones habituales son tomar el píxel más inferior o más superior, o más a la izquierda o a la derecha, de la figura representada en la imagen. Pero esta opción sigue sin resolver realmente el problema de que se asigne más de una cadena a un objeto. Esto se debe a que para un mismo objeto, según su ángulo de rotación, el píxel inicial con respecto al contorno



Cadena: 0001101111210110100003333233030101011100033332330011011111212212222122222323233233333333

Figura 8.1: Ejemplo de código de contorno con 4 direcciones y de la cadena resultante

variará.

La solución al problema pasa por hacer una representación de cadenas de contorno que sea invariante a rotaciones [20], de manera que para un mismo objeto, independientemente de la rotación que presente, la codificación sea la misma. Así es como surge el concepto de cadena cíclica, que es aquella cadena de contorno invariante a rotaciones. Una cadena cíclica se puede obtener a partir de una cadena de contorno habitual haciendo una transformación que se basa en codificar cambios relativos de ángulos, en vez de cambios absolutos, en función de la longitud del contorno [20].

La definición formal de cadena cíclica se hace mediante la definición de una clase de equivalencia. Para ello, se define el desplazamiento cíclico de la cadena $s = s_1 \cdot s_2 \cdots s_l$ como $\sigma(s) = \sigma(s_1 \cdot s_2 \cdots s_l) = s_2 \cdot s_3 \cdots s_l \cdot s_1$. Si se nota por σ^k la composición de k desplazamientos cíclicos, entonces se define la cadena cíclica s como la clase de equivalencia que define la relación $s \equiv s' \Leftrightarrow s' = \sigma^k(s)$ para un cierto k [55]. A partir de ahora notaremos simplemente por s la clase de equivalencia que indica la cadena cíclica generada a partir de s.

Respecto a las medidas de disimilitud entre cadenas cíclicas, se demostró en su día para la distancia de edición [45] que la distancia de edición entre las cadenas cíclicas s y t, notada por $d_C(s,t)$, cumple la propiedad:

$$d_C(s,t) = \min_{0 \leq k < |t|} d(s, \sigma^k(t)) \tag{8.1}$$

El algoritmo trivial para calcular esta distancia tendrá un coste $O(|s||t|^2)$, ya que para cada una de los posibles desplazamientos cíclicos de t, que son en total $|t|$, se debe calcular la distancia de edición entre s y $\sigma^k(t)$, lo cual es de coste $O(|s||t|)$. Existen, sin embargo, algoritmos más eficientes para proporcionar el valor exacto de dicha distancia, como es el algoritmo de Maes [45], de complejidad $O(|s||t| \log |t|)$, el algoritmo de Gregor y Thomason [23] y el algoritmo

de Ramificación y Poda de Marzal y Barrachina [52]. También existen diversos algoritmos aproximados, que sacrifican la obtención de la distancia exacta a cambio de reducir el coste computacional a un orden $O(|s||t|)$. Es el caso de los algoritmos de Bunke y Bühler [6] o los propuestos por Mollineda [56, 57].

La adaptación de esta definición de distancia entre cadenas cíclicas para otras medidas de disimilitud, y más concretamente para la distancia de edición normalizada, es inmediata: se basa en calcular el valor de la distancia escogida para la cadena s y todas las posibles rotaciones de t, tomando finalmente la menor de las distancias computadas como dicha distancia cíclica [52]. Así pues, seguiremos manteniendo el uso de la distancia de edición normalizada con cadenas cíclicas.

8.2. Cadena media cíclica

A partir de la definición de distancia de edición entre cadenas cíclicas, la definición de cadena mediana y cadena media de un conjunto de cadenas cíclicas resulta inmediata. Así, dado un conjunto de cadenas cíclicas $S = \{s^1, s^2, \ldots, s^n\}$ sobre Σ^*, la cadena mediana cíclica de S, notada por sm_S^C, vendrá dada por:

$$sm_S^C = \underset{s \in S}{\operatorname{argmin}} \sum_{i=1}^{n} d_C(s, s^i) \tag{8.2}$$

Evidentemente, su coste computacional usando el algoritmo trivial será $O(n^2 l^3)$, con l longitud máxima de las cadenas de S, ya que el cálculo de d_C es de orden cúbico con la longitud de las cadenas implicadas.

De forma inmediata se obtiene también la definición de cadena media para cadenas cíclicas. Así, dado $S = \{s^1, s^2, \ldots, s^n\}$ conjunto de cadenas cíclicas sobre Σ^*, la cadena media cíclica de S, notada por m_S^C, vendrá dada en este caso por [71]:

$$m_S^C = \underset{s \in \Sigma^*}{\operatorname{argmin}} \sum_{i=1}^{n} d_C(s, s^i) \tag{8.3}$$

También es posible hacer una definición basada en la definición alternativa propuesta en el Capítulo 4, es decir, empleando el sumatorio de distancias al cuadrado.

Dado que el cálculo de d_C implica el cálculo de una distancia de edición común entre dos cadenas, esta definición sólo viene a añadir un grado de complejidad más al proceso de búsqueda de la cadena media. Por tanto, el problema de encontrar la cadena media cíclica también será al menos NP-Duro.

Los algoritmos aproximados presentados en el Capítulo 3 tenían la característica de ser independientes de la distancia usada entre las cadenas implicadas. Por tanto, dichos algoritmos son aplicables inmediatamente para hallar aproximaciones a la cadena media cíclica, aunque evidentemente sufrirán el incremento de complejidad computacional asociado al mayor coste de cómputo de la distancia de edición cíclica.

La Sección 8.3 va dedicada a describir los experimentos realizados para comparar la calidad como prototipos entre la cadena mediana cíclica y las aproximaciones a la cadena media cíclica obtenidas por los algoritmos propuestos en esta tesis.

8.3. Experimentos comparativos

En esta sección describiremos un corpus de imágenes codificadas como cadenas de contorno que han sido transformadas a cadenas cíclicas, para posteriormente aplicar la extracción de prototipos (cadena mediana cíclica y cadena media cíclica aproximada) y realizar experimentos de clasificación que nos permitan verificar las ventajas de usar las aproximaciones a la media frente a la mediana.

8.3.1. El corpus *chicken*

El corpus *chicken* es un corpus de imágenes de despiece de pollo, que comprende un total de 446 imágenes binarizadas [3]. Cada imagen contiene una pieza de despiece, la cual puede clasificarse en cinco categorías distintas: ala (117 muestras), cuarto trasero (76 muestras), muslo (96 muestras), contramuslo (61 muestras) y pechuga (96 muestras).

En la Figura 8.2 se pueden observar distintos ejemplos de imágenes para cada una de las cinco clases. Estas imágenes se obtuvieron a partir de las imágenes de piezas reales en niveles de grises, colocadas en posición aleatoria y aplicando la caja de mínima inclusión sobre cada una de ellas, efectuando posteriormente un proceso de binarización. A estas imágenes binarias se les aplicó un remuestreo usando una rejilla de resolución 16×16 píxeles.

A partir de estas imágenes binarias, se crean las cadenas de contorno usando el clásico código de 4 direcciones [16], obteniendo así una representación en forma de cadenas no cíclicas y, por tanto, sensibles a rotaciones. Para obtener las cadenas cíclicas equivalentes, se aplicó el proceso destinado a obtener una cadena que indique los cambios relativos del ángulo en cada uno de los desplazamientos [20]. Para ello, si tenemos la cadena $s = s_1 \cdot s_2 \cdots s_l$, con $s_i \in \{0, 1, 2, 3\}$, definimos en primer lugar que $s_0 = s_l$. A partir de aquí, se obtiene la cadena $s' = s'_0 \cdot s'_1 \cdots s'_l$ como[1]:

$$s'_i = (s_i - s_{i-1} + 4) \mod 4, 1 \leq i \leq l$$

$$s'_0 = s'_m$$

Tras este proceso, se obtiene un conjunto de 446 cadenas cíclicas que representan cada una de las imágenes originales. Un resumen de las características del corpus obtenido se presenta en el Cuadro 8.1. En [55] se describe una experimentación exhaustiva con este corpus pero usando el muestreo de 64×64 píxeles. Los resultados alcanzados en dicho trabajo se sitúan entre el 22 % y el 33 % de

[1] Este conjunto de datos fue proporcionado con esta codificación descrita en [55]

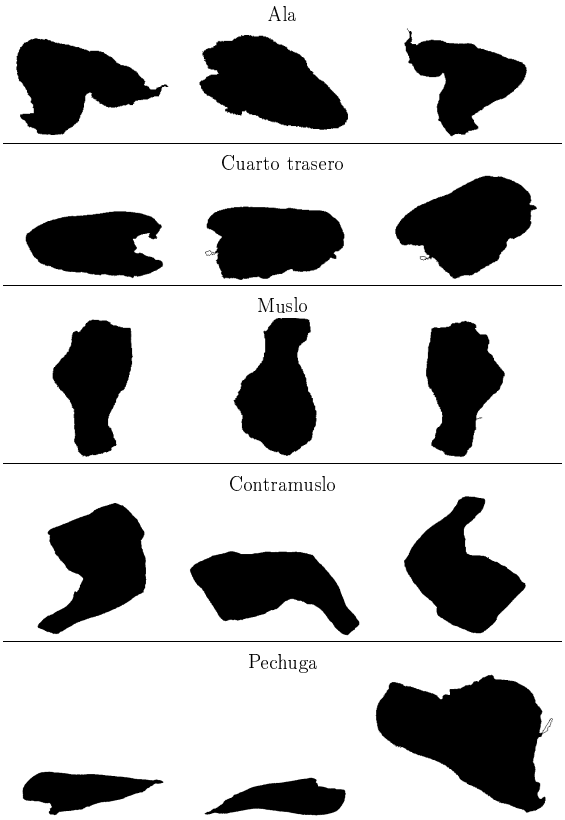


Figura 8.2: Ejemplos de imágenes de piezas de pollo para las diversas clases

Cuadro 8.1: Características del corpus *chicken*

Número de clases	5
Número de objetos	446
Tamaño del alfabeto	4
Longitud de las cadenas (mínima-máxima)	40-78

γ	0	1	2	3	λ
0	0	1	2	1	1
1	1	0	1	2	1
2	2	1	0	1	1
3	1	2	1	0	1
λ	1	1	1	1	-

Figura 8.3: Matriz de pesos usada para el corpus *chicken*

error. Estas cifras que no son en absoluto alcanzables con el muestreo 16×16, algo que tampoco es el objetivo de esta primera aproximación al problema de la cadena media cíclica.

A fin de obtener varios representantes por clase y poder realizar una clasificación mediante validación cruzada, se realizó un proceso de obtención de agrupamientos usando el algoritmo k-medianas inicializado por el algoritmo maxmin modificado, tal y como se presentó en la Sección 7.3. Dicho proceso se realizó para obtener 5 y 10 agrupamientos por clase, de manera que se pudieron obtener 5 (o 10) prototipos por cada una de las cinco clases.

En el proceso de clasificación por validación cruzada, siempre se tomará el i-ésimo agrupamiento de cada clase para constituir el conjunto de datos a clasificar, usando como prototipos los obtenidos del resto de agrupamientos.

8.3.2. Comparación entre la cadena mediana cíclica y la cadena media cíclica

Los experimentos de clasificación se han realizado para los prototipos obtenidos de cada agrupamiento, ya sea cadena mediana cíclica o cadena media aproximada (con las dos variantes del proceso de optimización) cíclica. La matriz de pesos usada es la que se muestra en la Figura 8.3. Esta matriz trata de castigar las sustituciones de un símbolo por el que representa el sentido opuesto (es decir, 0 contra 2 y 1 contra 3), dando un menor coste para direcciones no tan diferentes (las perpendiculares).

Los experimentos de clasificación se hicieron usando un clasificador k-NN, usando la distancia cíclica como medida de disimilitud. Se optó por una validación cruzada, tomando las cadenas extraídas de todos los conjuntos excepto el de test como prototipos. Esto permitió hacer una clasificación completa del conjunto de datos, haciendo las clasificaciones parciales de cada uno de los agrupamientos obtenidos.

En el caso de los experimentos para cinco agrupamientos, el valor del número de vecinos varió desde 1 hasta 5. Los resultados de clasificación, junto con los intervalos de confianza del 95 %, se muestran en la gráfica de la Figura 8.4. En dicha gráfica se puede apreciar que la cadena media aproximada cíclica presenta un mejor comportamiento que la cadena mediana cíclica. A medida que aumenta el número de vecinos usado en el clasificador estas diferencias van disminuyendo,

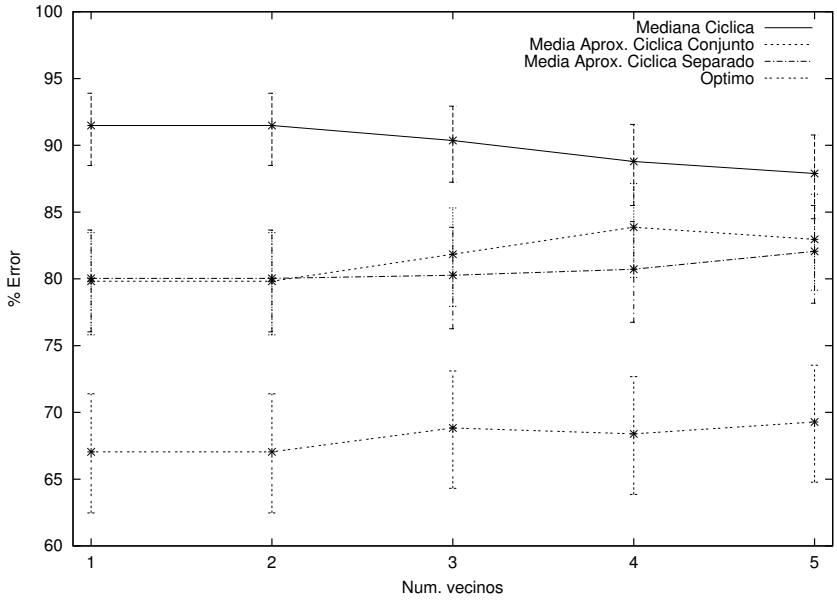


Figura 8.4: Resultados de clasificación usando la cadena mediana cíclica y la cadena media aproximada cíclica (con los métodos conjunto y separado) para cinco agrupamientos variando el número de vecinos del clasificador. Se incluye el óptimo (usar todas las muestras de entrenamiento como prototipos) para comparar

debido a que se presentan ciertas irregularidades en la clasificación con un mayor número de vecinos (como puede verse en el caso óptimo, es decir, cuando todas las muestras no incluídas en el test actúan como prototipos).

También se observa que el método separado ofrece, en términos absolutos, mejores resultados, aunque los intervalos de confianza nos indican que dichas diferencias con el método conjunto no son significativas. Se observa que el error absoluto en este caso también va aumentando con el incremento del número de vecinos. Estas irregularidades, comunes a todos los casos excepto a la cadena mediana cíclica, podrían achacarse a la escasez de datos, ya que 446 muestras en total, repartidas además de manera irregular entre las clases (con lo cual no existe igualdad entre las probabilidades de las mismas) es un número a todas luces escaso para otorgar una plena fiabilidad a los experimentos. Sin embargo, sí que es admisible que la cadena media cíclica tiene mejor comportamiento que la mediana cíclica (pues presenta diferencias significativas en clasificación).

Para confirmar los resultados, se recurre a la experimentación con diez agrupamientos, cuyos resultados de clasificación se muestran en la Figura 8.5. Como puede verse, la cadena media aproximada se comporta mejor, sea cual sea el método de optimización usado. Las diferencias con la cadena mediana cíclica

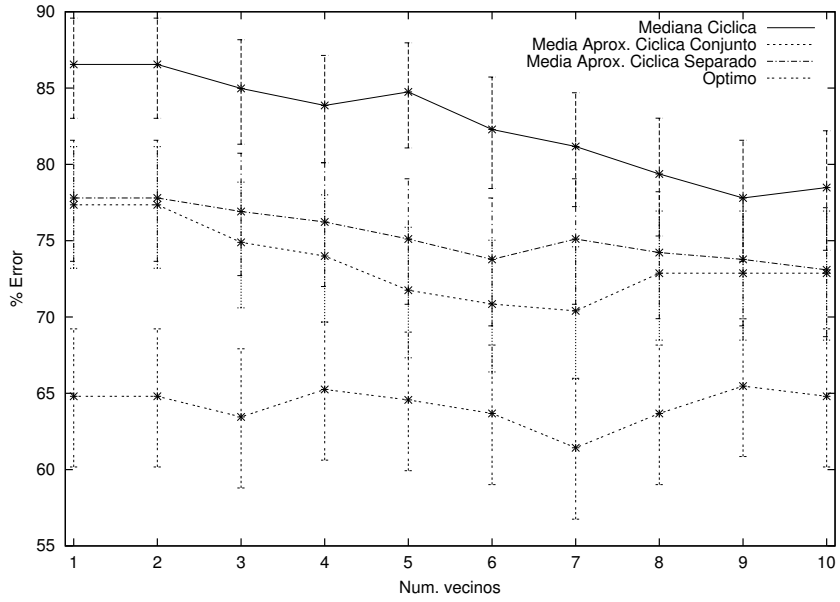


Figura 8.5: Resultados de clasificación usando la cadena mediana cíclica y la cadena media aproximada cíclica (con los métodos conjunto y separado) para diez agrupamientos variando el número de vecinos del clasificador. Se incluye el óptimo (usar todas las muestras de entrenamiento como prototipos) para comparar

van disminuyendo al incrementarse el número de vecinos.

Se observa también que las diferencias entre las medias aproximadas obtenidas usando el proceso conjunto o separado son ahora claramente favorables al método conjunto en términos absolutos, aunque el solapamientos de los intervalos de confianza nos muestra que no son diferencias significativas. También se muestra en dicha gráfica la evolución del caso óptimo, es decir, cuando todas las muestras que no se incluyen en el test se usan como prototipos. En este caso, el comportamiento es también irregular, sin una tendencia clara a aumentar o disminuir el error al aumentar el número de vecinos. De nuevo esto puede deberse a la escasez de datos, lo cual hace que se deba tener ciertas reservas ante los resultados obtenidos. De todas maneras, sí se concluye que las cadenas medias aproximadas para el caso cíclico tienen un mejor comportamiento que la mediana (siendo las diferencias claramente significativas para un clasificador que use pocos vecinos), aunque tampoco se libran de la irregularidad del comportamiento que tiene el caso óptimo (aunque es menos acusada que en el caso de cinco agrupamientos).

Así, se puede concluir que la aproximación a la cadena media cíclica otorga resultados mejores que la cadena mediana cíclica. Aunque las diferencias son

significativas, esta conclusión necesitaría de una adecuada confirmación usando un conjunto de datos más extenso que permitiera soslayar las dificultades e irregularidades que presenta el corpus *chicken* y que son claramente apreciables en el caso óptimo.

8.4. Resumen

En este capítulo hemos introducido el concepto de cadena cíclica como cadena que representa a un objeto (bidimensional) de manera invariante a las rotaciones que éste pueda sufrir. Asociado a cadena cíclica aparece el concepto de distancia de edición cíclica, que puede definirse en base a la distancia de edición clásica (no normalizada), de manera que se computa la distancia de edición entre una cadena s y la rotación más parecida a s de la otra cadena t. Definida esta distancia, de manera inmediata se sigue el concepto de cadena mediana y cadena media cíclicas. El caso de cadena media cíclica vuelve a aparecer como un problema NP-Duro que debe de resolverse de manera aproximada, usándose para ello los algoritmos del Capítulo 3. Dichas aproximaciones se aplican para una tarea de clasificación sobre un corpus de imágenes codificado como cadenas cíclicas. De los resultados de clasificación se concluye que también en el caso de cadenas cíclicas la cadena media aproximada resulta mejor prototipo que la cadena mediana.

Capítulo 9

Aproximación probabilística a la cadena media

Este capítulo establece una última asociación entre el problema de la cadena media y el marco probabilístico en el que se apoyan las técnicas paramétricas de Reconocimiento de Formas. De esta manera, se establece la relación entre técnicas paramétricas y no paramétricas para este problema y se estudia experimentalmente la eficacia de un tipo de modelos paramétricos (los modelos ocultos de Markov) para resolver el problema de clasificación en el corpus de cromosomas *Copenhagen*.

9.1. Aproximación por máxima verosimilitud a la cadena media

En el Capítulo 1, comentábamos que se distinguía, en función de los modelos usados, entre los métodos de clasificación paramétricos y no paramétricos [15]. En los métodos de clasificación paramétricos, las clases vienen representadas por un modelo, el cual viene dado por un conjunto de parámetros representativos. En cambio, en los métodos no paramétricos son los propios objetos los que constituyen el modelo, sin obtener explícitamente una descripción paramétrica de la clase.

La forma más usual de representar un modelo en el caso paramétrico es una distribución de probabilidad. Estas distribuciones de probabilidad vienen definidas por unos ciertos parámetros fundamentales (por ejemplo, en el caso de una distribución normal, serían la media y la varianza de la distribución), y cada clase viene representada por una distribución distinta.

A la hora de realizar la clasificación de un cierto objeto, se recurre generalmente al criterio de máxima verosimilitud [15]: el objeto se clasifica en aquella clase cuya probabilidad de que el objeto esté generado por dicha clase sea máxima. Así, dado un conjunto de clases C, cada una de las clases está repre-

sentada por una distribución de probabilidad (modelo paramétrico). Por tanto, el conjunto de clases está representado por un conjunto de distribuciones de probabilidad Θ. Así, dado un objeto x, éste se clasificaría en la clase $\hat{\theta}$ tal que:

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}} \Pr_\theta(x) \tag{9.1}$$

El criterio de máxima verosimilitud se aplica también en el aprendizaje de los modelos. Así, si tuviéramos un conjunto de objetos $X = \{x_1, x_2, \ldots, x_n\}$ de una cierta clase de la cual queremos estimar su modelo $\hat{\theta}$, dicho modelo sería:

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}} \prod_{i=1}^{n} \Pr_\theta(x_i) = \underset{\theta \in \Theta}{\operatorname{argmax}} \sum_{i=1}^{n} \log(\Pr_\theta(x_i)) = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{n} -\log(\Pr_\theta(x_i)) \tag{9.2}$$

Mediante esta formulación del modelo paramétrico es posible formular el problema de la cadena media, ya que existe una equivalencia con la definición de la misma. Así, si asumimos que existe una definición de distancia d entre un modelo θ y un objeto x, es posible definir la probabilidad de que un cierto objeto x sea generado por la distribución θ como:

$$\Pr_\theta(x) = \exp(-d(x, \theta)) \tag{9.3}$$

La definición de la Ecuación (9.3) es razonable ya que intutivamente es claro que cuanto más cercano esté el objeto a la distribución, mayor probabilidad tendrá de ser generado por esta, alcanzando su mayor valor (1) cuando la distancia es mínima (0, por definición). Igualmente, dicha probabilidad va disminuyendo a medida que nos alejamos de la distribución (debido al signo negativo de la exponencial).

Si aplicamos ahora el criterio de máxima verosimilitud para la estimación de $\hat{\theta}$, tal y como hemos definido en la Ecuación (9.2), para esta distribución de probabilidad, tendremos que dicha estimación sigue la regla de asignar el modelo dado por $\hat{\theta}$ tal que:

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{n} -\log[\exp(-d(x_i, \theta))] = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{n} d(x_i, \theta) \tag{9.4}$$

Así, si asumimos que el conjunto de datos X es un conjunto de cadenas $S = \{s^1, s^2, \ldots, s^n\} \subset \Sigma^*$ y que el conjunto de modelos es Σ^*, tenemos que:

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{n} d(x_i, \theta) = \underset{s \in \Sigma^*}{\operatorname{argmin}} \sum_{i=1}^{n} d(s^i, s) = m_S \tag{9.5}$$

Por tanto, se concluye que el problema de la cadena media es abordable mediante el uso de la aproximación por máxima verosimilitud, usando como

distribución de probabilidad la exponencial negativa de la distancia entre la cadena y el modelo de la clase. Dicho modelo es una cadena de Σ^* (la cadena media de S).

Así pues, el siguiente paso es recurrir a una formulación mediante modelos paramétricos del problema de clasificación para cadenas. Dichos modelos podrían también emplearse para obtener la cadena de máxima probabilidad (que sería una aproximación a la cadena media). Como estamos tratando con cadenas, que son secuencias de símbolos sobre un alfabeto, el modelo paramétrico que mejor se ajusta a esta situación son los modelos ocultos de Markov.

9.2. Modelos ocultos de Markov

Un modelo oculto de Markov [28] (generalmente conocido por sus siglas HMM, del inglés *Hidden Markov Models*) es un modelo estocástico constituído a partir de un cierto conjunto de estados $Q = \{q_1, q_2, \ldots, q_n\}$ y un cierto conjunto de símbolos Σ, y definido mediante el conjunto de distribuciones de probabilidad (A, B, π), con:

$$A : Q \times Q \to [0,1] \quad \text{t.q.} \quad \sum_{\forall q' \in Q} A(q, q') = 1 \qquad \forall q \in Q$$

$$B : Q \times \Sigma \to [0,1] \quad \text{t.q.} \quad \sum_{\forall a \in \Sigma} B(q, a) = 1 \qquad \forall q \in Q$$

$$\pi : Q \to [0,1] \quad \text{t.q.} \quad \sum_{\forall q \in Q} \pi(q) = 1$$

A estas tres distribuciones se les conoce respectivamente como probabilidad de transición entre estados, probabilidad de emisión de símbolos y probabilidad de estado inicial.

Así, un HMM puede utilizarse para modelar conjuntos de secuencias de símbolos, lo cual lo hace ideal para actuar como modelo de un conjunto de cadenas. Para ello, es necesario reestimar los parámetros del HMM (principalmente las distribuciones de probabilidad A y B, pues se puede asumir que el estado inicial sea único) a partir del conjunto de cadenas que tiene que modelar.

El algoritmo usual para este proceso de reestimación es el algoritmo de Baum-Welch [28], que utiliza un método de descenso por gradiente para estimar los parámetros de un HMM basándose en las secuencias de estados que se recorren dada una cadena. Existe sin embargo la posibilidad de utilizar el algoritmo de Viterbi para este proceso de reestimación [28]. En dicho proceso se utiliza únicamente la secuencia de estados de máxima probabilidad en dicho proceso de reestimación, actualizando únicamente las probabilidades implicadas en dicha secuencia.

Un punto que es necesario resolver antes del proceso de reestimación es la inicialización del HMM, es decir, dar un valor inicial a las probabilidades del modelo. Entre otras técnicas, destaca la técnica de la segmentación lineal, que

será la usada en los experimentos. Esta técnica parte de un HMM con una topología lineal definida (es decir, se tienen definidos los estados y las transiciones izquierda a derecha), y divide las cadenas a modelar en tantos segmentos como número de estados tenga el modelo. Así, quedan definidas las frecuencias de emisión por cada estado y las frecuencias de transición entre estados, y a partir de dichas frecuencias se obtienen las probabilidades iniciales correspondientes.

Así pues, con un HMM h podemos modelar un conjunto de cadenas S, y para cada posible conjunto de cadenas tendríamos un HMM distinto. A la hora de realizar la clasificación de una cadena desconocida s en el conjunto de clases representado por el conjunto de HMM H, la aproximación por máxima verosimilitud nos indicaría escoger la clase representada por el modelo $\hat{h}$ tal que:

$$\hat{h} = \underset{h \in H}{\operatorname{argmax}} \Pr_h(s) \tag{9.6}$$

Existe también la posibilidad de que una clase venga representada por más de un HMM (por ejemplo, que la clase esté dividida en diversos subconjuntos de datos y que con cada uno de ellos se estime un HMM). En este caso, la alternativa tomada es, tras estimar un HMM por cada agrupamiento, obtener después la probabilidad *a posteriori* de cada uno de los modelos estimados. La clasificación se hace en la clase tal que la suma de probabilidades *a posteriori* de los HMM pertenecientes a la misma (un HMM por cada agrupamiento de la clase) sea máxima respecto al resto de clases.

Es decir, si tenemos que las cadenas de C_j se han subdividido (utilizando cualquier técnica de obtención de agrupamientos) en los subconjuntos de cadenas $C_j^1, C_j^2, \ldots, C_j^{n_j}$, y para cada una de ellas se ha estimado el correspondiente modelo de Markov h_j^i, la clasificación de la cadena s se hace en la clase $\hat{C}$ siguiendo la regla:

$$\hat{C} = \underset{C_j}{\operatorname{argmax}} \sum_{i=1}^{n_j} \Pr(h_j^i | s) \tag{9.7}$$

En la Sección 9.3 realizaremos experimentos de clasificación usando HMM como modelos y comparando los resultados obtenidos con los que otorgan los clasificadores NN usando las aproximaciones a la cadena media descritas a lo largo de esta tesis.

9.3. Experimentos comparativos

En esta sección vamos a comparar los resultados de clasificación entre clasificadores NN usando la cadena mediana y una aproximación a la cadena media, y la clasificación mediante HMM. Para la experimentación usaremos el corpus *Copenhagen* descrito en el apartado 2.2.1, dividido en diversos agrupamientos usando la técnica de agrupamiento k-medias generalizado por cadena mediana, con inicialización por maxmin modificado con k-medianas (tal y como se describe en el Capítulo 7).

Cuadro 9.1: Error de clasificación usando un HMM por clase para diversos factores de longitud f (número de estados del HMM igual a la longitud media de las cadenas de entrenamiento por f)

Factor	0'5	0'8	1'0	1'1	1'2
% Error	24'41 %	9'55 %	6'57 %	6'75 %	5'57 %
Factor	1'3	1'4	1'5	1'7	2'0
% Error	5'55 %	5'36 %	6'34 %	7'86 %	7'57 %

Cuadro 9.2: Número de parámetros de cada uno de los modelos usado en la clasificación. Para los HMM es el número total de componentes elementales de las probabilidades iniciales, de transición y de emisión. Para las cadenas, el total de símbolos

Número de agrupamientos	HMM	Cadena mediana	Cadena media aprox. inic. mediana proc. conjunto
1	266503	2157	2156
5	1342718	10772	10575
10	2682255	21702	21277

Para la construcción de los HMM es necesario en primer lugar seleccionar la topología adecuada para la clase. En nuestro caso, la inicialización por segmentación lineal determina que sea una topología clásica de izquierda a derecha. Las transiciones entre estados (incluyendo saltos y bucles) vienen determinadas por la propia inicialización. El único factor que queda por determinar es el número de estados del HMM.

En nuestro caso, la implementación de la creación de los HMM nos permite especificar el número de estados como una cantidad proporcional a la longitud media de las cadenas de la clase. Es decir, si la longitud media de las cadenas a modelar es l y el factor usado es f, el HMM tendrá $l \cdot f$ estados.

Se hizo una batería de pruebas sobre una serie de valores para este factor, inicializando un HMM por cada una de las 22 clases disponibles y luego reestimando sus parámetros hasta la convergencia. En el Cuadro 9.1 se muestran los resultados a nivel de error de clasificación (siguiendo el protocolo por validación cruzada que se ha usado en el resto de este trabajo) para cada valor probado del factor, lo que nos llevó a elegir como factor de longitud $1'4$.

La estimación de los modelos de Markov se hizo para 1, 5 y 10 agrupamientos, realizándose posteriormente la clasificación. El número de parámetros por modelo usado se indica en el Cuadro 9.2, donde los parámetros de los HMM son el número de probabilidades a estimar (de estado inicial, de transición entre estados y de emisión de símbolos), y los parámetros de las cadenas es el número

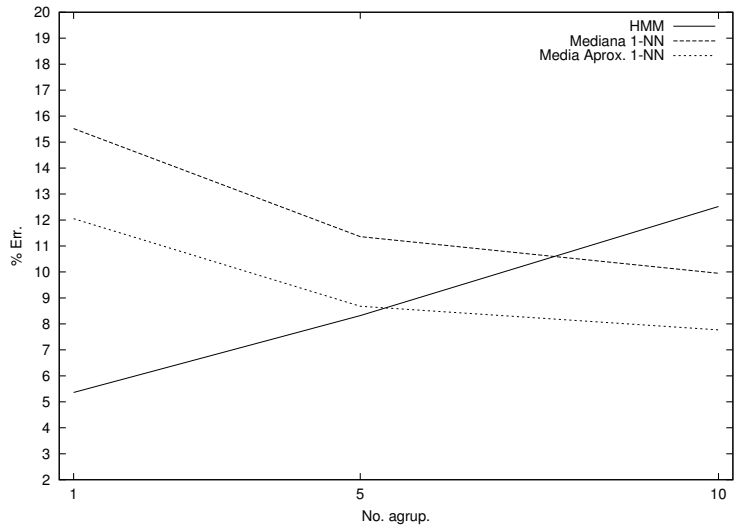


Figura 9.1: Resultados de clasificación usando modelos de Markov (HMM) y clasificador por vecino más cercano usando como prototipos la cadena mediana y la media aproximada por inicialización mediana y proceso conjunto

de símbolos total del conjunto de prototipos. Puede verse que el crecimiento del número de parámetros resulta en todos los casos proporcional al número de agrupamientos, pero el número de parámetros de los HMM es mucho más elevado que el de las cadenas. Sin embargo, como la naturaleza de ambos parámetros es muy diferente, no se puede establecer una comparación clara.

Los resultados a nivel de clasificación usando HMM y usando clasificadores NN utilizando cadena mediana y cadena media aproximada con inicialización por mediana y proceso conjunto se presentan en la gráfica de la Figura 9.1.

De los resultados obtenidos se puede deducir que los HMM funcionan mejor para un número pequeño de agrupamientos. Esto es razonable, pues un número pequeño de agrupamientos implica gran número de muestras por agrupamiento, y a mayor número de muestras la estimación de los parámetros de los HMM es mucho más precisa. Así, a medida que aumenta el número de agrupamientos, a pesar de disponer de varios HMM con los cuales computar la probabilidad *a posteriori* para saber la probabilidad de que una cadena pertenezca a la clase, estos modelos no se estiman adecuadamente (existen demasiados parámetros y pocos datos para estimarlos adecuadamente) y resulta mucho mejor a nivel de clasificación usar un clasificador NN con la cadena mediana o la media aproximada de cada agrupamiento como conjunto de prototipos.

En términos absolutos, el mejor resultado hallado para HMM da una tasa de error de un 5'36 % (caso de un solo agrupamiento), mientras que para NN tenemos un mejor resultado de 5'91 % (usando todas las cadenas como prototipos, es decir, el equivalente a 100 agrupamientos). Usando un clasificador k-NN,

para $k = 9$ llegamos a mejorar los HMM obteniendo una tasa de error del 3'89 % en las mismas condiciones que el NN. Hay que destacar que en dichas circunstancias el número total de símbolos de las cadenas alcanza a ser 212170, con lo que se tiene un número de parámetros semejante al de los HMM con un solo agrupamiento.

Hay que tener en cuenta de todas maneras que este estudio se ha hecho para un caso muy específico, con HMM con un número de estados proporcional de manera fija a la longitud media de las cadenas que modela, con inicialización por segmentación lineal y reestimados mediante el algoritmo de Viterbi. Sería necesario estudiar otras variantes en la obtención de los HMM para validar de manera completa los resultados obtenidos para este caso particular.

También quedaría por estudiar una aplicación posterior a la obtención de los HMM, que sería obtener la cadena de máxima probabilidad generable por el HMM. Este es un problema NP-Duro, aunque es abordable mediante una aproximación que use el algoritmo de Viterbi, aproximación que es de coste polinómico. El posterior uso de dicha cadena como prototipo en clasificadores k-NN, nos permitiría comparar su efectividad como prototipo frente a la cadena mediana y las cadenas medias aproximadas obtenidas mediante los métodos basados en perturbación descritos en este trabajo.

9.4. Resumen

En este capítulo hemos abordado la interpretación probabilística del problema de la cadena media, demostrando que existe una equivalencia entre un modelo paramétrico por máxima verosimilitud y la definición de cadena media. Ello nos ha llevado a considerar la representación por modelos paramétricos, más específicamente por modelos ocultos de Markov, de conjuntos de cadenas. La comparación a nivel de clasificación con el corpus *Copenhagen* muestra que los HMM son excelentes modelos para un número pequeño de agrupamientos (gran número de cadenas por agrupamiento), pero que su calidad se degrada a medida que aumenta este número de agrupamientos, siendo superados en clasificación por un clasificador por vecino más cercano usando cadena mediana o aproximaciones a la cadena media.

Capítulo 10

Conclusiones y trabajos futuros

En este capítulo hacemos un resumen final de las conclusiones que se han ido obteniendo a lo largo del trabajo desarrollado en esta tesis. Finalmente, se proponen futuras líneas de investigación y trabajos a desarrollar dentro de la aplicabilidad del concepto de cadena media para el Reconocimiento de Formas, así como sobre el propio concepto y sus métodos de obtención.

10.1. Conclusiones

A lo largo de la tesis hemos trabajado primordialmente con algoritmos de obtención de aproximaciones a la cadena media, ya que desde el principio tenemos la limitación de que dicha obtención es un problema NP-Duro [11]. La aproximación clásica es la cadena mediana, obtenible en coste polinómico pero de dudosa precisión en el caso general. Una aproximación alternativa es obtener la cadena media aproximada mediante un proceso constructivo voraz [9]. Sin embargo, dicha aproximación también resulta demasiado burda y está además basada en el proceso de obtención de la distancia de edición ponderada; esto la hace difícilmente extrapolable a otras distancias entre cadenas, como es el caso de la distancia de edición normalizada [53], que es la distancia que se ha usado a lo largo de este trabajo.

Basándonos así en la propuesta de Kohonen de realizar un proceso de perturbación [38], se han propuesto dos algoritmos que, a partir de una cadena inicial dada, realizan un proceso de perturbación buscando decrementar la distancia acumulada al conjunto de cadenas (ya que la cadena media es aquella de menor distancia acumulada). Los resultados en clasificación de dichas aproximaciones resultan claramente superiores a los obtenidos usando la cadena mediana [46, 48], con lo que se concluye que nuestros algoritmos proporcionan una mejor aproximación para representar un conjunto de datos. Todo ello, por

desgracia, bajo la necesidad de incrementar espectacularmente el coste computacional de la obtención de dichos representantes.

Igualmente, los resultados obtenidos por aplicar uno u otro algoritmo han demostrado que la diferencia de calidad de los representantes obtenidos no es significativa, con lo cual ambos procesos son igualmente efectivos para la obtención de aproximaciones de calidad a la cadena media.

Los algoritmos propuestos se basan en realizar perturbaciones a partir de una cadena inicial, con lo cual son sensibles a la inicialización. Este punto era necesario estudiarlo, usando para ello inicializaciones razonables, como pudieran ser la cadena mediana o la cadena obtenida por el proceso constructivo voraz [9]. Los experimentos realizados para verificar posibles diferencias debidas a la inicialización han descartado que dichas diferencias sean significativas, lo cual nos permite concluir que nuestros algoritmos son lo suficientemente robustos con respecto a la inicialización.

Por otro lado, se ha propuesto también una definición alternativa de cadena media, basada en suma de distancias al cuadrado [47], que evita ciertas incoherencias conceptuales cuando se define la cadena media para conjuntos de dos cadenas. El uso de dicha definición alternativa como criterio de optimalidad no arroja, en la práctica, diferencias significativas en las aproximaciones obtenidas por el algoritmo, las cuales presentan un comportamiento semejante en clasificación con respecto a las obtenidas usando la definición clásica.

Como hemos comentado previamente, el precio a pagar por obtener estas aproximaciones es el incremento de la complejidad computacional. Por tanto, el estudio de técnicas que permitieran la reducción del coste era necesario. Dos técnicas han sido propuestas para permitir esa reducción del coste [50], incidiendo en dos factores distintos al respecto. Por un lado, la técnica de la división, la cual incide sobre el coste asociado a la longitud de las cadenas. Por otro lado, la técnica de la optimización local, que incide en el tamaño del alfabeto sobre el que se definen las cadenas. Ambas técnicas han sido experimentadas, proporcionando una reducción de la complejidad temporal notable, pero también degradando la calidad de las aproximaciones obtenidas. Esta degradación varía según la técnica aplicada, y es, como cabía esperar, directamente proporcional a la reducción de coste obtenida. Los resultados obtenidos muestran que esta degradación es mínima para optimización local. En general, todas resultan mejores aproximaciones que la cadena mediana y además presentan un coste temporal más reducido que las aproximaciones no optimizadas.

Todas estas técnicas se han experimentado intensivamente sobre un corpus de cromosomas bastante usado en tareas de Reconocimiento de Formas [44], usando diferentes tamaños de conjuntos de cadenas y usando una división estándar en dos partes iguales para hacer validación cruzada [1].

Por otra parte, no era posible renunciar a buscar una solución exacta al problema de la cadena media. Ya que dicho problema es NP-Duro, se planteó su resolución mediante una estrategia de Ramificación y Poda, lo cual llevó a la

[1] Se realizaron también experimentos preliminares con un corpus de cromosomas de mayor tamaño, presentados en el Apéndice B

necesidad de definir funciones de cota para una cierta cadena (indicando la mínima distancia acumulada que podría obtener cualquier cadena de la que la cadena actual fuera prefijo). La cota proporcionada para la distancia de edición normalizada (basada en una cota para la no normalizada) presenta el problema de depender de la longitud máxima de la cadena media. Se ha propuesto una cota para dicha longitud máxima, pero su veracidad ha quedado como simple conjetura, aunque razones intuitivas nos llevan a pensar en su admisibilidad. La falta de esta demostración se suple con la demostración de cotas inferiores para la distancia acumulada de la cadena media, a fin de verificar la desviación de distancia acumulada del resultado obtenido frente al óptimo teórico, y con la comprobación empírica para ciertos conjuntos de datos de que la conjetura se cumple sobradamente.

Los experimentos realizados para la obtención de la cadena media exacta se realizaron sobre un corpus artificial controlado, con cadenas sobre un alfabeto reducido y longitud limitada. Aún así, la complejidad natural del problema llevó a que de ciertos experimentos sólo se obtuvieran resultados parciales, que de todos modos obtenían mejoras respecto a nuestras aproximaciones a nivel de distancia acumulada al conjunto de cadenas. Los resultados obtenidos a nivel de clasificación nos permiten concluir que las aproximaciones propuestas son razonablemente buenas con respecto a la media exacta, a pesar de haberse obtenido con un corpus artificial (aunque se hicieron pequeñas pruebas que lo validaban para una codificación reducida en algunas clases del corpus *Copenhagen*).

El siguiente paso fue la aplicación de las aproximaciones propuestas en técnicas de obtención de agrupamientos. Así, se propone un nuevo algoritmo de obtención de agrupamientos, denominado k-medias generalizado [49], directamente inspirado en los algoritmos k-medias [15] y k-medianas [33]. Además, como dicho proceso requiere de un método de inicialización, se ha propuesto un nuevo método de inicialización basado en el algoritmo maxmin [34], pero incorporando en el mismo la optimización de los agrupamientos con los métodos propuestos. Las diversas combinaciones entre inicialización y posterior reagrupamiento han sido experimentadas sobre el corpus *Copenhagen* y comparadas entre sí y con los agrupamientos estándar usados para dicho corpus en el resto de la tesis, demostrando que algunas de las técnicas (en concreto, las que inicializan con la versión modificada del maxmin con k-medianas y reagrupan con k-medias generalizado) proporcionan una mejora de resultados para un cierto rango del número de agrupamientos.

Tras ello, se ha extendido el concepto de cadena mediana y cadena media al caso de cadenas cíclicas, es decir, cadenas que representan un objeto bidimensional de manera invariante a las rotaciones. De igual manera, se han podido extender los algoritmos de obtención de aproximaciones a este caso. Dichas aproximaciones, así como la cadena mediana, han sido computadas para extraer prototipos de un corpus de despiece de piezas de pollo [3], de tamaño reducido, codificado en cadenas de contorno cíclicas en código de 4 direcciones [16]. Los resultados obtenidos muestran que las aproximaciones a la cadena media tienen mejor comportamiento que la cadena mediana en las tareas de clasificación, aunque el reducido tamaño del corpus hace que los resultados presenten

ciertas irregularidades que provoca que se deba tener ciertas reservas sobre la representatividad de los mismos.

Finalmente, se ha abordado el problema de la cadena media desde una perspectiva probabilística, reduciéndose el problema de hallar la cadena media a una estimación por máxima verosimilitud de una distribución de probabilidad. Dada dicha equivalencia, se han podido estimar modelos ocultos de Markov (HMM) para modelar conjuntos de cadenas y probar posteriormente su calidad en la clasificación frente a clasificadores NN que usan cadenas como prototipos (en particular, las cadenas medias aproximadas obtenidas con los métodos propuestos). Dicha experimentación ha revelado un excelente comportamiento de los HMM cuando el número de agrupamientos es reducido (y por tanto hay un número suficiente de cadenas para estimar sus parámetros), pero que se degrada rápidamente cuando existen varios agrupamientos. El clasificador NN muestra un comportamiento inverso.

En resumen, las aportaciones realizadas se pueden concretar en los siguientes puntos:

- Propuesta de dos algoritmos aproximados para la obtención de la cadena media.
- Propuesta de una nueva definición de cadena media que resuelve los problemas que presenta la definición clásica para ciertos casos concretos.
- Propuesta de optimizaciones de los algoritmos aproximados a fin de reducir el coste de cómputo de las aproximaciones sin perder por ello calidad en las mismas.
- Experimentación intensiva con los algoritmos y definiciones propuestas con un corpus de cromosomas.
- Experimentos complementarios de índole preliminar con un corpus de cromosomas de gran tamaño.
- Propuesta de un algoritmo basado en Ramificación y Poda (con sus consiguientes cotas) para la obtención de la cadena media exacta y aplicación del mismo a un corpus sintético.
- Propuesta de un algoritmo de obtención de agrupamientos para cadenas (k-medias generalizado) basado en las aproximaciones propuestas a la cadena media y aplicación del mismo a un corpus de cromosomas.
- Propuesta de un algoritmo de inicialización para los algoritmos de obtención de agrupamientos basado en el algoritmo maxmin y los métodos k-medianas y k-medias generalizado y aplicación del mismo a un corpus de cromosomas.
- Aplicación de los algoritmos de cómputo de cadena mediana y cadena media aproximada para el caso de cadenas cíclicas y aplicación de los mismos a un corpus de imágenes de piezas de pollo.

- Modelado con HMM de cadenas y comparación de los mismos como modelos en tareas clasificación respecto a las cadenas medias aproximadas.

Las publicaciones de carácter científico que se han desarrollado a partir del trabajo recogido en esta tesis son las siguientes (por orden cronológico):

1. C. D. Martínez Hinarejos, A. Juan, y F. Casacuberta. Use of median string for classification. En *Proceedings of the 15th Int. Conf. on Pattern Recognition (ICPR 2000)*, volumen 2, páginas 907–910, Barcelona, Septiembre 2000.

2. C. D. Martínez Hinarejos, A. Juan, y F. Casacuberta. Improving classification using median string and NN rules. En *Proc. of the IX Spanish Symposium on Pattern Recognition and Image Analysis*, volumen II, páginas 391–395, Benicàssim, Mayo 2001.

3. C. D. Martínez Hinarejos, A. Juan, F. Casacuberta, y R. Mollineda. Reducing the computational cost of computing approximated median strings. En Terry Caelli, Adnan Admin, Robert P. W. Duin, Mohamed Kamel, y Dick de Ridder, editores, *Structural, Syntactic and Statistical Pattern Recognition, Joint International Workshops SSPR 2002 and SPR 2002 Proceedings*, Lecture Notes in Artificial Intelligence LNCS/LNAI 2396, páginas 47–55, Windsor, Ontario, Canadá, Agosto 2002. Springer-Verlag.

4. C. D. Martínez Hinarejos, A. Juan y F. Casacuberta. Prototype Extraction for k-NN Classifiers using Median Srings. En D. Chen y X. Cheng editores, *Pattern Recognition and String Matching*, páginas 465–476, Kluwer Academic Publishers, Diciembre 2002.

5. C. D. Martínez Hinarejos, A. Juan, y F. Casacuberta. Median strings for k-nearest neighbour classification. *Pattern Recognition Letters*, 24(1-3):173–181, 2003.

6. C. D. Martínez Hinarejos, A. Juan, y F. Casacuberta. Generalized k-Medians Clustering for Strings. En Francisco José Perales, Aurélio J. C. Campilho, Nicolás Pérez de la Blanca y Alberto Sanfeliu, editores, *Pattern Recognition and Image Analysis, First Iberian Conference IbPRIA 2003 Proceedings*, Lecture Notes in Computer Science LNCS 2652, páginas 502-509, Port d'Andratx, Mallorca, España, Junio 2003. Springer-Verlag.

10.2. Trabajos futuros

A partir de los trabajos desarrollados en esta tesis existen una gran cantidad de temas interesantes sobre los cuales seguir trabajando, tanto respecto al uso de la cadena media en tareas de Reconocimiento de Formas como al concepto de cadena media en sí mismo.

Por un lado, y desde el punto de vista teórico, resultaría interesante poder demostrar la conjetura sobre la longitud de la cadena media, la cual resulta fundamental para asegurar que las cotas usadas son completamente correctas. En el mismo camino, la obtención de nuevas cotas más ajustadas y que, por tanto, llevarían a una búsqueda más eficiente, es un objetivo a asumir en desarrollos posteriores que impliquen este concepto.

En esta tesis nos hemos limitado a la obtención de la cadena media usando la distancia de edición normalizada la cual, por otra parte, parece la más adecuada para las tareas de clasificación efectuadas. Sin embargo, existen otras tareas de clasificación en las cuales la medida de distancia más adecuada entre cadenas puede ser cualquier otra. Si bien los algoritmos de obtención de las aproximaciones son aplicables a cualquier otra distancia definida, las cotas necesarias para la obtención de la cadena media exacta varían con toda seguridad respecto a las definidas para la distancia de edición normalizada. Por tanto, desde el punto de vista teórico también resulta interesante el uso de otras medidas de distancia para la definición de la cadena media y estudiar cómo afectan dichas medidas a la obtención de las aproximaciones y de la media exacta.

Desde el punto de vista práctico, se abre un campo muy extenso para la experimentación con la cadena media y sus aproximaciones. Su aplicación a diversas tareas de clasificación, usando corpora distintos de los utilizados en esta tesis, resulta algo inmediato. Ahora bien, esto no quiere decir que el uso de la cadena media sea la técnica más adecuada en cualquier corpora, ya que en muchas tareas (como OCR, por ejemplo), la calidad de las aproximaciones basadas en métodos geométricos es tan alta que ya cuestiona de por sí la codificación de los objetos como cadenas. Sin embargo, para datos que presenten una codificación natural en cadenas de símbolos (como secuencias biológicas), es de inmediata aplicación el concepto de cadena media para obtener prototipos.

Evidentemente, los algoritmos propuestos en esta tesis se han basado únicamente en el enfoque de perturbaciones sucesivas y minimización de la distancia acumulada tal y como se propuso por Kohonen [38]. Sin embargo, existen otras posibilidades de aproximar la cadena media, como pudiera ser ir obteniendo aproximaciones de manera incremental, es decir, obtener la media aproximada de dos, y con la resultante obtener la media aproximada con otra más, y así sucesivamente [30]. Esta aproximación se basa en el concepto de media ponderada [7], por el cual se puede controlar cuánto queremos que pese cada una de las cadenas consideradas en el cómputo de la media. Sin embargo, dichos trabajos sólo se han desarrollado para conjuntos de datos muy específicos y haciendo uso de la distancia de edición clásica. Por tanto, la ampliación de dichos conceptos para la distancia de edición normalizada (u otras medidas de distancia entre cadenas) y su aplicación a otros corpora (como los usados en esta tesis) es una de las tareas a abordar.

Además de esta posible aproximación, es claro que pueden existir otras muchas aproximaciones basándose tanto en el concepto de perturbación de una cadena (por ejemplo, combinar múltiples perturbaciones sería otra alternativa) como en otros métodos (corrección de errores, algoritmos genéticos, etc.).

De la misma manera, aunque se han explorado diversas combinaciones en

la obtención de agrupamientos, existen otras posibles combinaciones que no se han experimentado y que sería interesante comprobar. En particular, resulta interesante la tarea de saber hasta qué punto es dependiente de la inicialización el algoritmo k-medias generalizado. Ya hay estudios desarrollados para k-medianas [34], pero no existen ese tipo de estudios para este nuevo algoritmo y sería interesante verificar su robustez ante la inicialización. Igualmente, la aplicación de k-medias generalizado en el ámbito del aprendizaje no supervisado es una aplicación interesante por estudiar.

Por otro lado, la experimentación con cadenas cíclicas, debido a las dificultades que entraña (su alto coste computacional ante todo) ha sido muy reducida. Varios puntos de investigación se abren por esta área: aplicación de un algoritmo eficiente para el cómputo de distancia de edición normalizada para cadenas cíclicas [52], aplicación de métodos de aproximación para ese cálculo, propuesta de nuevos métodos para hallar la cadena media cíclica,... Si bien se han realizado otros trabajos respecto a la obtención de la cadena media de cadenas cíclicas [71], estos no se basan en un trabajo directo con las cadenas, sino en recodificaciones de segmentos de la cadena como funciones matemáticas, calculando posteriormente la media de dichas funciones y retornando a la codificación original por cadenas. Por tanto, dicha aproximación conlleva una nueva dificultad: el ajuste de una función al contorno representado. Así pues, la necesidad de explorar métodos que no requieran de este tipo de esfuerzos adicionales (semejantes a los propuestos en esta tesis) resulta evidente.

El haber usado cadenas es realmente un caso particular de usar secuencias de datos. Usando cadenas, hemos hecho que la secuencia de datos se forme a partir de un número reducido de unidades básicas. Sin embargo, la idea de una extensión de dichas unidades básicas a conjuntos no discretos, como números reales o puntos de un espacio real de n dimensiones, resulta inmediata, quedando así secuencias de (vectores de) números reales. En estos casos, las ideas que fundamentan nuestros algoritmos (proceso perturbativo que mejore la aproximación actual) son aplicables, pero surgen una serie de problemas de planteamiento dados por la infinitud del "conjunto de símbolos" en el cual se toman las posibilidades de perturbación. Y si bien la diferencia entre componentes elementales se puede determinar de una manera más clara (la distancia euclídea parece una medida adecuada), su coste computacional también resulta otro escollo en la resolución de este problema. Por tanto, un cuidadoso estudio sobre las posibilidades de aplicar nuestras técnicas para secuencias (vectoriales) reales puede ser una fuente de nuevas investigaciones.

Por último, una línea de investigación con eminentes consecuencias prácticas es la búsqueda de la reducción del coste temporal de los algoritmos de aproximación, más allá de las técnicas propuestas en esta tesis. Si bien existe una posibilidad inmediata a probar que es la combinación de las técnicas de la división y de la optimización local, el desarrollo de otras alternativas que nos proporcionen buenas aproximaciones en un tiempo inferior continúa siendo una área abierta, donde una gran diversidad de aportaciones pueden ser realizadas. También la propia mejora de las técnicas propuestas puede ser un campo interesante; por ejemplo, determinar adecuadamente los símbolos a usar en la

optimización local usando otros parámetros ajenos a la matriz de pesos, o hacer las divisiones atendiendo a características locales de las cadenas (en vez de hacerlas todas iguales) son posibilidades interesantes a estudiar.

10.3. Resumen

En este capítulo hemos expuesto las principales conclusiones obtenidas y las aportaciones realizadas en el trabajo realizado en esta tesis y las diversas posibilidades de continuar la investigación dentro del área desarrollada a lo largo de este trabajo.

Apéndice A

Resultados complementarios

Este apéndice presenta una serie de gráficas de resultados omitidas en los capítulos de esta tesis a fin de evitar la saturación de gráficas.

A.1. Comparando inicializaciones para el método separado

Las siguientes gráficas corresponden a la comparación entre la inicialización voraz y por cadena mediana usando el método de optimización separado. Estos resultados completan los ofrecidos en el apartado 3.3.2.

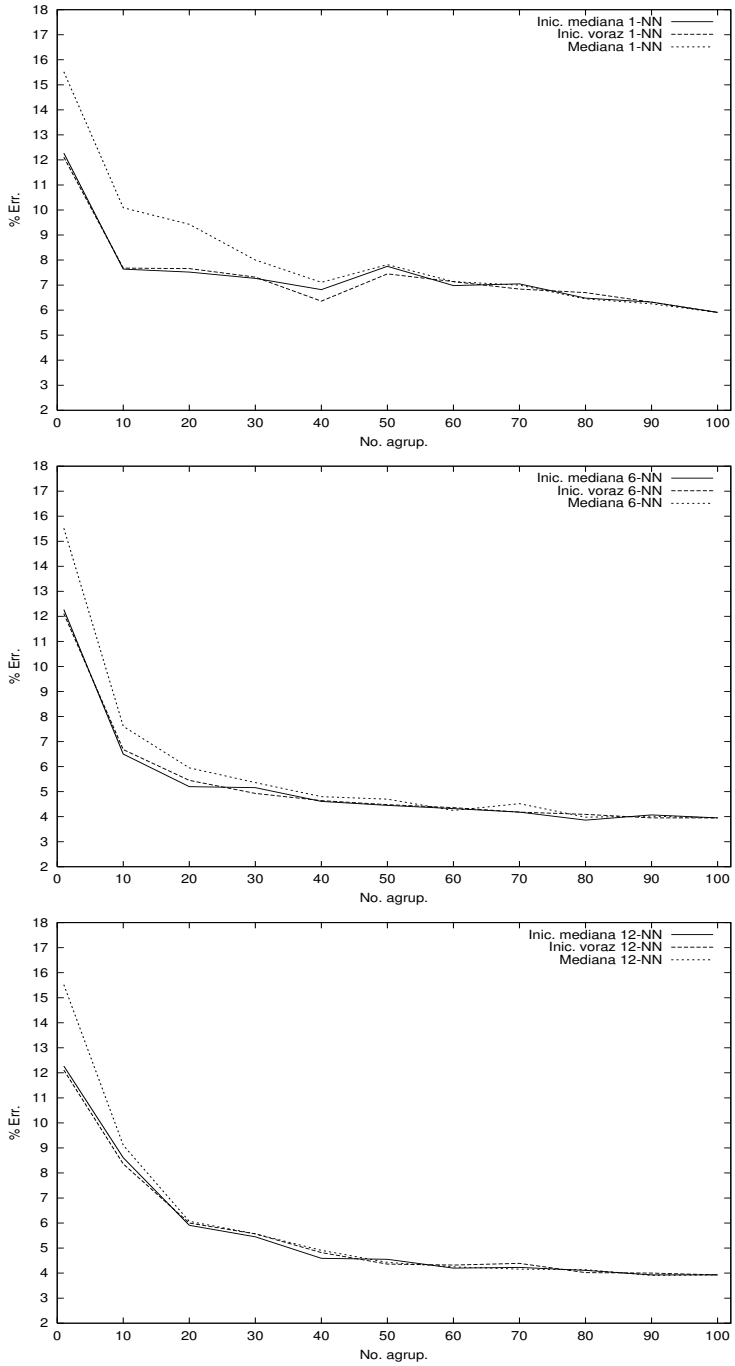


Figura A.1: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximada según la distinta inicialización, usando método iterativo separado para 1, 6 y 12-NN

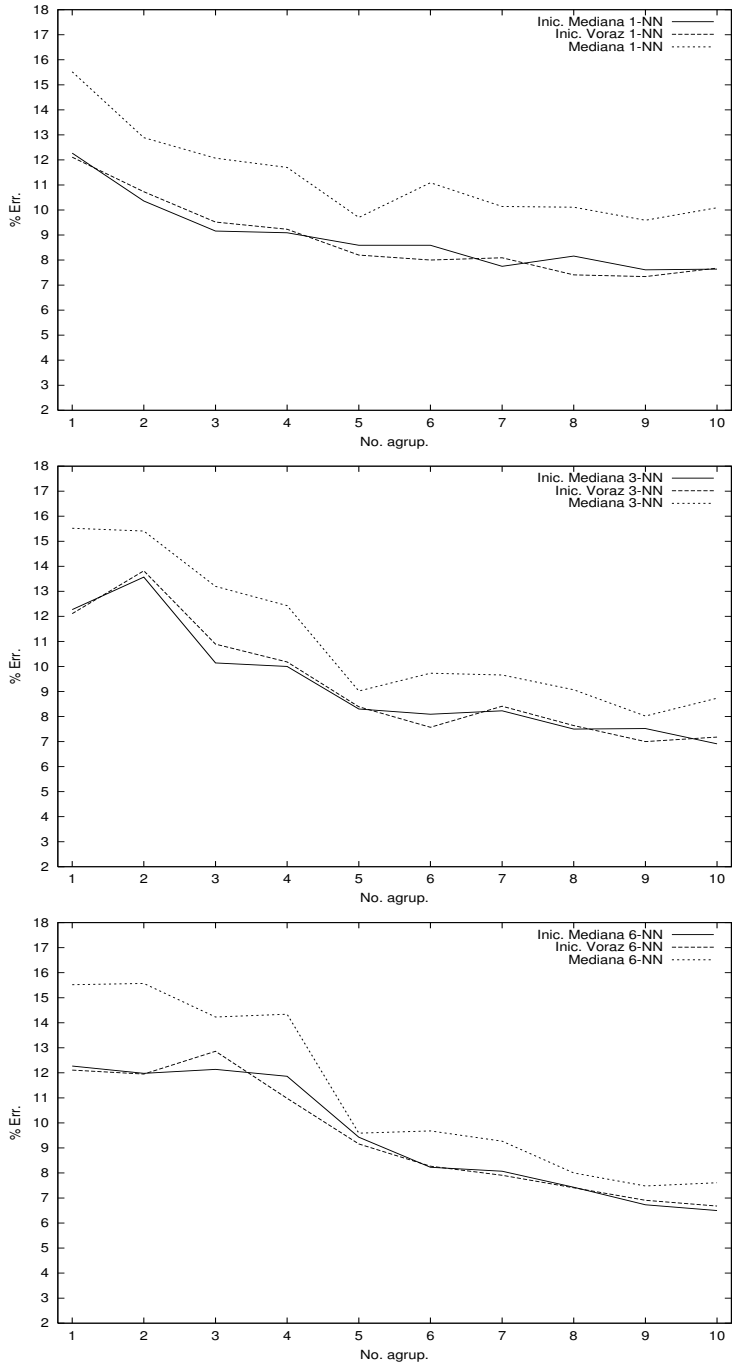


Figura A.2: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximada según la distinta inicialización, usando método iterativo separado para 1, 3 y 6-NN

A.2. Comparando métodos de optimización para inicialización voraz

Las siguientes gráficas corresponden a la comparación entre el método conjunto y separado de optimización usando como inicialización la cadena obtenida por el proceso voraz. Estos resultados completan los ofrecidos en el apartado 3.3.3.

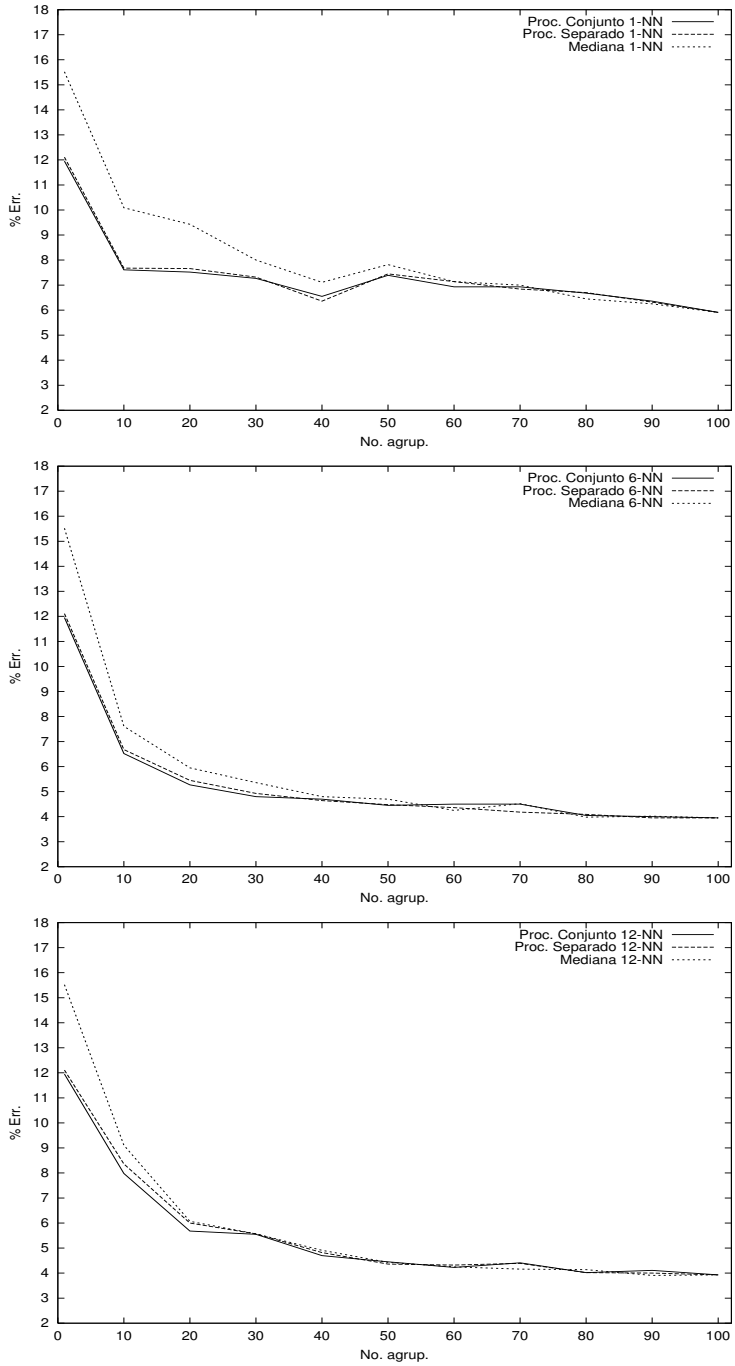


Figura A.3: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según los distintos procesos de optimización, usando la cadena voraz como inicialización, para 1, 6 y 12-NN

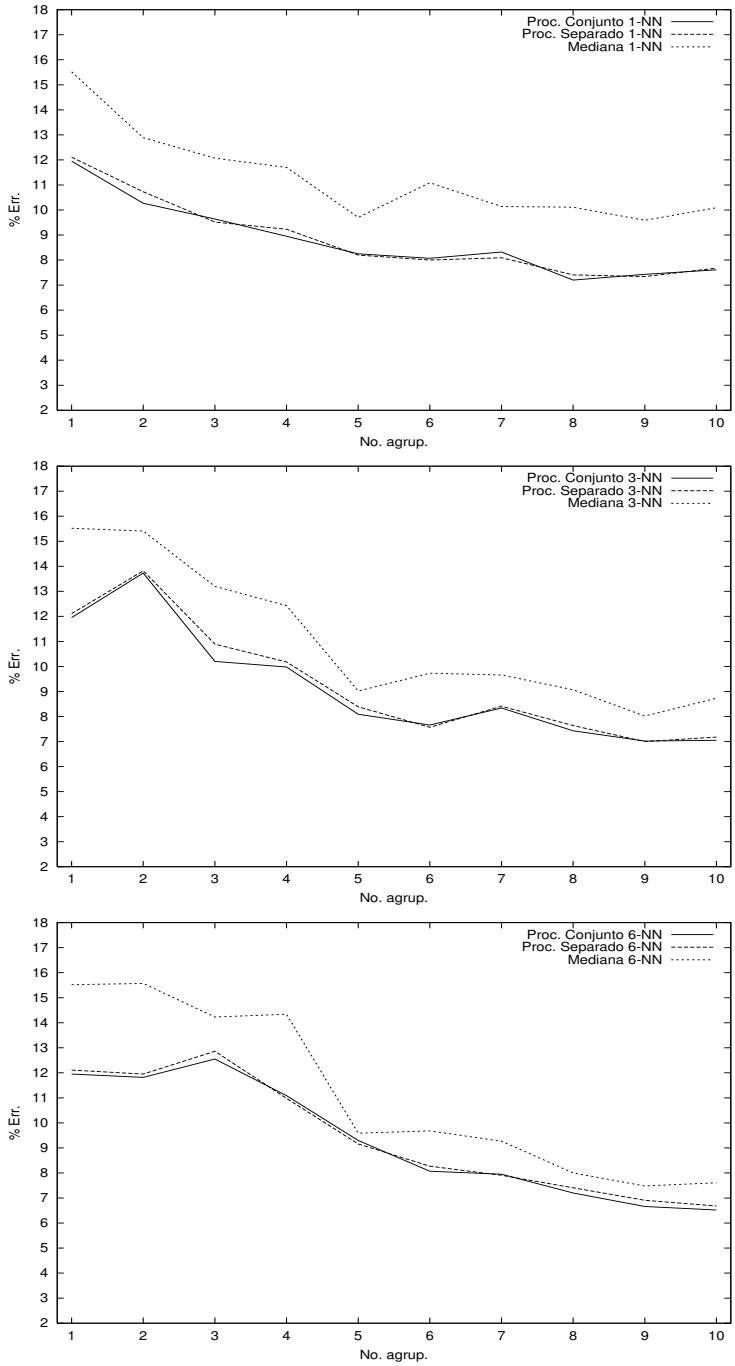


Figura A.4: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según los distintos procesos de optimización, usando la cadena voraz como inicialización, para 1, 3 y 6-NN

A.3. Comparando definiciones de cadena media para inicialización mediana y método separado

Las siguientes gráficas corresponden a la comparación entre la definición clásica y cuadrática de la cadena media usando como cadena inicial la cadena mediana y el método de optimización separado. Los resultados ofrecidos completan los aportados en la Sección 4.2.

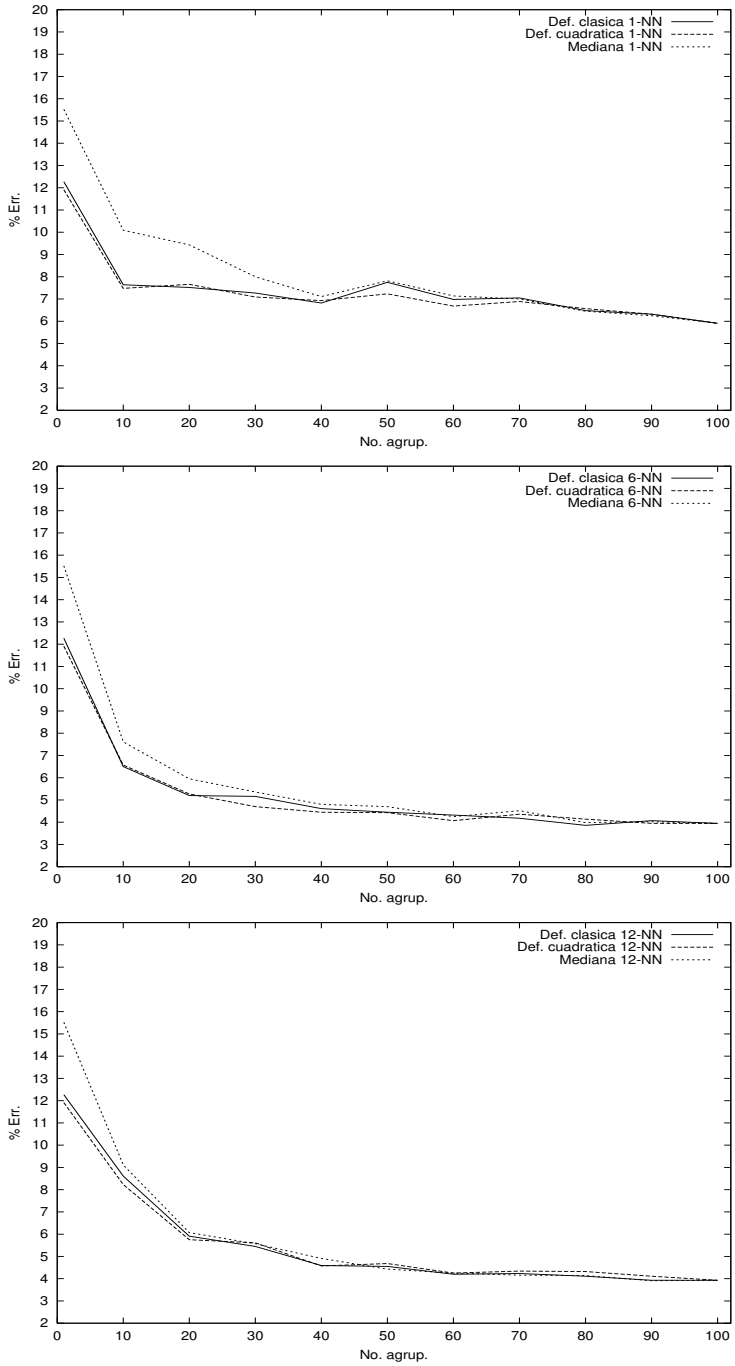


Figura A.5: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la definición de cadena media usada, usando la cadena mediana como inicialización y el método separado, para 1, 6 y 12-NN

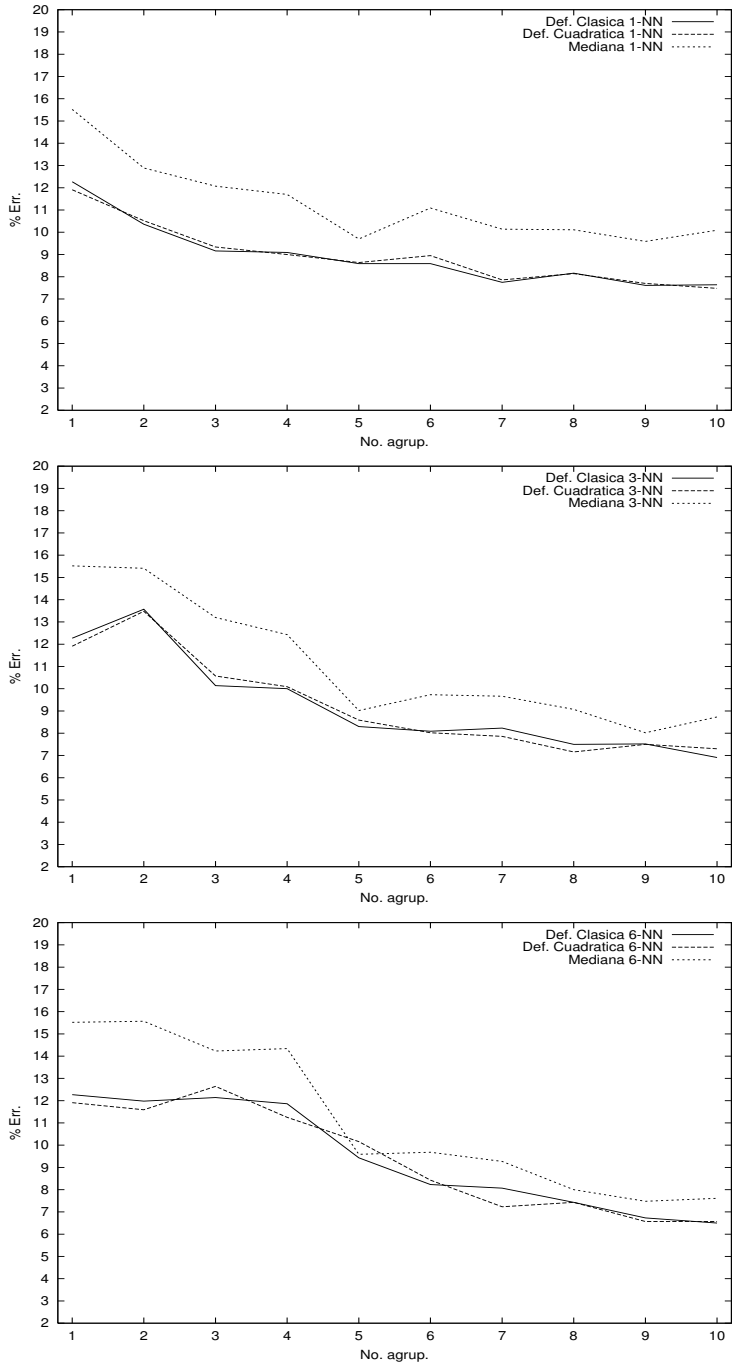


Figura A.6: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la definición de cadena media usada, usando la cadena mediana como inicialización y el método separado, para 1, 3 y 6-NN

A.4. Comparando definiciones de cadena media para inicialización voraz y método conjunto

Las siguientes gráficas corresponden a la comparación entre la definición clásica y cuadrática de la cadena media usando como cadena inicial la cadena obtenida por el proceso voraz y el método de optimización conjunto. Estos resultados completan los mostrados en las Secciones 4.2 y A.3.

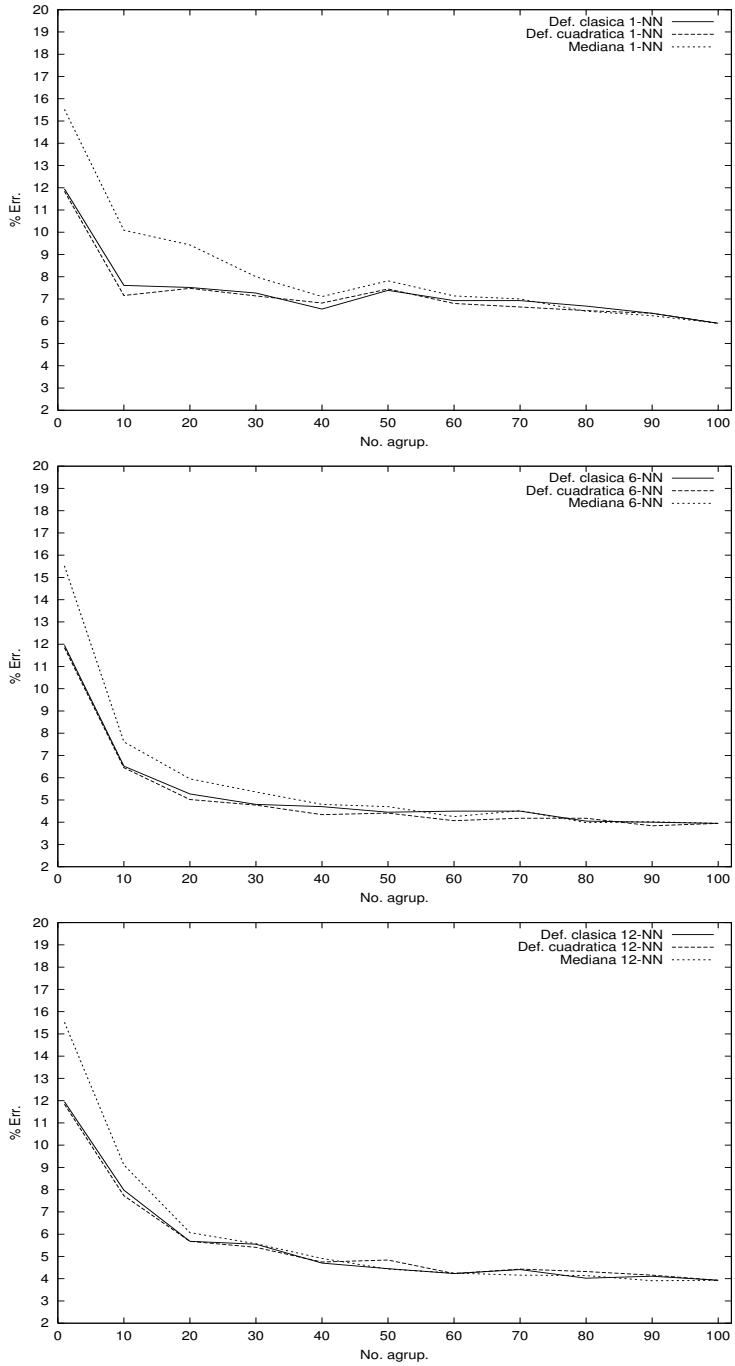


Figura A.7: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la definición de cadena media usada, usando la cadena voraz como inicialización y el método conjunto, para 1, 6 y 12-NN

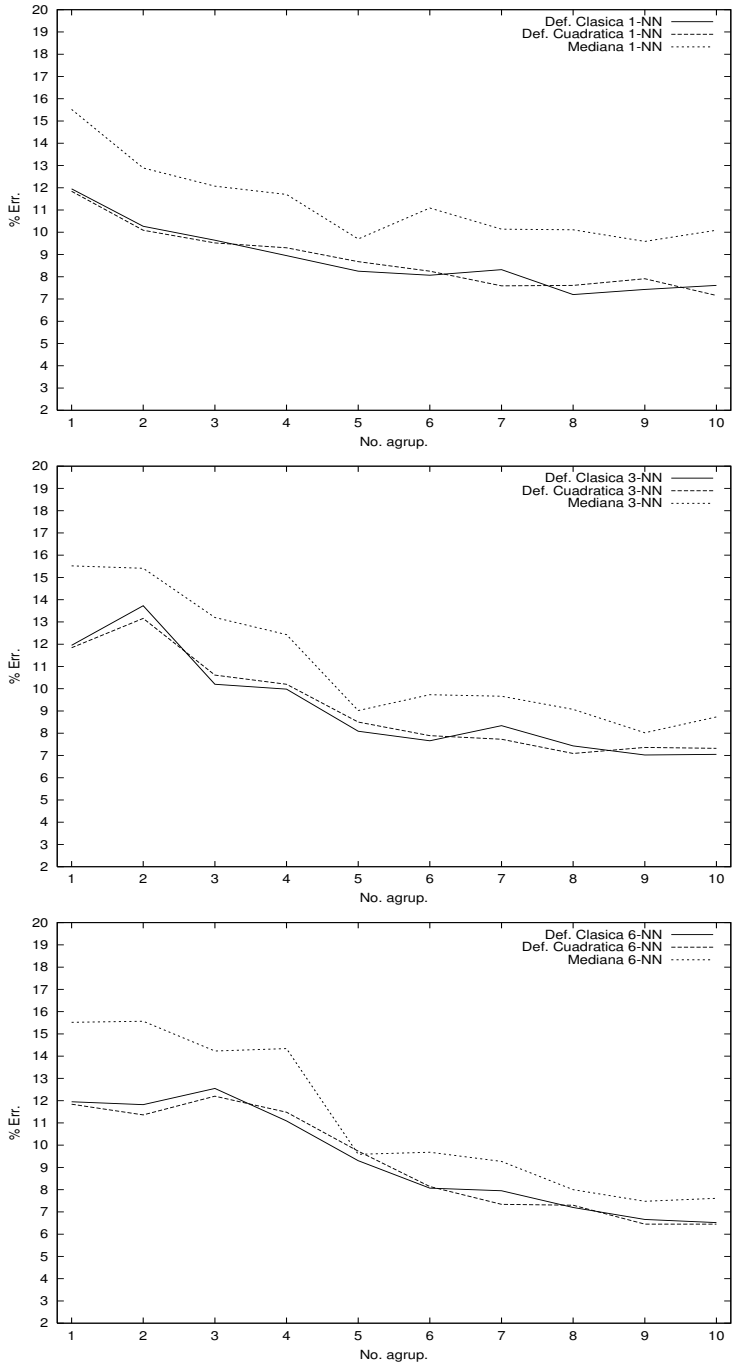


Figura A.8: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la definición de cadena media usada, usando la cadena voraz como inicialización y el método conjunto, para 1, 3 y 6-NN

A.5. Comparando definiciones de cadena media para inicialización voraz y método separado

Las siguientes gráficas corresponden a la comparación entre la definición clásica y cuadrática de la cadena media usando como cadena inicial la cadena obtenida por el proceso voraz y el método de optimización separado. Con estos resultados se completan los ofrecidos en las Secciones 4.2, A.3 y A.4.

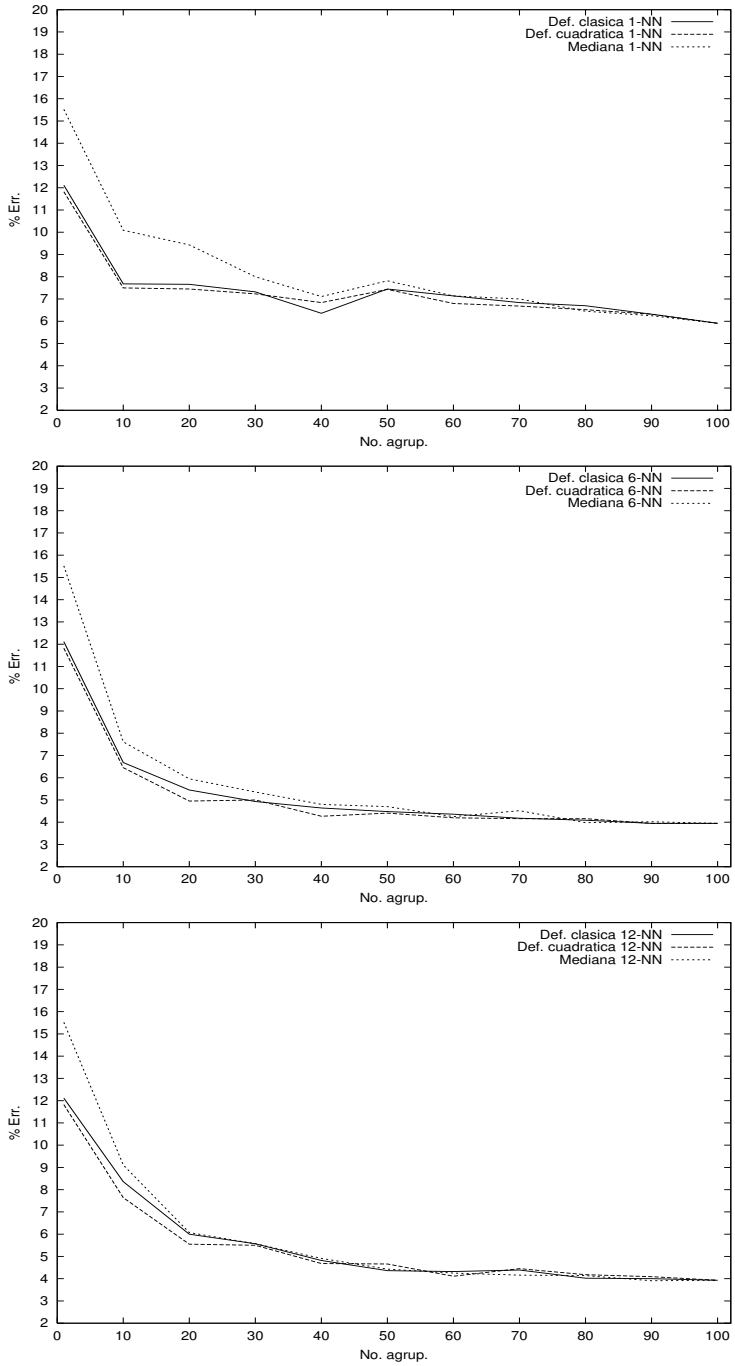


Figura A.9: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la definición de cadena media usada, usando la cadena voraz como inicialización y el método separado, para 1, 6 y 12-NN

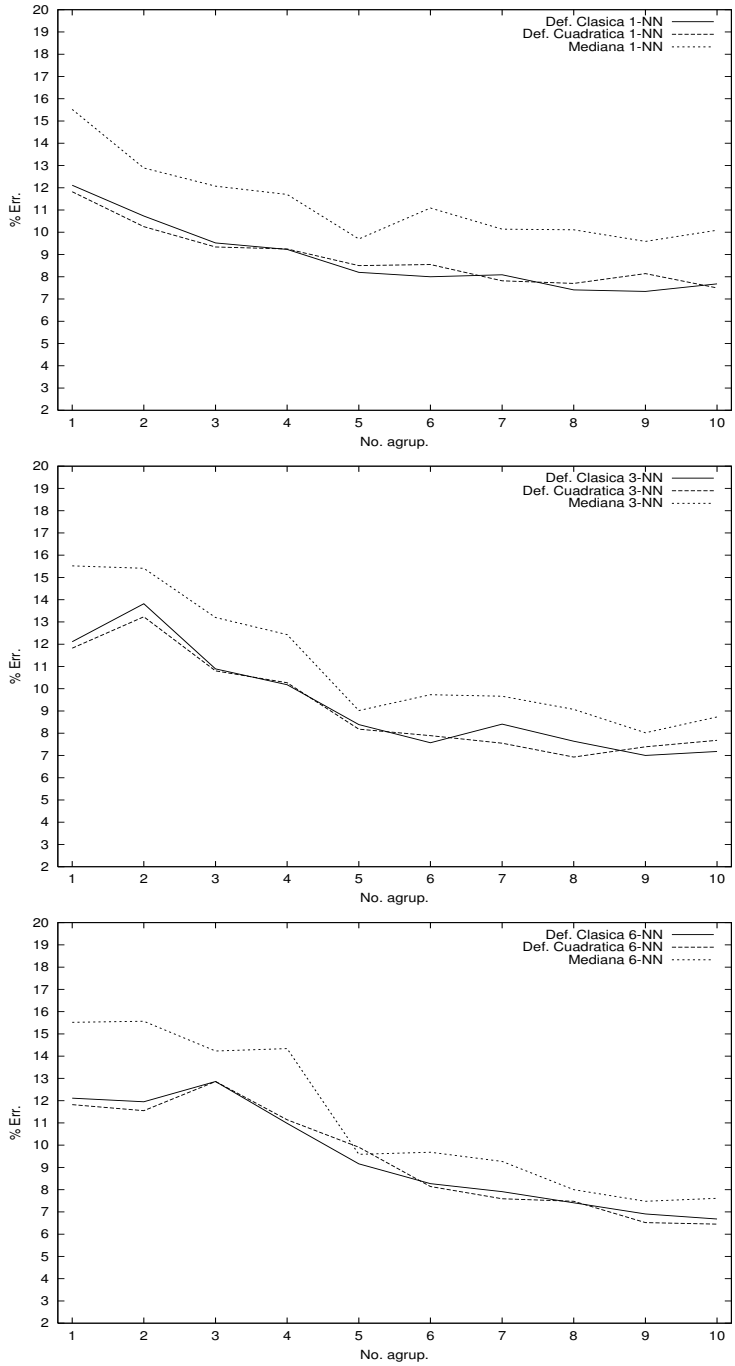


Figura A.10: Resultados de clasificación usando la cadena mediana y las cadenas medias aproximadas según la definición de cadena media usada, usando la cadena voraz como inicialización y el método separado, para 1, 3 y 6-NN

A.6. Comparando las diversas optimizaciones temporales

Las siguientes gráficas corresponden a la comparación entre las cadenas obtenidas sin métodos de optimización y las obtenidas usando los diversos métodos de optimización usando como cadena inicial la cadena mediana y el método de optimización conjunto. Con estos resultados se complementan los ofrecidos en la Sección 5.4.

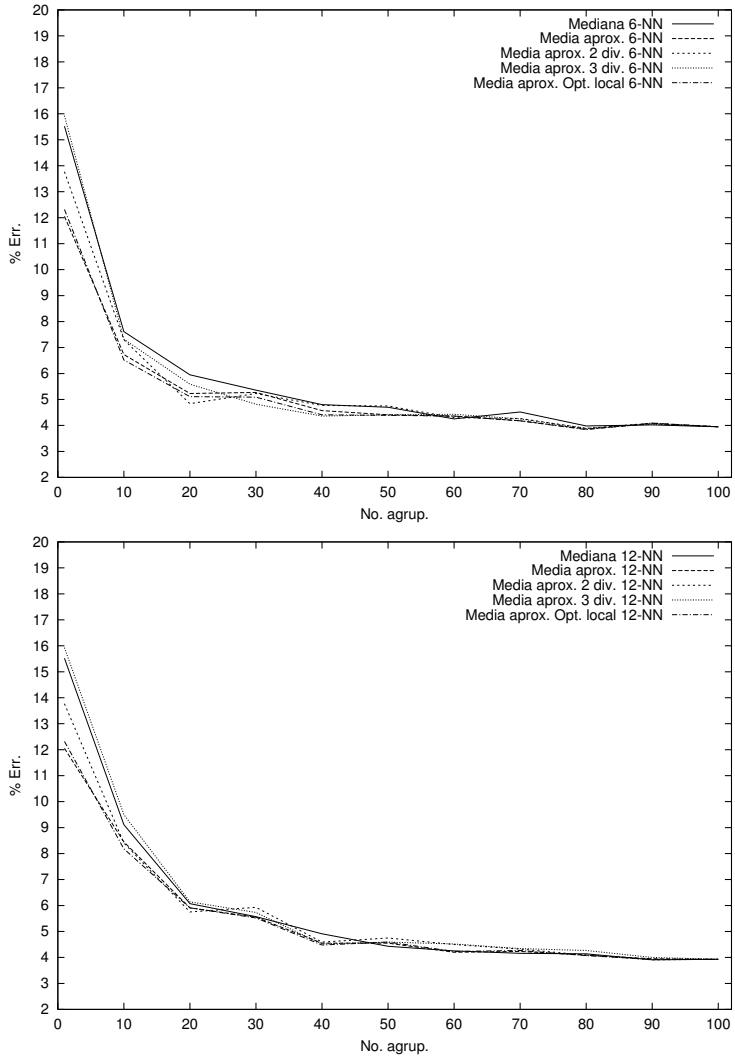


Figura A.11: Resultados de clasificación usando la cadena mediana, la cadena media aproximada sin optimizar y la cadenas medias aproximadas obtenidas por optimización local y 2 y 3 divisiones, usando la cadena mediana como inicialización y el método conjunto, para 6 y 12-NN

A.7. Comparando los diversos tipos de agrupamiento para cadena mediana

Las siguientes gráficas corresponden a las comparación de los resultados obtenidos con los diversos métodos de obtención de agrupamientos usando la cadena mediana de cada uno de los agrupamientos obtenidos como prototipo en clasificación. Estos resultados completan los del apartado 7.3.1, específicamente los de las Figuras 7.5 y 7.8.

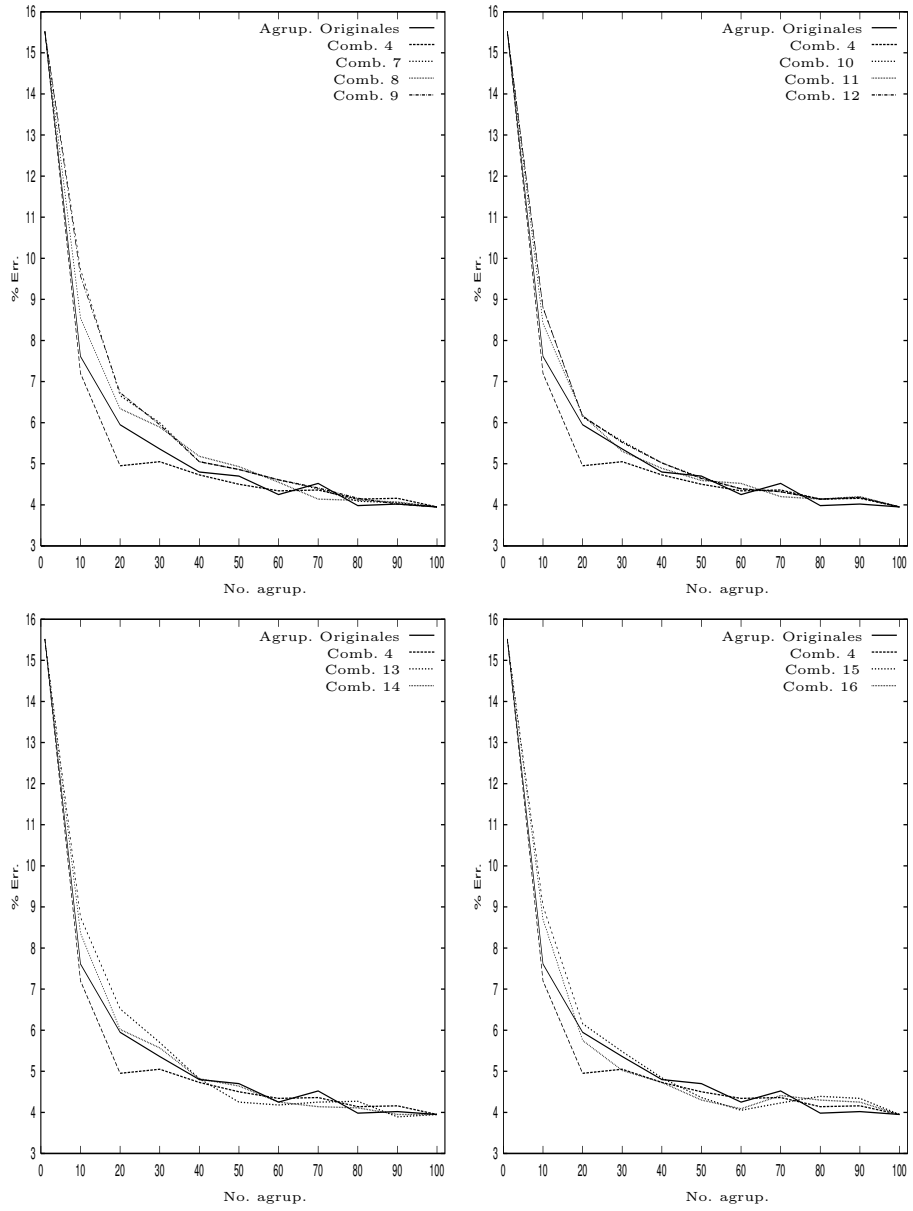


Figura A.12: Resultados de clasificación para 6NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

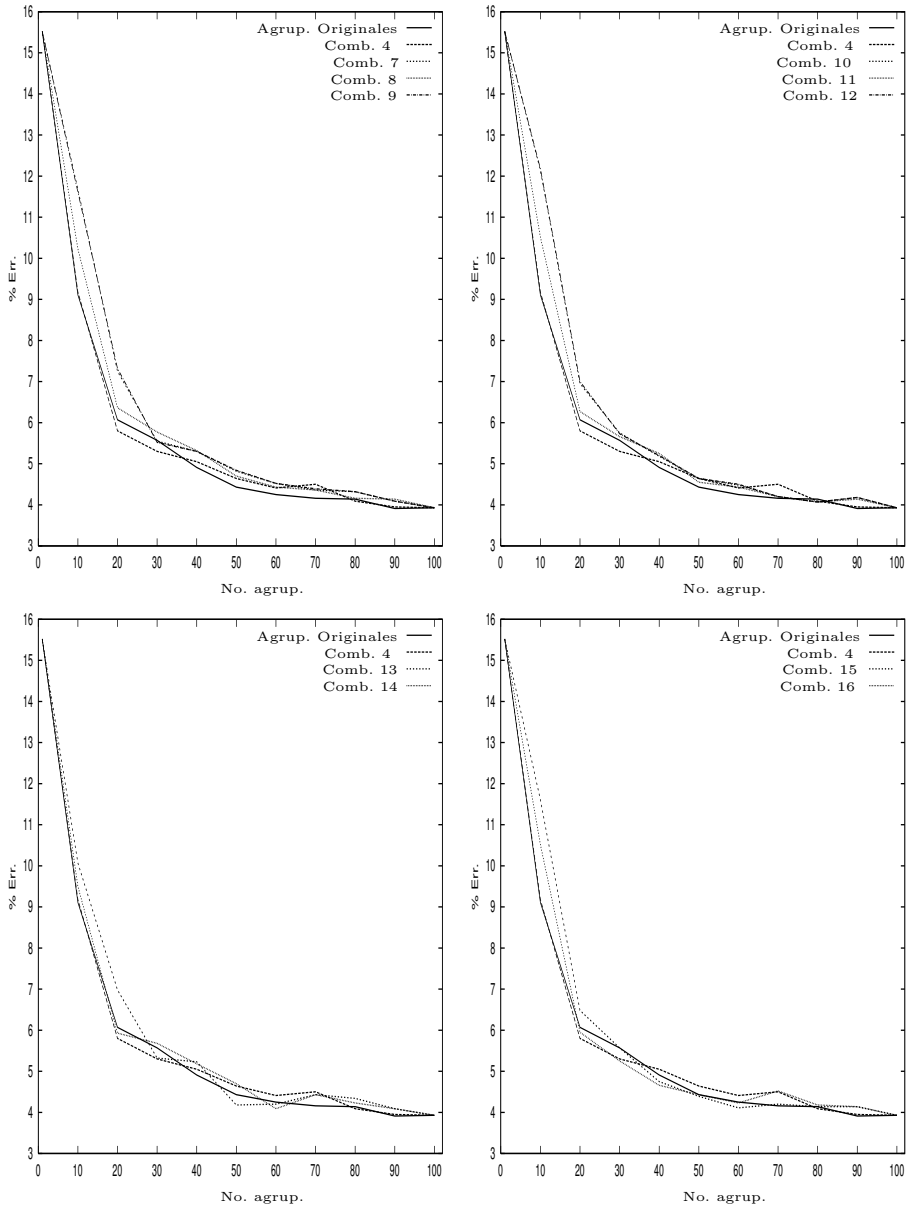


Figura A.13: Resultados de clasificación para 12NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

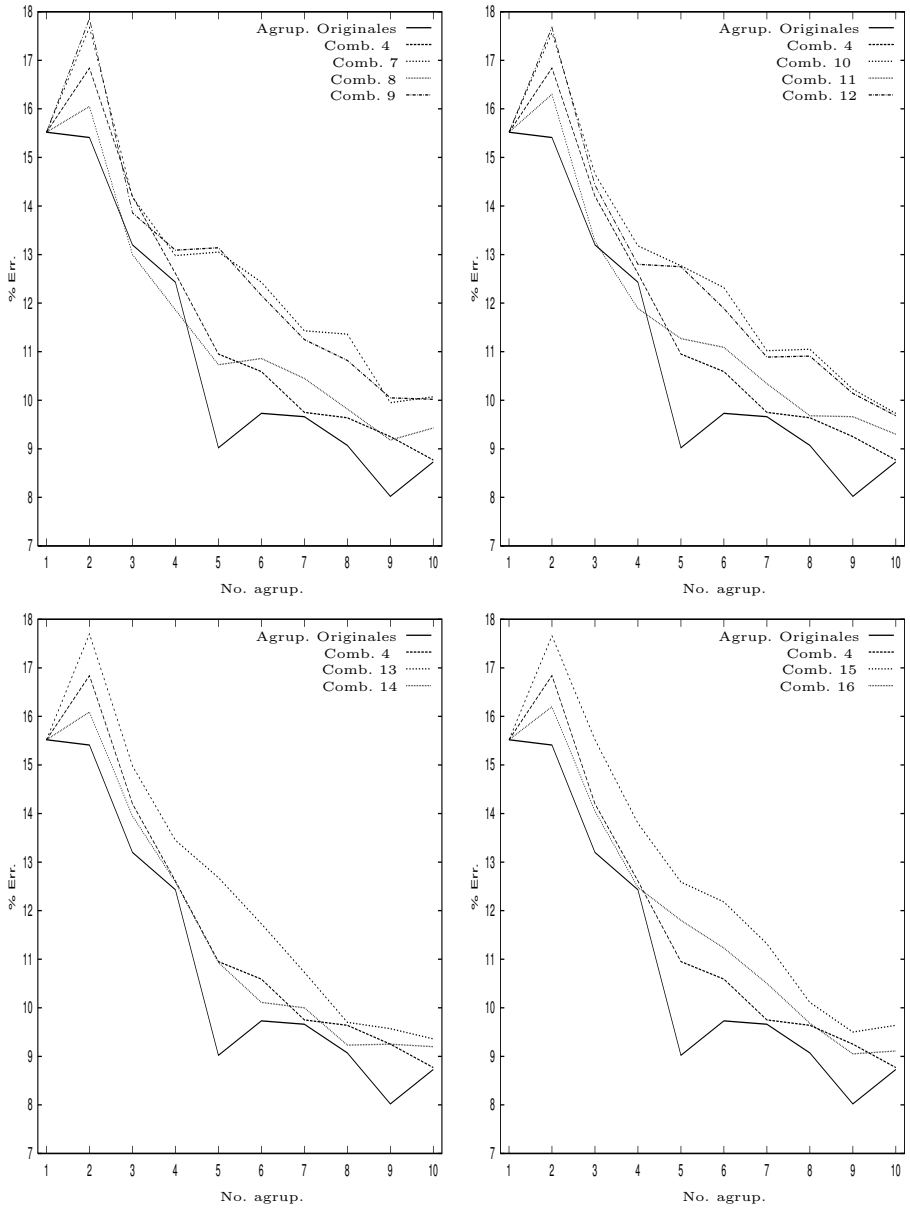


Figura A.14: Resultados de clasificación para 3NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

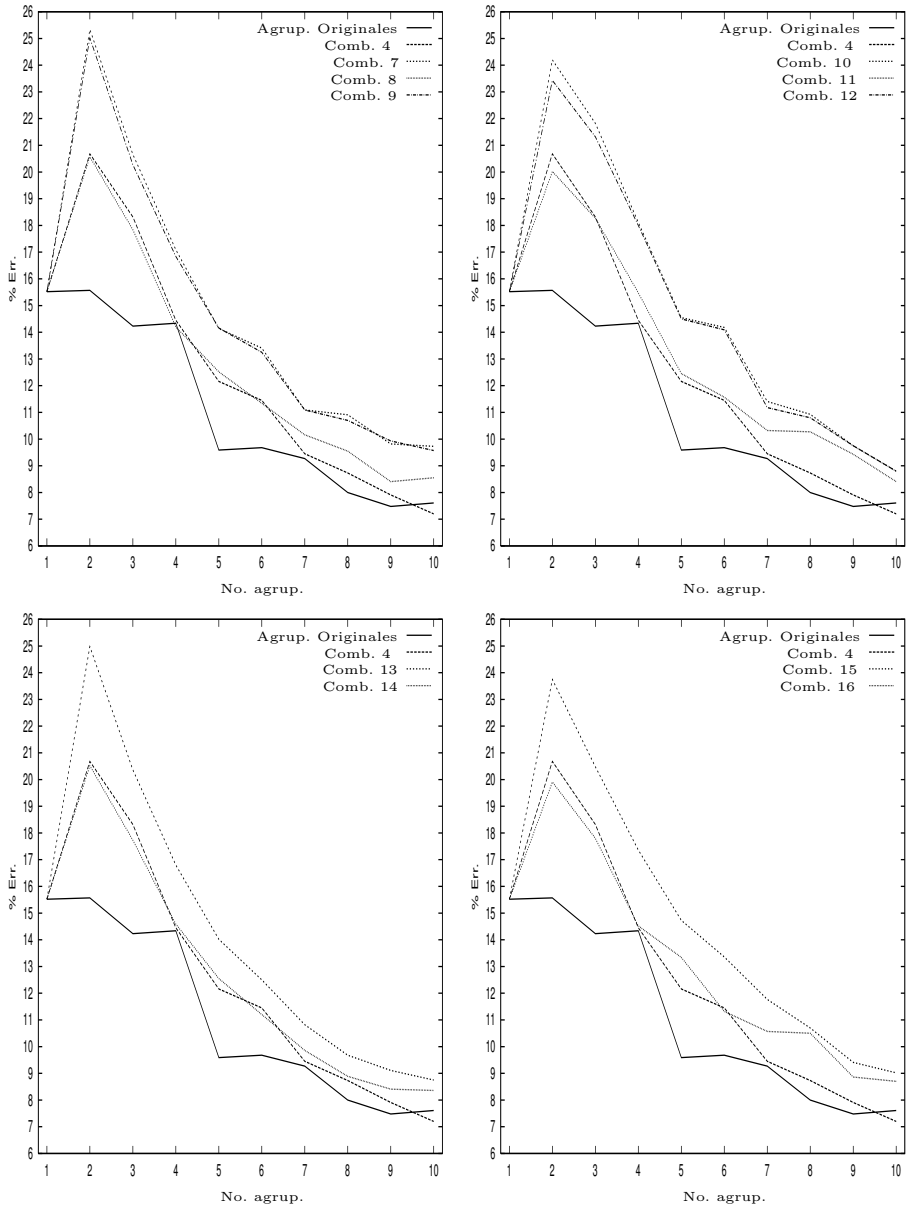


Figura A.15: Resultados de clasificación para 6NN usando la cadena mediana como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

A.8. Comparando los diversos tipos de agrupamiento para cadena media por método conjunto

Las siguientes gráficas corresponden a las comparación de los resultados obtenidos con los diversos métodos de obtención de agrupamientos usando la cadena media obtenida por el método conjunto de cada uno de los agrupamientos obtenidos como prototipo en clasificación. Estos resultados completan los del apartado 7.3.2, específicamente los de las Figuras 7.11 y 7.14.

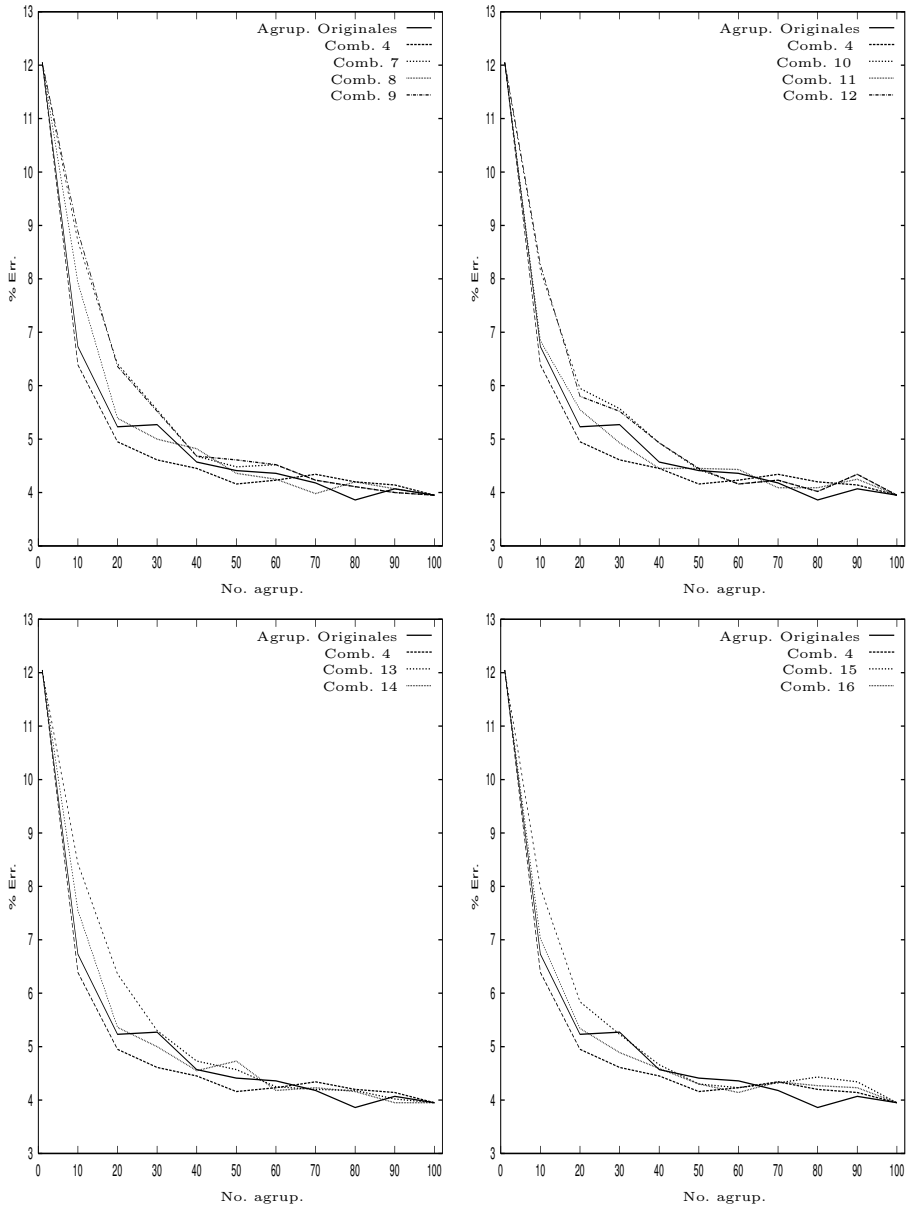


Figura A.16: Resultados de clasificación para 6NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

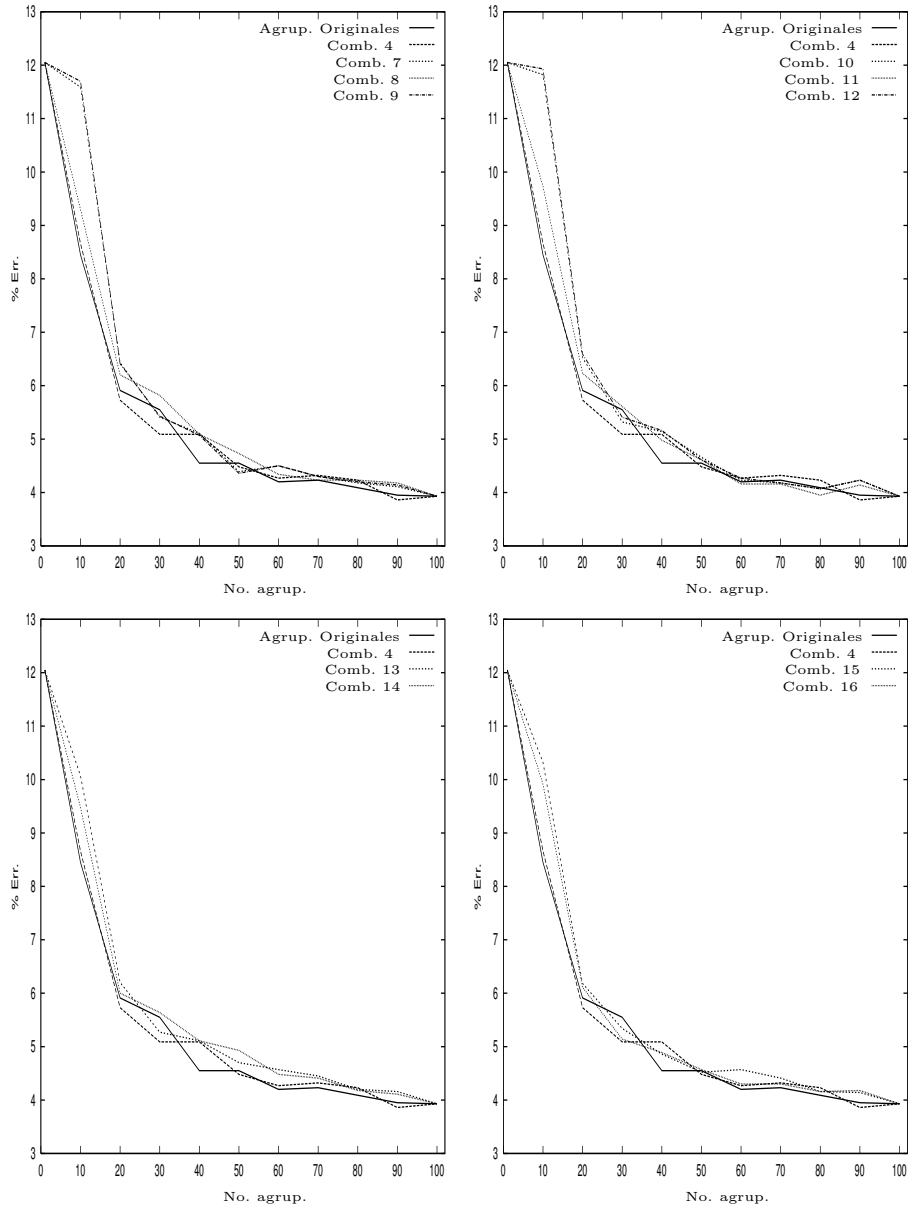


Figura A.17: Resultados de clasificación para 12NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

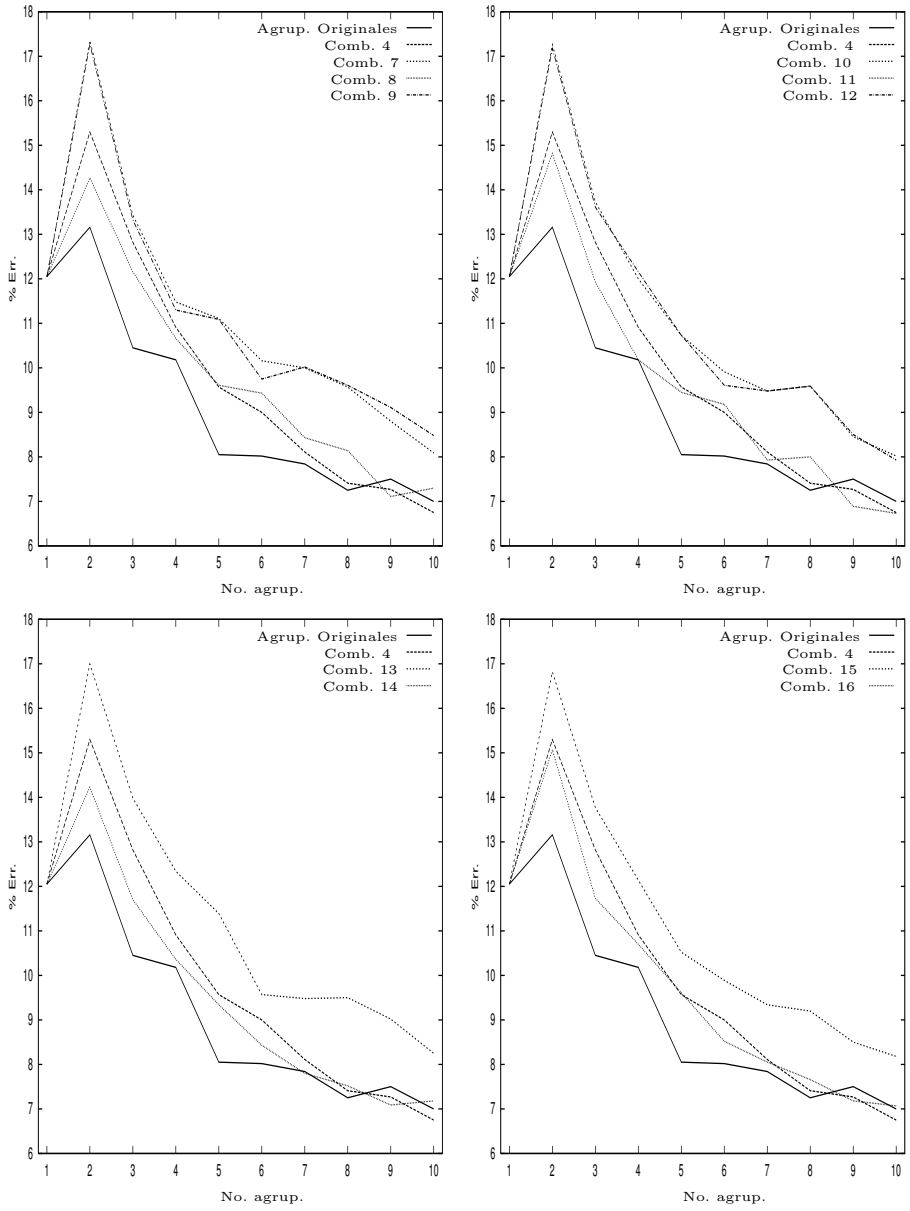


Figura A.18: Resultados de clasificación para 3NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

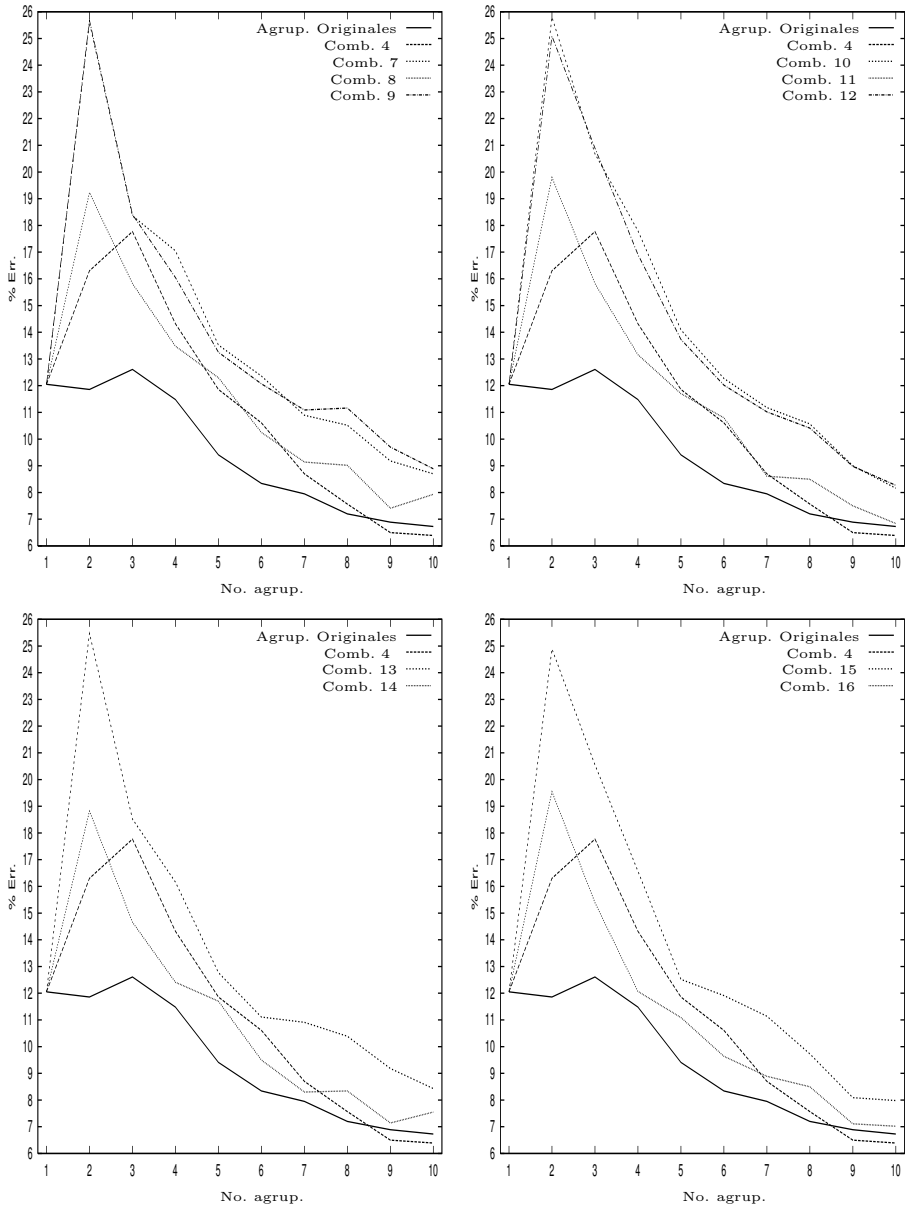


Figura A.19: Resultados de clasificación para 6NN usando la cadena media aproximada por método conjunto como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

A.9. Comparando los diversos tipos de agrupamiento para cadena media por método separado

Las siguientes gráficas corresponden a las comparación de los resultados obtenidos con los diversos métodos de obtención de agrupamientos usando la cadena media obtenida por el método separado de cada uno de los agrupamientos obtenidos como prototipo en clasificación. Estos resultados completan los del apartado 7.3.3, específicamente los de las Figuras 7.17 y 7.20.

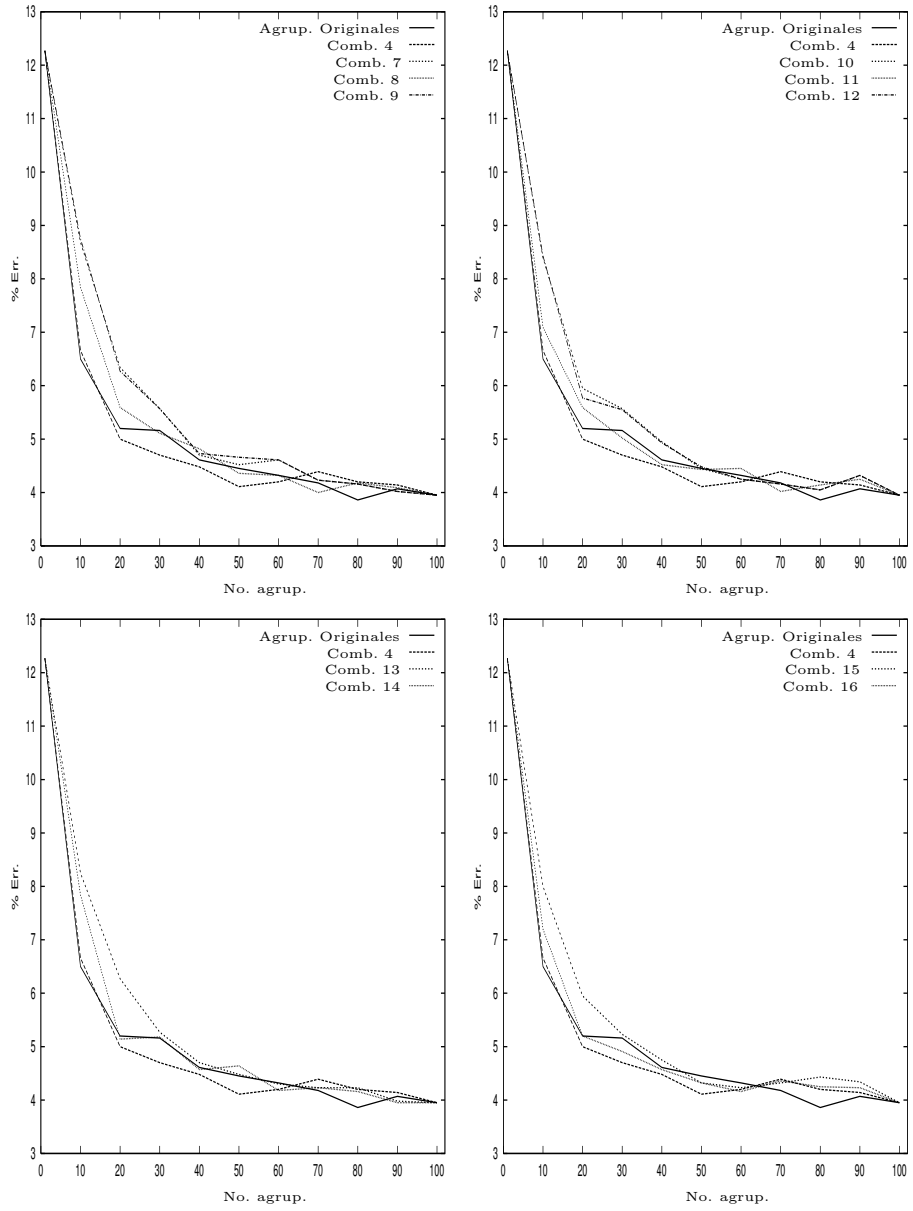


Figura A.20: Resultados de clasificación para 6NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

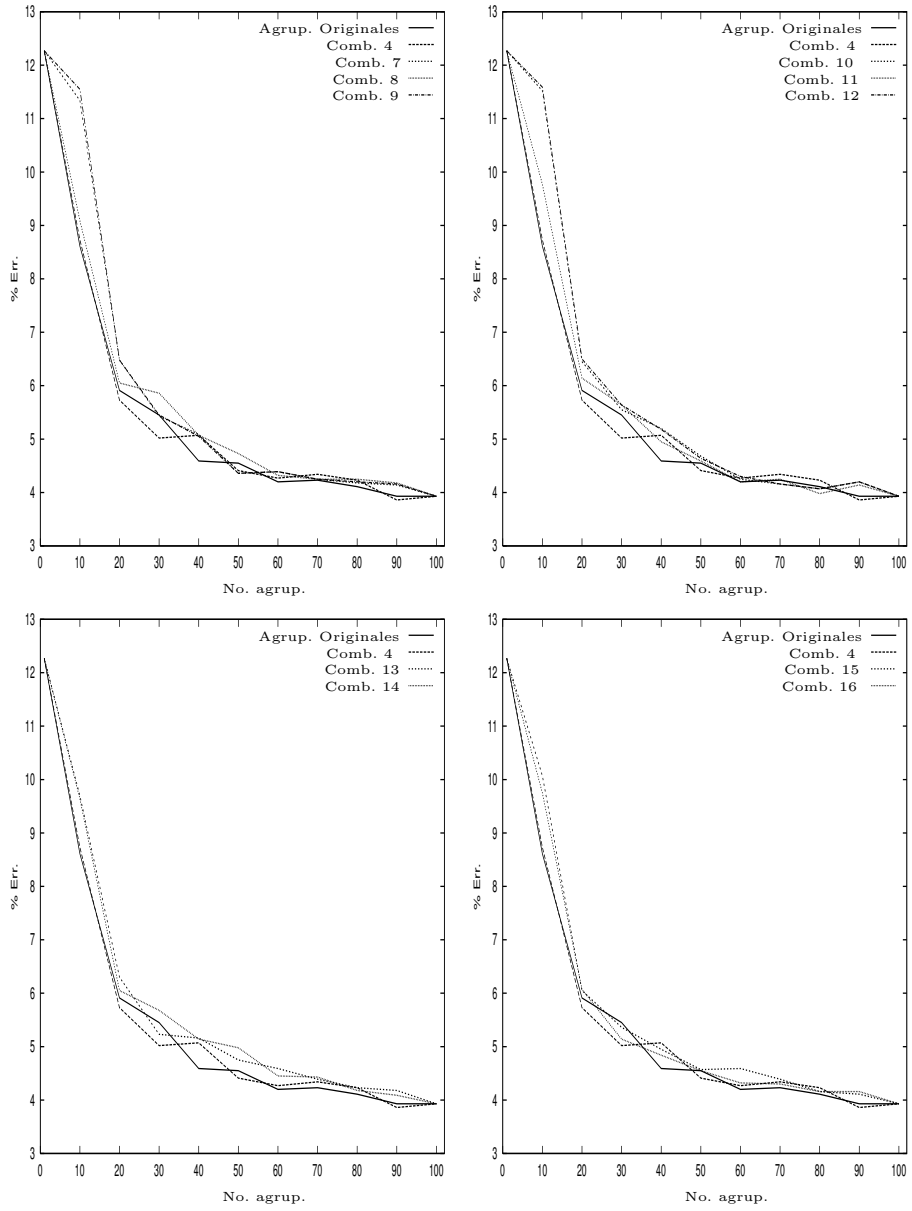


Figura A.21: Resultados de clasificación para 12NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

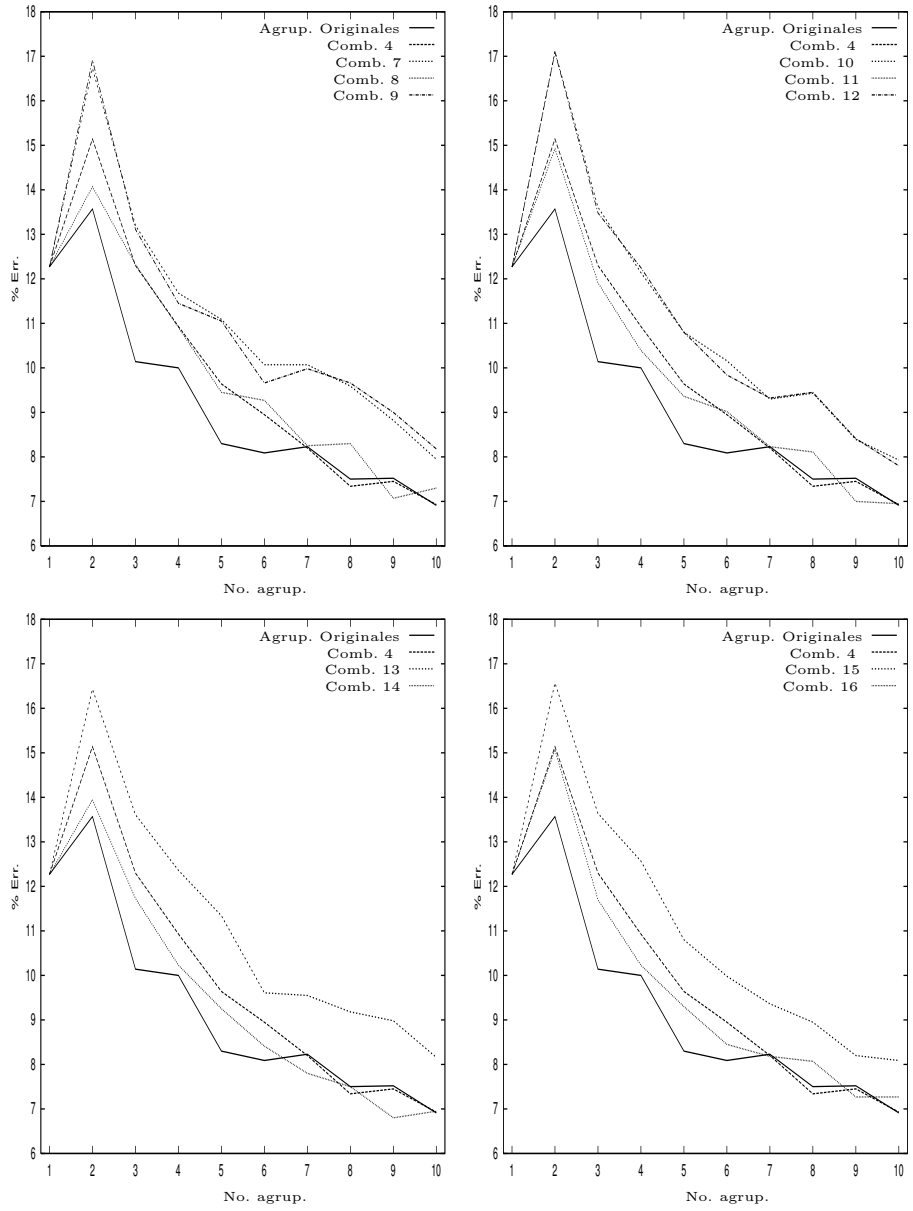


Figura A.22: Resultados de clasificación para 3NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

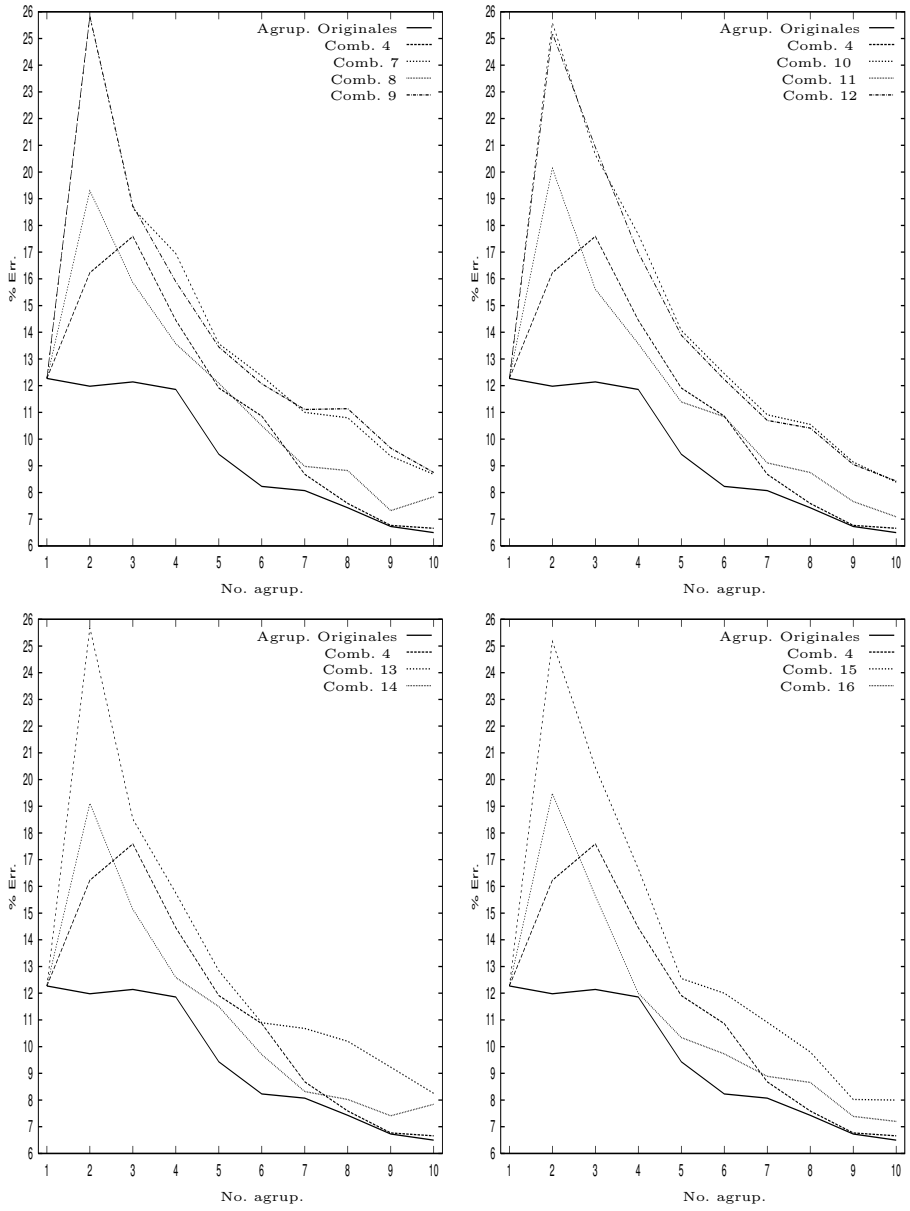


Figura A.23: Resultados de clasificación para 6NN usando la cadena media aproximada por método separado como prototipo a partir de los agrupamientos obtenidos usando la inicialización con maxmin modificado usando k-medianas (todas las gráficas, combinación 4), k-medias por representante conjunto (1.ª gráfica, combinaciones 7, 8 y 9) y separado (2.ª gráfica, combinaciones 10, 11 y 12), k-medias por mediana conjunto (3.ª gráfica, combinaciones 13 y 14) y separado (4.ª gráfica, combinaciones 15 y 16), con todos los posibles reagrupamientos (k-medianas y k-medias generalizado). Se muestran los resultados con los agrupamientos originales para comparar.

Apéndice B

Experimentos complementarios

En este apéndice se pretenden dar unos resultados complementarios para generalizar las conclusiones obtenidas en los Capítulos 3 y 5 sobre las características como prototipos de las cadenas medias aproximadas. Para ello, se realizará un trabajo preliminar sobre un corpus de cromosomas, conocido como *Cpr*, de mucho mayor tamaño, con el fin de confirmar dichas conclusiones.

B.1. Corpus de cromosomas *Cpr*

Otro corpus de cromosomas utilizado para la prueba de técnicas de clasificación en RF es el llamado *Cpr* [63]. Este corpus comprende las imágenes de los cariotipos de la metafase de 2804 células distintas, siendo 1344 femeninas y 1460 masculinas. De este total de células, existen algunas que presentan ciertas anomalías, con lo cual sólo se consideran como normales un total de 2740 células. Para nuestra experimentación, únicamente consideraremos las células femeninas (a fin de poder usar sin problemas los cromosomas sexuales en la clasificación) con una estructura normal, lo que da un total de 1318 células. Como cada célula dispone de 46 cromosomas (23 pares), se tiene un total de 60628 cromosomas disponibles (2636 por clase). Este conjunto de datos ha sido utilizado en varios trabajos experimentales de clasificación de cromosomas [68, 67, 66, 69]. Los resultados obtenidos en dichos trabajos presentan una tasa de error muy baja (en torno al 1 %), utilizando técnicas de codificación y clasificación muy sofisticadas. Queda fuera de nuestros objetivos alcanzar resultados de tal calidad, buscando únicamente confirmar la ventaja que presentan las aproximaciones a la cadena media frente a la cadena mediana como prototipo en clasificación.

Por tanto, la codificación del corpus en cadenas de símbolos se hace de manera muy semejante a como hemos descrito en el apartado 2.2.1, es decir, mediante la extracción de la traza del cromosoma a lo largo de su eje longitudinal. En este caso, al ser mayor el número de cromosomas, se sacrifica la idoneidad de los mis-

Cuadro B.1: Características del corpus *Cpr*

Número de clases	23
Número de objetos	60628
Tamaño del alfabeto	201
Longitud de las cadenas (mínima-máxima)	13-124

mos; es decir, no todos los cromosomas se presentan alineados en torno a un eje longitudinal más o menos recto, sino que varios de ellos presentan alineamientos extravagantes (se encuentran muy deformados respecto al eje longitudinal ideal), o también existen imágenes tomadas en partes tardías de la metafase en las que las cromátidas son difíciles de apreciar. Este es el motivo principal por el cual en los trabajos más exhaustivos se han aplicado otras técnicas de extracción de características para cromosomas [69]. Debido a la complejidad de las mismas y a la naturaleza preliminar de nuestros experimentos, seguiremos usando la codificación a partir de trazas del cromosoma.

Las trazas se consiguen a partir de ficheros de las imágenes de los cromosomas en niveles de gris (en formato PGM) con una precisión de 8 bits (256 niveles distintos). La codificación numérica se hace promediando los niveles de gris del eje normal muestreado, lo cual hace que el alfabeto de dicha codificación sea $\{0, 1, \ldots, 254, 255\}$. Como nuevamente lo interesante es marcar los contrastes entre las bandas de los cromosomas, se recurre a la codificación por diferencias, asumiendo que el nivel 0 está al inicio y el fin de la codificación numérica intermedia y construyendo la cadena final como la diferencia entre el número actual y el siguiente. Con dicha codificación tendríamos un alfabeto de 511 símbolos, $\Sigma = \{-255, -254, \ldots, -1, 0, 1, \ldots, 254, 255\}$, pero considerando que una diferencia de más de 100 entre símbolos es lo suficientemente extrema, podemos codificar como -100 cualquier símbolo de menor valor y como 100 cualquiera de mayor valor. Por tanto, el alfabeto final de la codificación en cadenas de diferencias es $\Sigma = \{-100, -99, \ldots, -1, 0, 1, \ldots, 99, 100\}$. Las características resumidas del corpus se describen en el Cuadro B.1.

B.2. Experimentos con el corpus *Cpr*

En el corpus *Cpr* la longitud de las cadenas es semejante al corpus *Copenhagen*, pero el tamaño del alfabeto es mayor (un orden de magnitud mayor, pues pasamos de 11 a 201 símbolos). Esto hace que sea inviable una experimentación exhaustiva con todos los tipos posibles de prototipos, lo cual hace que nos limitemos a una experimentación con la cadena mediana, la media aproximada con inicialización por cadena mediana y proceso conjunto, y esa misma media aproximada pero usando optimización local (la cual permitirá, claramente, una ganancia temporal extraordinaria en este caso).

Respecto a la realización de los agrupamientos, y debido al carácter prelimi-

nar de estos experimentos, se ha tomado la decisión de hacerlo de manera trivial: fijado el número de agrupamientos c, para cada clase se introduce el i-ésimo cromosoma de la lista en el agrupamiento $i \mod c$. Debido a que la cantidad de datos es considerable (hablamos de más 2600 muestras de cada clase), se puede asumir que los agrupamientos serán lo suficientemente representativos de los diversos elementos de cada clase. Se ha optado por la realización de los agrupamientos para 10, 100 y 1000 agrupamientos, lo cual nos da un promedio de 265'4, 26'5 y 2'7 cromosomas por agrupamiento, respectivamente.

En cuanto al uso de la validación cruzada, en este corpus se ha optado por hacerlo en base a los propios agrupamientos realizados para la extracción de prototipos. De esta manera, si se han realizado c agrupamientos por clase, escogeremos los datos del agrupamiento de índice i (para todas las clases) como datos de prueba y los prototipos obtenidos de los $c-1$ agrupamientos restantes como los auténticos prototipos a usar por el clasificador k-NN. Como la creación de los agrupamientos se ha realizado por el método trivial, es de suponer que no habrá diferencias significativas en los resultados de clasificación al clasificar un agrupamiento con los prototipos obtenidos del resto de agrupamientos.

La matriz de pesos usada se define por la diferencia absoluta entre los dos símbolos a sustituir en el caso de los pesos de sustitución (es decir, sustituir -9 por 17 tendría un peso de 26). Con esto se consigue que la probabilidad de sustitución de símbolos que representan variaciones de brillo semejantes sea alta (peso de sustitución bajo), mientras que la de las variaciones de brillo muy diferentes tiende a ser menos probable (peso de sustitución alto). Las inserciones y borrados de un símbolo tienen como peso el promedio de los pesos de sustitución de dicho símbolo, garantizando de esa manera que los símbolos que indican una evolución más uniforme del brillo tienen mayor probabilidad de darse en la secuencia que representa el cromosoma.

Para las clasificaciones se ha usado un clasificador k-NN con valores de k desde 1 hasta 10. Dicho clasificador se aplica usando como prototipos las cadenas extraídas de cada agrupamiento (tal y como se han definido previamente).

Para cada agrupamiento, se extrajeron la cadena mediana, la media aproximada con inicialización mediana y proceso conjunto, y la media aproximada con optimización local con inicialización por cadena mediana y proceso conjunto. Los resultados de clasificación, para 10, 100 y 1000 agrupamientos, con k variando de 1 a 10, se pueden ver en las gráficas de la Figura B.1. En estas gráficas no se muestran los intervalos de confianza debido a que la gran cantidad de datos con la que se trabaja hace que, al obtener dichos intervalos mediante la aproximación a una distribución normal, el intervalo de confianza sea prácticamente nulo (es decir, los resultados son significativos por sí mismos).

Como se puede ver, el uso de la cadena media aproximada provoca una mejora en la clasificación (para cualquier valor de k de los tomados) con respecto al uso de la cadena mediana. Para 10 agrupamientos, la mejora es cercana a 2 puntos en términos absolutos en varios casos, lo cual es aproximadamente un 9 % de mejora en clasificación. Para el mejor resultado de ambos, la diferencia absoluta es de 1'3 puntos de error (un 5'7 % de mejora en términos relativos).

Respecto a los resultados con 100 agrupamientos, lo primero que llama la

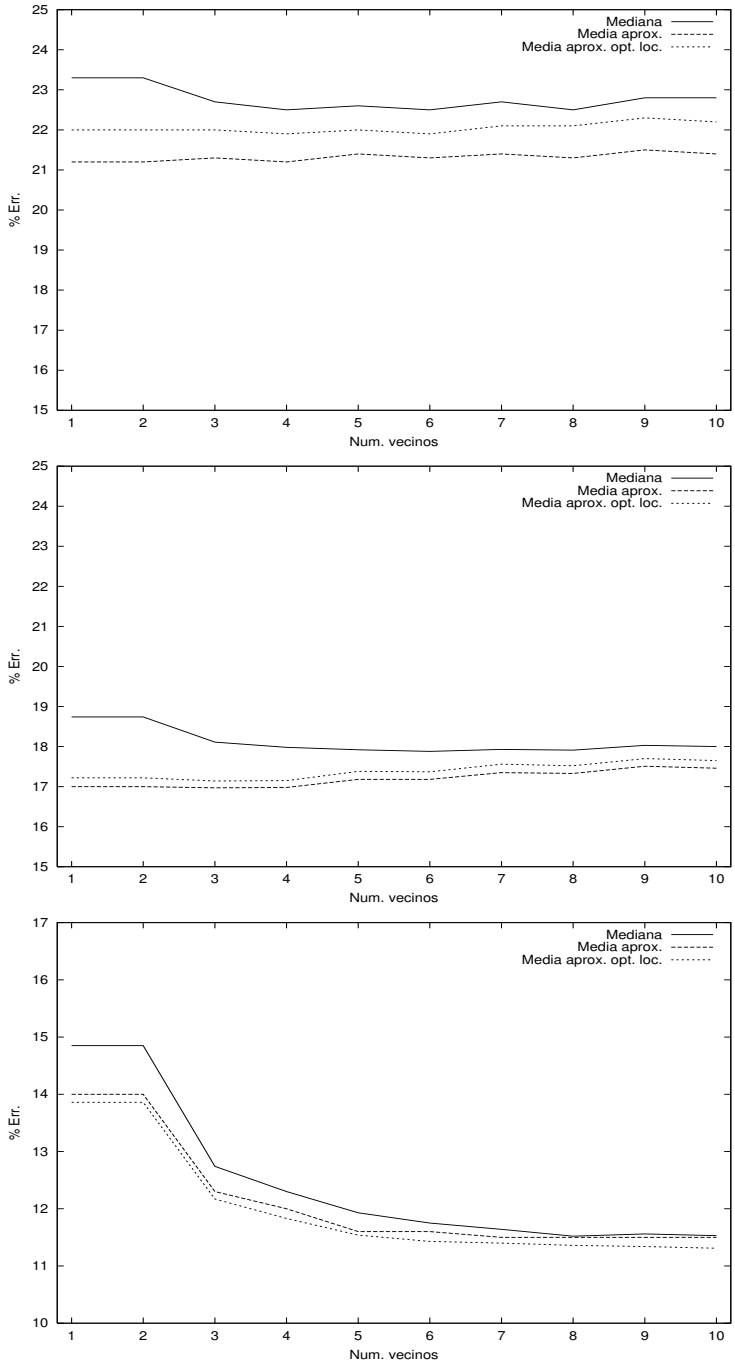


Figura B.1: Resultados de clasificación para el corpus *Cpr* con 10, 100 y 1000 agrupamientos, según el valor del número de vecinos (k) del clasificador, para la cadena mediana y las medias aproximadas con inicialización por cadena mediana y proceso conjunto

atención es la gran disminución de la tasa de error en clasificación con todas las clases de prototipos. Esto era de esperar, pues en general un aumento del número de prototipos provoca una disminución del error de clasificación, y eso es algo que también queda a la vista en los resultados de la Sección 3.3.

En cuanto a los resultados para 1000 agrupamientos, donde el número de cadenas en cada agrupamiento es pequeño (entre 2 y 4 cadenas por agrupamiento), se ve en primer lugar que el incremento del número de prototipos afecta notablemente a los resultados de clasificación, provocando descensos de hasta un 5 % con respecto a los resultados obtenidos para 100 prototipos por clase (nótese el cambio de escala de la gráfica).

Respecto a los resultados alcanzados con el uso de la optimización local, presentan una diferencia bastante significativa respecto a los obtenidos sin usarla en el caso de 10 agrupamientos (en términos absolutos de alrededor de 0'8 puntos, con lo cual la mejora relativa que ofrece usar el método no optimizado es de alrededor de un 3'6 %). Sin embargo, a medida que aumenta el número de agrupamientos esta diferencia se hace menos significativa, llegando incluso en el caso de 1000 agrupamientos a dar mejores resultados la aproximación con optimización local.

Otro hecho apreciable es el irregular comportamiento del error de clasificación a medida que aumenta el valor de k. En principio, para 10 y 100 agrupamientos apenas es apreciable (incluso en 100 agrupamientos se ve un ligero aumento del error en ciertos casos) y sólo para 1000 agrupamientos es notable la caída del error al aumentar k. Así pues, parece haber una relación directa entre la cantidad de prototipos disponibles y la influencia de k. También se pueden achacar las irregularidades descritas a la forma de obtener la codificación y a la forma de hacer los agrupamientos.

Así pues, de estos resultados presentados se puede concluir que usar la cadena media aproximada como prototipo es, en general, mejor que usar la cadena mediana.

Un experimento equivalente al de 1000 agrupamientos con el corpus *Copenhagen* (usando 40 agrupamientos, lo que da un promedio de 2'5 muestras por agrupamiento, cantidad semejante a la de *Cpr* con 1000 agrupamientos) se realizó para validar estas conclusiones. Dichos resultados se presentan en la gráfica de la Figura B.2, donde se aprecia un comportamiento casi idéntico al de la gráfica de *Cpr* con 1000 agrupamientos. De nuevo, las diferencias entre mediana y medias aproximadas van disminuyendo con el incremento de número de vecinos y la aproximación que utiliza optimización local presenta un mejor comportamiento que la no optimizada, aunque las diferencias son pequeñas entre ellas.

De los resultados de la Figura B.1 parece deducirse también que usar optimización local es sensible al tamaño del alfabeto, ya que las diferencias no son significativas cuando éste es reducido (tal y como vimos en los experimentos del corpus *Copenhagen* en la Sección 5.4), pero sí que son destacables cuando el alfabeto es más amplio (aunque dichas diferencias varían según el número de agrupamientos).

Para confirmar este aspecto, se hizo un experimento manteniendo estos 100

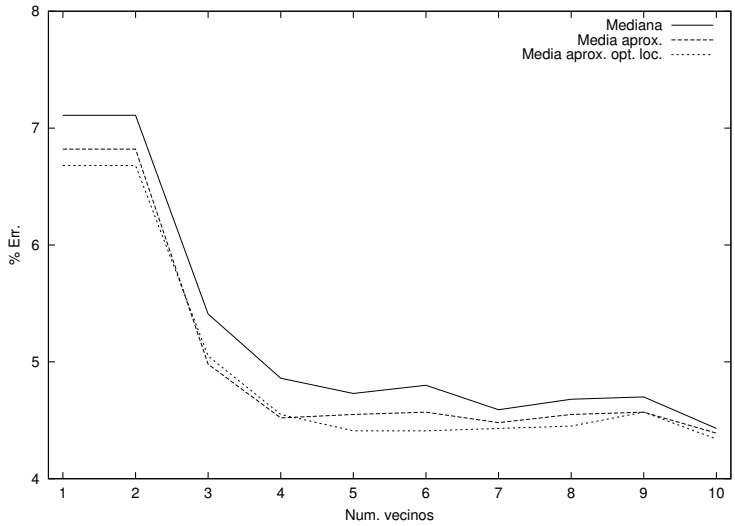


Figura B.2: Resultados de clasificación para el corpus *Copenhagen* con 40 agrupamientos (2'5 muestras de media por agrupamiento), según el valor del número de vecinos del clasificador, para la cadena mediana y las medias aproximadas con inicialización por cadena mediana y proceso conjunto

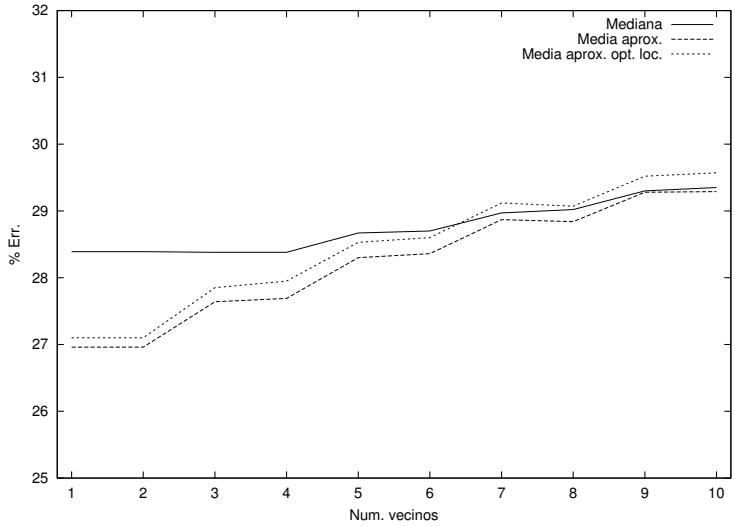


Figura B.3: Resultados de clasificación para el corpus *Cpr* con 100 agrupamientos y con una codificación en 101 símbolos, según el valor del número de vecinos del clasificador, para la cadena mediana y las medias aproximadas con inicialización por cadena mediana y proceso conjunto

agrupamientos y recodificando las cadenas para un alfabeto de 101 símbolos, es decir, desde -50 hasta 50, de manera que -100 y -99 pasaban a ser -50, -98 y -97 a ser $-49 \ldots 97$ y 98 a ser 49 y 99 y 100 a ser 50. Los resultados de clasificación pueden verse en la gráfica de la Figura B.3. Lo primero a destacar es el incremento del error que se produce al usar el nuevo alfabeto y codificación, error que llega a superar al obtenido al usar sólo 10 agrupamientos. También es destacable el irregular comportamiento del clasificador k-NN, pues en este caso el error aumenta con el número de vecinos. Este hecho tan extravagante sólo puede achacarse a la forma de hacer la codificación y a la obtención arbitraria de los agrupamientos llevada a cabo.

Lo más significativo con respecto al comportamiento de la optimización local es el mantenimiento de las diferencias entre la cadena media aproximada no optimizada y la obtenida usando optimización local. De nuevo, las diferencias de los resultados entre ambos tipos son muy bajas (0'3 puntos a nivel absoluto es la mayor diferencia, lo cual vuelve a ser sobre un 1 % de mejora relativa).

De todos estos resultados parece concluirse que la sensibilidad de la optimización local es tanto mayor cuanto más cadenas estén implicadas en la extracción del prototipo, y que la influencia de la talla del alfabeto no es tan alta como en principio podía suponerse. Si para tamaños grandes hemos visto que se aprecia una clara diferencia entre usar o no optimización local (primera gráfica de la Figura B.1), esta diferencia se ha convertido en insustancial con agrupamientos pequeños, llegando incluso a ser beneficioso el uso de la optimización local a nivel de resultados de clasificación (tercera gráfica de la Figura B.1). De esto parece deducirse que el uso de la técnica de optimización local está especialmente indicado para agrupamientos de menor tamaño, algo que está en consonancia con los resultados obtenidos en la Sección 5.4.

Apéndice C

Corpus *abecede*

En este apéndice se muestra la composición de cada una de las clases del corpus *abecede* usado para los experimentos de obtención de la cadena media exacta. Cada columna muestra uno de los agrupamientos (subclases) en los que se ha dividido el total de cadenas de la clase.

C.1. Clase C1

caccd	dbbca	abca	caaad	caaac
aaca	dac	dbcc	acaa	bdac
baba	bdaadda	baccd	addc	aaa
bddb	cbabac	ccdcdd	babacd	ccaa
caba	aaa	baac	dcacba	ccaadc
aacaa	aabaad	cabab	ddca	aacaa
aada	aaacdd	acaaa	baa	aabc
bcb	baca	ababb	caac	aaa
abba	aaba	dabcaa	aaab	acba
dacb	acacac	bdaaca	cab	abbbbaa
cacaa	adabaa	aabcab	ac	caa
cdadb	ccbdb	aaaca	caa	bbbbac
aabcaa	caaba	baa	aaaaad	aadaaac
aaaa	aaa	bbb	dbcc	aac
ddcab	cbcaa	aabbc	aaabba	acba
acbb	dacad	ac	aaab	acbca
baa	ac	adbb	aaaaa	aabab
ddcacb	aadba	aba	dbab	ccaaa
abaacc	aadca	acbaa	caac	aaabab
aacab	bbab	abbba	bbaba	aaaac

C.2. Clase C2

cbccdbb	bbcbbcb	abccab	ccccbcc	dbccccc
cbcccb	cbcad	bcccbc	ccbbcc	cbccbc
cddccb	acbbcccdc	bbbdacb	bccccb	ccccc
cabbcb	cabcaabc	bcddccbc	caadbcbc	cccbca
cbbdbc	cccab	bcddcb	bbcbccab	ccccbcca
ccbcccd	dbcbdaca	ccbddcc	cccbbb	cbccbcb
bccccc	adcbbbdc	cbbcccc	cbccc	cacccc
bbcda	bcccca	cbbccbc	bcbcbb	ccdaa
dcabcc	ccbdcc	cccbbcbb	cbcabb	ccbbcd
baacab	ccaccdcc	bbddcccc	bcbcc	ccccddccb
ddcbcbc	dcddccbc	bbccbbab	dccc	cbcdc
cccacdd	cacbcbc	cbbbacd	bcacc	cdbcccdb
cbccbcbc	bbbacbc	cbccc	ccccbccc	cdbccccbc
ccbbcb	cdbcc	bcccb	baacbc	bcdbc
cdccccd	bccdcbc	cccbbbc	ccaabbcc	cbbbcc
bccdcb	cabccbc	cdbc	bbcccb	ccaccba
ddbcc	ccbc	bcaccb	cbbcccd	badcccd
ccbcbcbb	bcacccc	bccac	bbcccc	bccccdb
bccbdbbc	bbcccbc	dcbccaa	dbdcdc	bcbcccbd
cddcbcb	bdccbc	bbccbaa	cdccccd	bcccdbb

C.3. Clase C3

cabbd	babdd	ab	abbdb	bddca
aab	b	aa	ac	bba
ab	abbababcc	abcbb	baa	d
bb	aacabd	aadacad	bcabbb	dcb
abd	b	ac	ababab	baacbb
dbba	bcbdac	aadb	bb	cbaab
baa	bdbadd	aaad	b	bab
a	ac	cabb	aab	b
ca	bb	bbabbad	bba	cb
ba	bababa	bcbbab	b	aabaaaad
adabc	aabbada	cdccdb	a	d
babb	abbd	ddbba	d	bccbaa
bdabbb	dbaba	c	bbabaac	bbaaaabaa
bca	a	c	ba	a
acda	abaa	abab	aabbba	abd
dba	babab	a	baa	babb
a	a	ba	adac	bacbb
bcaaab	daba	a	aab	abbca
aaddcab	abbbd	aabab	ddb	aabbdac
abbb	bba	bdabb	baba	abab

C.4. Clase C4

dbdddabdb	ddaadaabc	dccdcb	ddddbdadd	bcdddacbd
dcdbddd	ddda	addadc	dddcbd	dbddaba
ddadab	adbdadadaadda	ddabcaabb	ddaaada	dddda
adacaa	abddbddcda	dddddcbbddb	dddaabbdda	ddbabad
dbdddbd	ddbd	aaadbb	bdddbdaadd	ddddaccddd
dcdbdddc	ddddaddddd	bacddbdd	ddabdd	abdbddcbd
dbbddad	dddddcadda	cadddbdc	dddbd	addddab
daadd	dbdbdb	dbbbdddb	ddcdddd	abdaa
baacab	bbadbd	dacaddbadda	ddddbdd	bdaaac
dddbdb	ddbddacddc	cddbcdaabb	ddddb	ddadabccbbdd
ddddacddd	bdcabddbdcc	ddaaacddca	dca	aacd
bddbdacd	bdadacdd	ddddbbdca	dddd	dbdadbddbd
dbbdbddddd	dbadddbdb	ddddb	cddddadaddd	dbadddddbdbda
cdbdddc	dddbd	daada	dacdad	daaa
bddaddcd	abddaabd	cbdbbdbd	bdcacbcadd	badadda
bdbcbab	ddbdbbada	da	ddccdaa	dcbdadcd
dcada	db	bdddbb	ddbbdddc	ddddbbdda
acdddddbdbb	acddddad	adadd	cddddddb	dbbdcbddc
bdbadadadad	bddddbddb	dbcbcbbda	dbcdcdd	bdbbdbdaddd
adabdddd	dddcdab	dabbbdbdd	acdddbdd	bdddbbad

Bibliografía

[1] G. Agam y I. Dinstein. Geometric separation of partially overlapping non-rigid objects applied to automatic chromosome classification. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 19(11):1212–1222, 1997.

[2] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, 1973.

[3] G. Andreu, A. Crespo, y J. M. Valiente. Selecting the toroidal self-organizing features map (tsofm) best organized to object recognition. En *Proceedings of ICNN'97*, volumen 2, páginas 1341–1346, Houston, Texas (USA), Junio 1997. IEEE.

[4] R. Bisiani. Beam search. En *Encyclopedia of Artificial Inteligence*, volumen 2. John Wiley and Sons, segunda edición, 1992.

[5] L. Bobrovsky y J. C. Bezdek. c-means clustering with the l_1 and l_∞ norms. *Transactions on Systems, Man and Cybernetics*, 21(3):545–554, 1991.

[6] H. Bunke y U. Bühler. Applications of approximate string matching to 2d shape recognition. *Pattern Recognition*, 26(12):1797–1812, 1993.

[7] H. Bunke, X. Jiang, K. Abegglen, y A. Kandel. On the weighted mean of a pair of strings. *Pattern Analysis and Applications*, 5(1):23–30, 2002.

[8] A. Carothers y J. Piper. Computer-aided classification of human chromosomes: A review. *Statistics and Computing*, 4(3):161–171, 1994.

[9] F. Casacuberta y M. de Antonio. A greedy algorithm for computing approximate median strings. En *Proceedings of the VII Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, páginas 193–198, Bellaterra, Abril 1997.

[10] C. Chang. Finding prototypes for nearest neighbor classifiers. *IEEE Transactions on Computers*, 23(11):1179–1184, nov 1974.

[11] C. de la Higuera y F. Casacuberta. Topology of strings: Median string is np-complete. *Theoretical Computer Science*, 230:39–48, 2000.

[12] D. Defays. An efficient algorithm for a complete link method. *Computer Journal*, 20:364–366, 1977.

[13] P. Devijver y J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, 1982.

[14] L. Devroye, L. Györfi, y G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[15] R. O. Duda, P. Hart, y D. G. Stork. *Pattern Classification*. John Wiley, 2001.

[16] H. Freeman. Computer processing of line drawing images. *Computer Surveys*, 6:57–98, 1974.

[17] K. S. Fu. *Syntactic pattern recognition and applications*. Prentice Hall, 1982.

[18] G. W. Gates. The reduced nearest neighbour rule. *IEEE Transactions on Information Theory*, 18(3):431–433, mayo 1972.

[19] S. Geva y J. Sitte. Adaptative nearest neighbor pattern classification. *IEEE Transactions on Neural Networks*, 2(2):318–322, 1991.

[20] R. C. González y R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[21] E. Granum y M. G. Thomason. Automatically inferred markov network models for classification of chromosomal band pattern structures. *Cytometry*, 11:26–39, 1990.

[22] E. Granum, M. G. Thomason, y J. Gregor. On the use of automatically inferred markov networks for chromosome analysis. En C. Lundsteen y J. Piper, editores, *Automation of Cytogenetics*, páginas 233–251. Springer Verlag, Berlin, 1989.

[23] J. Gregor y M. G. Thomason. Dynamic programming alingment of sequences representing cyclic patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(2):129–135, Febrero 1993.

[24] J. Gregor y M. G. Thomason. A disagreement count scheme for inference of constrained markov networks. En L. Miclet y C. de la Higuera, editores, *Grammatical Inference: Learning Syntax from Sentences*, volumen 1147 de *Lecture Notes in Computer Science*, páginas 168–178. Springer, 1996.

[25] P. E. Hart. The condensed nearest neighbour rule. *IEEE Transactions on Information Theory*, 14:515–516, mayo 1968.

[26] J. E. Hopcroft y J. D. Ullman. *Introduction to automata theory, languages, and computation*. Addison Wesley, 1979.

[27] E. Horowitz y S. Shani. *Fundamentals of Computers Algorithms*. Computer Science, 1978.

[28] X. D. Huang, Y. Ariki, y M. A. Jack. *Hidden Markov Model for speech recognition*. Edinburg University Press, 1990.

[29] A. K. Jain y R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[30] X. Jiang, K. Abegglen, H. Bunke, y J. Csirik. Dynamic computation of generalized median strings. *Pattern Analysis and Applications*, accepted.

[31] X. Jiang y H. Bunke. Optimal lower bound for generalized median problems in metric space. En Terry Caelli, Adnan Admin, Robert P. W. Duin, Mohamed Kamel, y Dick de Ridder, editores, *Structural, Syntactic and Statistical Pattern Recognition, Joint International Workshops SSPR 2002 and SPR 2002 Proceedings*, Lecture Notes in Artificial Intelligence LNCS/LNAI 2396, páginas 143–151, Windsor, Ontario, Canada, Agosto 2002. Springer-Verlag.

[32] A. Juan. *Optimización de prestaciones en técnicas de aprendizaje no supervisado y su aplicación al reconocimiento de formas*. Tesis doctoral, Departamento de Sistemas Informáticos y Computación, 2000.

[33] A. Juan y E. Vidal. Fast k-means-like clustering in metric spaces. *Pattern Recognition Letters*, 15(1):19–25, 1994.

[34] A. Juan y E. Vidal. Comparison of four initialization techniques for the k-medians clustering algorithm. En *Proc. of Joint IAPR Int. Workshops SSPR 2000 and SPR 2000*, volumen 1876 de *Lecture Notes in Computer Science*, páginas 842–852, Alacant (España), Septiembre 2000. Springer-Verlag.

[35] O. Kariv y S. L. Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Math.*, 37(3):539–560, 1979.

[36] I. Katsavounidis. A new initialization technique for generalized lloyd iteration. *IEEE Signal Processing Letters*, 1(10):144–146, Octubre 1994.

[37] Y. Kitazume, E. Ohira, y T. Endo. Lsi implementation of a pattern matching algorithm for speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33(1):1–5, feb 1985.

[38] T. Kohonen. Median strings. *Pattern Recognition Letters*, 3:309–313, 1985.

[39] T. Kohonen. The self-organizing map. *Proc. of the IEEE*, 78(9):1464–1480, 1990.

[40] L. I. Kuncheva y J. C. Bezdek. Nearest prototype classification: Clustering, genetic algorithms, or random search? *Transactions on Systems, Man y Cybernetics*, 28(1):160–164, feb 1998.

[41] J. R. Lacadena. *Citogenética*. Editorial Complutense, 1996.

[42] G. N. Lance y W. T. Williams. A general theory of classificatory sorting strategies. 1. Hierarchical systems. *The Computer Journal*, 9:373–380, 1967.

[43] P. L'Ecuyer. Good parameters and implementations for combined multiple recursive random number generators. *Operation Researchs*, 47(1):159–164, 1999.

[44] C. Lundsteen, J. Philip, y E. Granum. Quantitative analysis of 6985 digitized trypsin g-banded human metaphase chromosomes. *Clinical Genetics*, 18:355–370, 1980.

[45] M. Maes. On a cyclic string-to-string correction problem. *Information Processing Letters*, 35(2):73–78, junio 1990.

[46] C. D. Martínez Hinarejos, A. Juan, y F. Casacuberta. Use of median string for classification. En *Proceedings of the 15th Int. Conf. on Pattern Recognition (ICPR 2000)*, volumen 2, páginas 907–910, Barcelona, sep 2000.

[47] C. D. Martínez Hinarejos, A. Juan, y F. Casacuberta. Improving classification using median string and NN rules. En *Proc. of the IX Spanish Symposium on Pattern Recognition and Image Analysis*, volumen II, páginas 391–395, Benicàssim, mayo 2001.

[48] C. D. Martínez Hinarejos, A. Juan, y F. Casacuberta. Median strings for k-nearest neighbour classification. *Pattern Recognition Letters*, 24(1-3):173–181, 2003.

[49] C. D. Martínez Hinarejos, A. Juan, y F. Casacuberta. Generalized k-medians clustering for strings. En Francisco José Perales, Aurélio J. C. Campilho, Nicolás Pérez de la Blanca, y Alberto Sanfeliu, editores, *Pattern Recognition and Image Analysis, First Iberian Conference IbPRIA 2003 Proceedings*, Lecture Notes in Computer Science LNCS 2652, páginas 502–509, Port d'Andratx, Mallorca, España, Junio 2003. Springer-Verlag.

[50] C. D. Martínez Hinarejos, A. Juan, F. Casacuberta, y R. Mollineda. Reducing the computational cost of computing approximated median strings. En Terry Caelli, Adnan Admin, Robert P. W. Duin, Mohamed Kamel, y Dick de Ridder, editores, *Structural, Syntactic and Statistical Pattern Recognition, Joint International Workshops SSPR 2002 and SPR 2002 Proceedings*, Lecture Notes in Artificial Intelligence LNCS/LNAI 2396, páginas 47–55, Windsor, Ontario, Canadá, Agosto 2002. Springer-Verlag.

[51] J. Di Martino. Dynamic time warping algorithms for isalated and connected word recognition. En R. De Mori y Y. Suen, editores, *New Systems and Architectures for Automatic Speech Recognition and Synthesis*. Springer Verlag, Berlin, 1985.

[52] A. Marzal y S. Barrachina. Speeding up the computation of the edit distance for cyclic strings. En *Proceedings of the 15th Int. Conf. on Pattern Recognition (ICPR 2000)*, volumen 2, páginas 895–898, Barcelona, sep 2000.

[53] A. Marzal y E. Vidal. Computation of normalized edit distance and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(9):926–932, sep 1993.

[54] P. B. Mirchandani y R. L. Francis, editores. *Discrete Location Theory*. Wiley, 1990.

[55] R. Mollineda. *Técnicas de Agrupamientos Jerárquicos para la Selección de Prototipos y Clasificación basadas en Distancias. Uso en Cadenas Cíclicas*. Tesis doctoral, Departamento de Sistemas Informáticos y Computación, 2001.

[56] R. A. Mollineda, E. Vidal, y F. Casacuberta. Efficient techniques for a very accurate measurement of dissimilarities between cyclic patterns. En *Proc. of Joint IAPR Int. Workshops SSPR 2000 and SPR 2000*, volumen 1876 de *Lecture Notes in Computer Science*, páginas 337–346, Alacant (España), Septiembre 2000. Springer-Verlag.

[57] R. A. Mollineda, E. Vidal, y F. Casacuberta. A windowed version of the Bunke-Bühler algorithm to better approximate dissimilarities between cyclic patterns. En *Proceedings of the 5th. Iberoamerican Symposium on Pattern Recognition*, páginas 311–321, Lisboa (Portugal), Septiembre 2000. Portuguese Association of Pattern Recognition.

[58] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *Computer Journal*, 26:354–359, 1983.

[59] M. Nadler y E. P. Smith. *Pattern Recognition Engineering*. John Wiley, 1993.

[60] G. L. Nemhauser y L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1998.

[61] C. H. Papadimitriou y K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, 1982.

[62] J. C. Pérez y E. Vidal. Constructive design of LVQ and DSM classifiers. En J. Mira, J. Cabestany, y A. Prieto, editores, *New Trends in Neural Computation*, Lecture Notes in Computer Science 686, páginas 334–339. Springer-Verlag, 1992.

[63] J. Piper. Variability and bias in experimentally measured classifier error rates. *Pattern Recognition Letters*, 13:685–692, 1992.

[64] J. Piper y E. Granum. On fully automatic feature measurement for banded chromosome classification. *Cytometry*, 10:1–14, 1989.

[65] L. Rabiner y B. Juang. *Fundamentals of speech recognition.* Prentice Hall, 1993.

[66] G. Ritter y K. Gaggermeier. Automatic classification of chromosomes by means of quadratically asymmetric statistical distributions. *Pattern Recognition*, 32:997–1008, 1999.

[67] G. Ritter y M. T. Gallegos. Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters*, 18:525–539, 1997.

[68] G. Ritter, M. T. Gallegos, y K. Gaggermeier. Automatic context-sensitive karyotyping of human chromosomes based on elliptically symmetric statistical distributions. *Pattern Recognition*, 28(6):823–831, 1995.

[69] G. Ritter y G. Scheib. Using dominant points and variants for profile extraction from chromosomes. *Pattern Recognition*, 34:923–938, 2001.

[70] F. J. Rohlf. Hierarchical clustering using the minimum spanning tree. *Computer Journal*, 16:93–95, 1973.

[71] G. Sánchez, J. Lladós, y K. Tombre. A mean string algorithm to compute the average among a set of 2d shapes. *Pattern Recognition Letters*, 23:203–213, 2002.

[72] D. Sankoff y J. B. Kruskal. *Time Warps, String Edits and Macromolecules: the Theory and Practice of Sequence Comparison.* Addison Wesley, 1983.

[73] R. Sibson. Slink: an optimally efficient algorithm for a complete link method. *Computer Journal*, 16:30–34, 1973.

[74] D. B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. En *Proceedings of the 11th International Conference on Machine Learning*, páginas 293–301, 1994.

[75] S. Theodoridis y K. Koutroumbas. *Pattern Recognition.* Academic Press, 1999.

[76] J. D. Thompson, D. G. Higgins, y T. J. Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.

[77] E. Vidal y M. J. Castro. Classification of banded chromosomes using error-correcting grammatical inference (ECGI) and multilayer perceptron (MLP). En *Proceedings of the VII Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, volumen 1, páginas 31–36, Bellaterra (España), 1997.

[78] E. Vidal, M. J. Castro, y J. A. Sánchez. Classification of banded chromosomes. Informe técnico, DSIC, Universidad Politécnica de Valencia, España, 1997.

[79] E. Vidal, A. Marzal, y P. Aibar. Fast computation of normalized edit distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):899–902, 1995.

[80] R. Wagner y M. Fisher. The string-to-string correction problem. *Journal of the ACM*, 21:168–178, 1974.

[81] J. H. Ward. Hierarchical grouping to optimise objective functions. *Journal of the American Statistical Association*, 58:236–244, 1963.

[82] A. Webb. *Statistical pattern recognition.* Edward Arnold, 1999.

[83] D. L. Wilson. Asymptotic properties of nearest neighbour rules using edited data. *Transactions on Systems, Man and Cybernetics*, (2):408–421, 1972.

[84] H. Yan. Prototype optimization for nearest neighbor classifiers using a two-layer perceptron. *Pattern Recognition*, 26(2):317–324, 1993.