

Mini-curso sobre procesos gaussianos y optimización bayesiana

Antonio Sala
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Mini-curso en vídeo

VERSIÓN DEL DOCUMENTO:

Fecha y hora:	29/08/2024 [12:57]
Núm. objetos de aprendizaje:	35
Tiempo total lecciones en video:	07:22:28
	[propio (35): 07:22:28, terceros (0): 00:00:00]

©2024, Antonio Sala Piqueras, DISA-UPV.

Se reservan todos los derechos de uso, copia, y propiedad intelectual de los materiales de mi autoría en este documento y los materiales audiovisuales/PDF/código enlazados desde el mismo, tanto respecto a su uso *verbatim* o como base de obras derivadas. La copia o redistribución de contenidos de terceros deberá respetar los derechos de uso que sus autores hayan establecido.

Presentación

El presente PDF presenta una colección de vídeos que abordan el modelado con procesos gaussianos y la estimación con dichos modelos, y su aplicación a optimización.

Se han seleccionado un subconjunto de vídeos de la colección completa, pero si estás interesado en profundizar más, quizás te interesen más vídeos del Apéndice B (rev. estadística), Capítulo 6 (procesos estocásticos), y Sección 19.5 (Métodos Kernel de regresión) de la colección completa <http://personales.upv.es/asala/DocenciaOnline/Cursos/Apuntes.html>; incluso puede que, involuntariamente, me haya saltado algún concepto importante y que debas revisarlo en la colección completa para poder entender alguno de los vídeos de esta selección.

La lista de vídeos o las descripciones de los mismos o materiales pueden haber sufrido alguna actualización. Si este PDF es “antiguo”, reemplazalo por la última versión: <http://personales.upv.es/asala/DocenciaOnline/Cursos/GPcurso.pdf>

* Por defectos en el software (realizado por mí) que procesa una base de datos asociada para generar este PDF de forma automática, los enlaces a vídeos de la colección completa no incluidos en esta selección salen con [??]. Haciendo click en el acrónimo al lado del número de vídeo se accede a la página web del vídeo en cuestión, donde los enlaces sí están correctamente insertados.

Capítulo 1

Estadística multivariable y procesos estocásticos: una introducción rápida

1.1 Preliminares/revisión de conceptos de estadística

[1: vard]	Variables aleatorias reales: función de densidad, función de distribución acumulativa, vbles. Mixtas Materiales: [vard.pdf]	***  13:34 [YouTube ▶]
[2: mvmp]	Parámetros (media, varianza, momentos) de una distribución de probabilidad Materiales: [ParamsDistribPr.pdf]	***  11:05 [YouTube ▶]
[3: va2d]	Variables aleatorias bidimensionales: distrib. marginal, covarianza Materiales: [vblesaleat2D.pdf]	***  12:14 [YouTube ▶]
[4: vamult]	Variables aleatorias multidimensionales (caso general): matriz de varianzas-covarianzas Materiales: [vblesMultidimensionales.pdf]	***  08:44 [YouTube ▶]
[5: vamdm]	Variables aleatorias multidimensionales: ejemplo Matlab Materiales: [CÓD.: VariablesNDimensionalesExample.mlx] [PDF]	***  13:34 [YouTube ▶]
[6: dadonorm2d]	Distribución normal 2-dimensional: simulación de muestras (covarianza dada) en Matlab Materiales: [CÓD.: TirarDadoNormal.mlx] [PDF]	**  09:16 [YouTube ▶]
[7: preli2]	Predicción lineal óptima: fórmulas basadas en matriz de varianzas-covarianzas Materiales: [FormulasPredOptimaLineal.pdf]	***  13:55 [YouTube ▶]
[8: prelim]	Predicción lineal óptima: ejemplo Matlab Materiales: [CÓD.: MejorPredLineal.mlx] [PDF]	**  10:11 [YouTube ▶]

1.2 Procesos estocásticos: introducción

[9: estoc1]	Procesos estocásticos (funciones aleatorias): motivación, definición y ejemplos para estudiantes ingeniería	**  15:47
	Materiales: [estocas1.pdf]	[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo plantea el concepto de “proceso estocástico”, que se utiliza en estadística, como formalización de lo que se entendería en ingeniería como una “señal con ruido”. Se discute su definición fundamental y ejemplos en procesamiento de señal industrial, audio-vídeo, vibraciones, así como los objetivos del estudio de los mismos. El objetivo del procesamiento de señal son los procesos estocásticos en el tiempo (series temporales) que los siguientes vídeos analizan; de todas formas, algunas de las ideas son generalizables de forma sencilla a procesos de dimensión mayor asociados a funciones aleatorias sobre un dominio espacial bidimensional (análisis de ruido en imágenes, por ejemplo) o incluso mixto (análisis de señales de vídeo); véase, por ejemplo, el caso bidimensional en el vídeo [18].

[10: estoct]	Señales con ruido: procesos estocásticos en el tiempo (series temporales), definiciones básicas	***  11:00
	Materiales: [estocas2.pdf]	[YouTube ▶]

Este vídeo presenta definiciones particulares a lo que se entiende como “serie temporal” (proceso estocástico indexado sobre el tiempo continuo o discreto). Introduce el concepto de realización. Las propiedades de la señal, como medias y varianzas, serían funciones del tiempo. También se define la covarianza o correlación entre variables aleatorias correspondientes a dos instantes de tiempo: la correlación estadística entre señales en distintos instantes permitirá extender el concepto de predicción lineal óptima del vídeo [7] a series temporales, que se explica más adelante en otros vídeos de esta colección. El vídeo finaliza enumerando los posibles objetivos del análisis de este tipo de señales (identificación, predicción, control predictivo, filtrado, suavizado, interpolación) y resumiendo las ideas principales en sus conclusiones.

Nota: los procesos estocásticos pueden venir bien con matrices de varianzas-covarianzas, y correlaciones “caídas del cielo” (o computadas experimentalmente) con lo que ya se puede aplicar las fórmulas de predicción, o bien ser generados a partir de un modelo físico de un sistema con entradas tipo ruido (en tiempo discreto –ecuaciones en diferencias estocásticas, vídeo [??]–, o en tiempo continuo –ecuaciones diferenciales estocásticas, vídeos [??] y [??], entre otros–). No es objetivo de este vídeo entrar en el detalle de “de donde vienen”, sino de definir “qué son”.

Aunque esté en la sección de optimización bayesiana más adelante, podría interesarte en este momento el revisar otra visión sobre cómo los procesos estocásticos se podrían usar en ciertos problemas de optimización, que describe de forma introductoria sin fórmulas el vídeo [BOmot1EN(15:40)].

Algunos conceptos teóricos de base

[11: estoce]	Procesos estocásticos estacionarios	****  13:12
	Materiales: [estocs3stac.pdf]	[YouTube ▶]

Este vídeo explica el concepto de estacionariedad en series temporales, que generaliza el concepto de “equilibrio” (valor constante) al concepto de “función de densidad conjunta que no depende del tiempo” (con más precisión, independiente de desplazamientos temporales) para definir los procesos estacionarios en sentido *estricto*, y una versión más relajada

del mismo sólo en términos de media, varianza y covarianzas constantes (estacionariedad en sentido *amplio* o *débil*) que resulta de gran utilidad en la práctica ya que las operaciones lineales sobre señales en muchos casos sólo requieren del cálculo de dichas medias, varianzas y covarianzas por lo que si son constantes los cálculos resultan más sencillos.

[12: psd] [Densidad espectral de potencia en señales aleatorias estacionarias](#)

Materiales: [estocs5PSD.pdf]

*****  17:24

[YouTube ▶]

En un proceso estacionario, la señal temporal $x(t)$ es aleatoria, pero la media, varianza y covarianza entre instantes de tiempo, si se repitiera dicho experimento infinitas veces, sí serían constantes. Obviamente, si $x(t)$ es aleatorio, también lo será su contenido en frecuencia. Sin embargo, en muchas señales físicas se “ven” oscilaciones (correlación entre los valores en un instante y los valores cierto tiempo “período” después). La relación entre la “correlación entre dos instantes separados” y “contenido frecuencial” es lo que explora este vídeo.

En concreto, este vídeo presenta el concepto de “*densidad espectral de potencia*” (power spectral density, PSD, en la literatura en lengua inglesa).

En los primeros cinco minutos de vídeo, se revisan los conceptos de proceso estacionario y de transformada de Fourier. La segunda mitad del vídeo esboza la demostración del teorema de Wiener–Khinchin–Einstein, que relaciona la respuesta “temporal” (covarianza entre diferentes instantes de tiempo) con la respuesta “frecuencial” (densidad espectral de potencia).

La identidad de Parseval, obtenida de la fórmula de la transformada inversa de Fourier, indica que la varianza de una señal es la integral de su PSD sobre el eje de frecuencias. Ello posibilita la interpretación de la PSD como “contribución de los componentes de frecuencia f a la varianza total”, o a la integral en un intervalo de frecuencias como “varianza en dicho rango de frecuencia”. La dualidad entre la interpretación estadística y la interpretación frecuencial permite entender determinados problemas de estimación en series temporales como problemas de *filtrado* en el dominio de la frecuencia. Se aconseja visualizar el vídeo [17] para ver ejemplos Matlab de pares autocovarianza/densidad espectral y su uso en filtrado.

[13: dadogauss] [Generación de trayectorias de un proceso estocástico gaussiano de autocovarianza dada](#)

Materiales: [[CÓD.:](#) TirarDadoProcesoGaussiano.mlx] [[PDF](#)]

****  12:46

[YouTube ▶]

En este vídeo se generan trayectorias de un proceso estocástico gaussiano con función de covarianza dada $\kappa(x_1, x_2)$. Una vez se tiene el rango de x a “simular” (discretizado a N puntos, para que el número de variables aleatorias sea finito, claro) lo único que hay que hacer es aplicar las ideas basadas en diagonalización del vídeo [6] cuya visualización previa a éste se aconseja a la matriz $N \times N$ obtenida calculando las covarianzas dos a dos. Como alternativa computacional más rápida, también se calcula aquí la factorización de Cholesky de la matriz de varianzas.

En este vídeo se ilustran realizaciones de el régimen estacionario de un filtro de primer orden y un filtro de segundo orden, usando la dualidad entre respuesta en frecuencia (power spectral density) y función de autocovarianza: la respuesta en frecuencia es la transformada de Fourier de la autocovarianza (ver vídeo [12] con detalle y demostración asociada); esto se hace por comprender mejor dicha interpretación, pero realmente si no se hace uno preguntas sobre el origen de $\kappa(x_1, x_2)$ sólo hay que directamente aplicar los resultados del vídeo [6] antes mencionado.

La parte final del vídeo comprueba empíricamente que la varianza y covarianza entre muestras de las realizaciones generadas coincide, efectivamente, con aquélla especificada por la función κ (cuando el número de muestras tiende a infinito).

[14: gpsambEN] [Sampling a Gaussian process \(realizations\), Matlab example](#)***  13:57**Materiales:** [[CÓD.:](#) GPSampletests2ENG.mlx] [[PDF](#)]

[YouTube ►]

This video discusses how to compute ‘realizations’ (random functions $f(x)$) of a Gaussian stochastic process with given mean function $\bar{f}(x)$ and covariance kernel $k(x_1, x_2)$. As the process is ‘continuous time’, samples of it will only be generated in a finite set of test points.

The video discusses the code to generate the covariance matrix in said set of test points, the meaning of the chosen stationary covariance kernel (quadratic exponential, but it can be any other), the band-type structure of the covariance matrix and the use of `mvnrnd` to generate the process realizations. The internal detail of the `mvnrnd` command requires carrying out diagonalization or Cholesky decomposition of the covariance matrix, but those implementation details are out of the scope of this video.

To better understand this type of processes, the correlation abscissa distance parameter is changed in the covariance function, also, in an additional example, white measurement noise is added (increasing the diagonal of the covariance), etc.

The video [15], a continuation of this one, will discuss the same problem (generating realizations) when observations are available at some ‘measurement’ points of the value of the process.

[15: gpsambpoEN] [Sampling a Gaussian process with some observations \(sampling the posterior\), Matlab example](#)**** 

14:55

Materiales: [[CÓD.:](#) GPSampletests2ENG.mlx] [[PDF](#)]

[YouTube ►]

This video discusses how to compute realizations of a stochastic process when you have observations of its value at a set of ‘measurement’ points. For example, the stochastic process may be the speed of each car at different points on a highway, and you want to ‘interpolate’ for a specific car the most likely trajectories knowing that, in two positions where there was a radar, its recorded speeds were 110 and 80 km/h.

The first part of the video reviews previous concepts and best linear prediction, and discusses the code to generate a posterior mean and covariance matrix on a set of test points.

The second part of the video plots the confidence intervals of that posterior and generates various realizations (random functions) to illustrate the concept, using `mvnrnd`. The development is quite parallel to the video [14], whose visualization is recommended prior to this one.

The final part of the video presents simulations of the results of changing the correlation distance in the covariance kernel.

Estimación y predicción

[16: krigt] [Kriging: estimación \(interpolación/extrapolación\) en series temporales con autocovarianza conocida](#)****  08:44**Materiales:** [GaussianStatisticInterp.pdf]

[YouTube ►]

En este vídeo se plantea cómo, a partir de la función de autocovarianza $\kappa(t_1, t_2)$ de un proceso estocástico temporal, construir la mejor predicción de $y(t)$ para t arbitrario a partir de muestras $\{y_1, y_2, \dots, y_N\}$ tomadas en instantes $T = \{t_1, t_2, \dots, t_N\}$. Se utiliza la fórmula de predicción lineal óptima (vídeo [7]).

De hecho, la idea es una generalización del código preliminar en el vídeo [??] que puede servir de ejemplo Matlab introductorio a lo que aquí se plantea. Otro ejemplo se desarrolla en el vídeo [17], continuación de éste.

En el caso gaussiano, la fórmula de la varianza del error de predicción permitiría obtener intervalos de confianza a partir de las fórmulas de la distribución normal.

Se aborda el caso estacionario y su interpretación frecuencial posibilitada por las ideas de densidad espectral de potencia (vídeo [12]), y la incorporación de posible ruido de medida en las muestras no correlado con $y(t)$ ni con los otros instantes.

Nota: el vídeo [??] estudia un problema similar con modelos en representación interna.

[17: krigtm]

Interpolación/extrapolación en series temporales estacionarias a partir de número finito de muestras: ejemplo Matlab (Kriging)

Materiales: [[CÓD.:](#) GaussProcFiniteSamplesFilter.mlx] [[PDF](#)]

*** 10:29
[YouTube ▶]

Este vídeo presenta un ejemplo Matlab de la metodología de interpolación/extrapolación de una señal generada por un proceso estocástico estacionario de autocovarianza (o densidad espectral de potencia, en caso estacionario, vídeo [12]) dada, cuya teoría se discutió en el anterior vídeo [16].

En concreto, para un cierto conjunto de muestras tomadas a intervalos irregulares (refinamientos de estas ideas pueden utilizarse en lo que se denomina control de sistemas con *muestreo no convencional*) se ilustran los resultados en media y en intervalo de confianza (proceso gaussiano) suponiendo que la dinámica que ha generado la señal ha sido un ruido blanco (densidad de potencia 1 a todas las frecuencias) excitando a un proceso lineal de orden 1, a un filtro Butterworth de segundo orden y a un filtro resonante de segundo orden. Si se trabajara con funciones de transferencia discretas $G(z)$ y transformada discreta de Fourier podrían hacerse desarrollos similares en tiempo discreto, que se dejan al lector (obviamente, las muestras tendrían que estar tomadas en múltiplos del período básico de muestreo para el que se haya descrito la función de transferencia).

Un caso de estimación en series temporales suponiendo que el proceso estocástico subyacente es no estacionario se aborda en el vídeo [??].

[18: krig2dm]

Interpolación en procesos gaussianos 2D (Kriging, Kernel Regression): ejemplo Matlab

Materiales: [[CÓD.:](#) regresionGaussian2inputs.mlx] [[PDF](#)]

**** 09:25
[YouTube ▶]

Las fórmulas de estimación del vídeo [17] pueden aplicarse a variables índice t que sean multidimensionales, $t \in \mathbb{R}^2$ en vez de $t \in \mathbb{R}$ por ejemplo. Ello permite extender la idea de “filtrado en el tiempo” a “interpolación” en coordenadas espaciales (o espaciales y temporales en procesos, por ejemplo, gobernados por ecuaciones en derivadas parciales). Este vídeo presenta un ejemplo de interpolación 2D de una función a partir de unos puntos donde la “suavidad” de la función viene definida por la función de covarianza (*kernel*) especificada por el diseñador. Se usan las ecuaciones de mejor predicción y intervalo de confianza (proceso gaussiano) y se explora su interpretación en problemas de regresión/interpolación entendidos como filtrado estadístico. Las ideas de este vídeo están muy relacionadas con los “métodos Kernel” de regresión discutidos en otros materiales (por ejemplo, los vídeos [??], [??]); de hecho, el vídeo [??] sobre métodos denominados “regresión Kernel” enfatiza la relación prácticamente directa de dicha regresión con los conceptos de este vídeo.

[19: gaussp]

Regresión Kernel: interpretación estadística/filtro de Wiener (Matlab)

Materiales: [[CÓD.:](#) KernelRegresGaussProc.mlx] [[PDF](#)]

Este es un fichero de Matlab que ejemplifica cómo generar una función $\kappa(x, y)$ para Kernel regression en el sentido estadístico descrito en el material [??].

Si se ejecuta la prueba “0”, se puede (comentar/descomentar) ver el efecto de usar un filtro de primer orden $a/(s+b)$ o un filtro de Butterworth de segundo orden $a/(s^2 + \sqrt{2} \cdot \omega s + \omega^2)$ para generar sus power-spectral-density y autocorrelación asociadas, siendo la última la propuesta para κ . La relación con la fórmula de reconstrucción de señales muestreadas de ancho de banda finito también es explorada (teorema de Shannon-Nyquist) haciendo **prueba=1**. En resumen, back to the basics: los métodos Kernel pueden ser reinterpretados como la versión multidimensional de resultados clásicos de “filtro de Wiener”, “Shannon”, etc. en procesamiento de señal.

Capítulo 2

Optimización bayesiana

La versión en español está pendiente de grabación/edición, porque priorizo contenidos de las asignaturas grado/máster que imparto, mis disculpas.

2.1 Motivación y esquema básico de la metodología

[20: BOmot1EN] Bayesian Optimization motivation (1/4): problem statement and model classes
Materiales: [BOIntroTheoENG.pdf]

******  **15:40**
[YouTube ▶]

This introductory video raises the need for statistical methods in optimization in some classes of applications (say, experimental optimization, data with additive random noise, etc.) where there is no model with ‘perfectly known equations’ to optimize (i.e., under the realm of deterministic optimization).

A set of algorithms that use statistical models in optimization are usually grouped under the name *Bayesian optimization* in literature. This video states the problem, raises the concept of ‘random function’ and the need for a covariance kernel to prove, among other things, continuity of the function and, in general, introduces the idea of stochastic processes and Gaussian processes as the basic tool that most works on Bayesian optimization exploit.

Other descriptions and applications of stochastic processes are covered in the introductory video [??]; forthcoming videos will develop the particular Bayesian optimization problem in greater depth, starting with [21].

[21: BOmot2EN] Bayesian Optimization motivation (2/4): methodology outline
Materiales: [BOIntroTheoENG.pdf]

*******  **11:20**
[YouTube ▶]

This video continues with the introduction and motivation to Bayesian Optimization problems which was started in video [20].

In here, we review said video in the first four minutes, and then we discuss the generic goal of Bayesian Optimization: it’s actually a type of ‘experiment design’ trying to choose samples that have a high likelihood of giving me a value close to the optimal (exploitation) or, well, maybe we wish to sample currently likely sub-optimal ones in exchange for them giving a lot of information on where the actual optimum will be for future samples (exploration).

So, BO algorithms end up comprising the following steps:

- Set up a prior statistical model

- Acquire samples (this may be a costly procedure in some relevant application cases)
- Compute a posterior (conditional to the measurements)
- Carry out a statistical analysis on the posterior, to decide next sample, and go to step 2, unless some termination criteria (sample budget, quality of the last sample, lack of progress) is met.

For brevity, a quick outline of the details of the third and fourth steps will be deferred to video [22], and application cases and concluding remarks in video [23] will conclude this brief motivation and presentation of the Bayesian optimization problem. A quick numerical example of the methodology appears in video [24], but you need to watch [22] first to better understand it.

[22: BOMot3EN]

10:49

Bayesian Optimization motivation (3/4): computation of posterior and decision (acquisition) on next sample (outline)

Materiales: [BOIntroTheoENG.pdf]



[YouTube ▶]

This video continues with the introduction and motivation to Bayesian Optimization problems which was started in video [20].

In video [21] we concluded that BO algorithms end up comprising the following steps:

- Set up a prior statistical model
- Acquire samples (this may be a costly procedure in some relevant application cases)
- Compute a posterior (conditional to the measurements)
- Carry out a statistical analysis on the posterior to decide next sample, and go to step 2, unless some termination criteria is met.

This video expands on the third step of the methodology, which is carried out in a Gaussian process with the conditional formulae from multivariate normal distributions.

The last step is also outlined... next sample may be selected in order to optimize the expected value (EV), probability of improving (PI), expected improvement (EI), we may be risky and choose a lower confidence bound (we will not likely achieve it, but we would obtain a very good sample if we did), or optimize the information gain by sampling (entropy search). A very brief description of each of these options is presented, just to get a glimpse of the main ideas, but a detailed analysis is the topic of other materials that analyse BO in more depth, see videos [26] and [??].

A quick numerical example of the methodology appears in video [24]. Application cases and concluding remarks in video [23] will conclude this brief presentation of the Bayesian optimization procedures.

[23: BOMot4EN]

Bayesian Optimization motivation (4/4): recommended application domains, remarks

Materiales: [BOIntroTheoENG.pdf]

**



19:33

[YouTube ▶]

After a quick review of prior material, this video discusses the application domains where Bayesian optimization IS recommended (model-free experimental optimization, configuration of long computations in neural networks, fluids, . . .), as well as some circumstances in which it is NOT recommended (simple functions, high-dimensional problems). If you did not watch videos [20], [21] y [22], I recommend doing it now to get a broader view.

The video concludes with a couple of remarks on, first, the fact that determinism is a limit case of statistics with uncertainty size tending to zero, so there are quite a few deterministic optimization problems that admit a ‘statistical interpretation’ (say, Least

squares, assuming normally distributed random measurement noise); there are also semi-parametric Bayesian optimization options.

The second remark pinpoints the fact that gradient-based methods may get stuck on local minima; hence, some algorithms (directed random search) use 'rolling the dice' steps to try to explore and get the global optimum; however, these issues do not change the fact that the underlying problem is deterministic and no uncertainty is assumed to be present in the objective function.

A summary and conclusions reviewing the ideas on this video and the preceding ones in the series that started with video [20] end the video and the 4-video set that motivated and briefly outlined the main features of Bayesian optimization.

[24: boloop1EN] Bayesian Optimization loop: a quick example of the methodology

Materiales: [[CÓD.:](#) BOIntroLOOPSimpleENG.mlx] [[PDF](#)]

***  12:29

[YouTube ▶]

This video presents an example of the Bayesian Optimization methodology which we outlined in videos [21] and [22]. Code issues are only sparingly mentioned, as later videos will delve into the details and add some refinements.

The video focuses on understanding the Bayesian optimization loop (acquire a sample, update posterior, evaluate acquisition function for next sample), via a simple example using 'lower confidence bound' as said acquisition function heuristic.

2.2 Detalles de código y de la interpretación de funciones de adquisición

[25: boinopEN] Bayesian optimization: implicit information about the optimum in a Gaussian process

Materiales: [[CÓD.:](#) BOIntroOptimalSample.mlx] [[PDF](#)]

***  18:53

[YouTube ▶]

This video plots the (approximate) distribution of global minima of the posterior of a function modeled as a Gaussian process of which three samples have been observed.

First, the video reviews how to obtain said posterior. Next, we review how to obtain realizations of it, and the code to repeat them and obtain the minimum of each one (this is the basic of the "Thompson sampling" heuristic in Bayesian optimization). With this code, six thousand minimums are obtained and three histograms are plotted:

- Marginal of x^* , probability density that in a given x^* there is a global minimum of some realization;
- Marginal of y^* , probability density that a certain y^* is the optimal value (global minimum) that achieves some realization;
- "joint" histogram approximating the joint density function over (x^*, y^*) representing the probability that some realization has the global minimum at x^* and its value is precisely y^* .

This is the implicitly available information about the location of the optimum. The goal of Bayesian optimization is to refine that information through sampling (explore) and/or to hit the optimum of my unknown underlying function as quickly as possible (exploit).

[26: boPIEN] Bayesian Optimization: probability of improvement, example

Materiales: [[CÓD.:](#) BOIntroAcquisitionShortEN.mlx] [[PDF](#)]

****  20:45

[YouTube ▶]

This video will discuss the meaning of some acquisition functions in Bayesian Optimization by means of an example. In particular, this first video will review the basic methodology, and it will set an example up with some mean, covariance kernel and dataset to work on it. If you are familiar with these concepts (if you already watched [21], [22] and [25]), you may skip to [10:30] when basic review and example setup have been already explained.

Then, the meaning of the ‘probability of improvement’ heuristic (PI) for BO acquisition function is discussed. In particular, it is shown to lie on the ‘conservative side’ if good samples (below the mean) are present: if I want to improve with almost certainty, I must descend a ‘little step’ in the downward direction (sort of gradient search), and also do not drift too far away from good samples to avoid uncertainty piling up. I will improve with large probability but, possibly, I will not jump quickly to the global minimum.

Other acquisition function options take a more ‘risky’ stance: they may not improve, but when they do improve, they may improve quite a lot more than the PI sample proposal.

[27: boEIEN] Bayesian Optimization: expected improvement, example
(+LCB,PI,EV)

Materiales: [[CÓD.:](#) BOIntroAcquisitionShortEN.mlx] [[PDF](#)]

****  19:58

[YouTube ▶]

This video, a continuation of [26], will discuss the meaning of the *Expected Improvement* (EI) acquisition function in Bayesian optimization. Basically, Expected value (Gaussian mean) should be used if getting a bad sample penalises some “utility” in my particular application, and expected improvement should be used if bad samples do not incur any cost as we have our ‘best’ sample in our historical record to provide as our final solution to the optimization problem.

A final discussion on the comparison with probability of improvement and confidence bound acquisition functions is also provided.

All discussions are supported via an example that is coincident with that in the above-mentioned video this is a continuation thereof.

[28: boloop2EN] Bayesian optimization: PI, EI, LCB detailed example (Matlab)

Materiales: [[CÓD.:](#) BOIntroLOOPENG.mlx] [[PDF](#)]

****  15:59

[YouTube ▶]

This video presents an example of Bayesian optimization with probability of improvement, expected improvement, and confidence interval bound as acquisition functions. The development is similar to that of the video [24], but with greater detail in the presentation of the Matlab code.

[29: boloop3EN] Bayesian optimization: bad performance examples

17:17

Materiales: [[CÓD.:](#) BOIntroLOOPENG.mlx] [[PDF](#)]

***** 

[YouTube ▶]

This video continues video [28] with additional examples. The aforementioned video is reviewed in the first two minutes, and then the following examples are discussed:

- Example with one local minimum and the global minimum at an extreme of the search range (works OK)
- Example with “overexploitation” that proposes quite unreasonable samples in its final phase.
- Example with prior confidence interval too small: it gets stuck at a local minimum
- Example with too small a bandwidth of the prior (too large correlation distance): exploration is very bad, blindsided, and ends at a point that is not even a local minimum

- Example with too large prior bandwidth (too small correlation distance): it only extrapolates to a very short distance and, therefore, progress is very slow and, before approaching the true optimal zone, it explores in an almost regular mesh (at the distance where correlation vanishes).

With all these examples, we seek to depict a complete image of the advantages and disadvantages of Bayesian optimization, and the importance of a good choice of the prior... or, if the prior is not so good, detect overexploitation and low probability of the samples (given the prior) to adjust the (hyper)parameters of said prior as samples are acquired. These issues are not considered here.

2.3 Refinamientos adicionales

Hiperparámetros del prior

[30: optimML]

Determinación de parámetros de un modelo probabilístico:
 ejemplo Matlab (max likelihood)
Materiales: [[Cód.:](#) MLVza.mlx] [[PDF](#)]

***  07:24
 [YouTube ▶]

Este vídeo presenta un caso sencillo de estimar un parámetro de un modelo probabilístico a partir de realizaciones experimentales maximizando la probabilidad (bueno, en rigor, la función de densidad) de los datos experimentales, esto es, el estimado max-likelihood.

El ejemplo es muy sencillo: dadas tres muestras de una distribución normal de media cero, estimar el parámetro de desviación típica σ de la misma. Si la densidad es $f(y, \sigma)$ y las muestras son independientes, se trata de maximizar $J = f(y_1, \sigma)f(y_2, \sigma)f(y_3, \sigma)$. La maximización en sí se hace por “fuerza bruta”, comprobando un montón de valores a ver cuál da un valor más alto de J . Se comprueba numéricamente que el óptimo coincide con la estimación de la desviación típica muestral $\sqrt{(y_1^2 + y_2^2 + y_3^2)/3}$.

[31: optimkern]

Optimización de hiperparámetros de un kernel (covarianza proc.
 estocástico) para regresión
Materiales: [[Cód.:](#) OptimHiperparameters.zip] [[PDF](#)]

****  12:57
 [YouTube ▶]

En este vídeo se determinan los (hiper)parámetros óptimos de un Kernel gaussiano $Me^{-\frac{x^2}{\sigma^2}}$ para ajustar una serie de datos.

Se utilizan dos criterios: – ajuste a unos datos de validación (error cuadrático, sin ponderar según vza., por simplicidad) – estimado maximum-likelihood (maximizar la probabilidad a priori de los datos experimentales). Una versión más sencilla de la idea se discute en el vídeo [30] que podría ser aconsejable visionar antes de éste.

Se comparan los resultados entre sí y con ajustes “manuales” de dichos parámetros.

Modelos “semiparamétricos”

Puede combinarse regresión por proceso gaussiano con cierto Kernel con un componente paramétrico, como en ajustes de mínimos cuadrados.

[32: semipar1]

Modelos semiparamétricos (Kernel+regresores): estimación de
 salidas y parámetros
Materiales: [SemiParametricRegression.pdf]

****  07:53
 [YouTube ▶]

Este vídeo plantea la regresión en modelos $y(x) = z(x) + \phi(x)\theta + \epsilon$ siendo ϵ un ruido de medida, θ un vector de parámetros ajustables, $\phi(x)$ un vector de regresores, y $z(x)$ un componente de tipo “proceso Gaussiano” con matriz de varianzas-covarianzas $\kappa(x_1, x_2)$ conocida.

El resultado es equivalente a utilizar en regresión un Kernel dado por $\tilde{\kappa}(x_1, x_2) = \kappa(x_1, x_2) + \phi(x_1)\Sigma_\theta\phi(x_2)^T$, siendo Σ_θ la matriz de varianzas-covarianzas del estimado *a priori* del componente paramétrico θ .

[33: semipar2]

Modelos semiparamétricos (Kernel+regresores): notas adicionales

09:00

Materiales:

[SemiParametricRegression.pdf]

[YouTube ▶]

Este vídeo amplía ciertos detalles del vídeo [32]. Los primeros dos minutos resumen rápidamente las ideas de dicho vídeo.

Luego, considera la estimación de la confianza de los parámetros explícitos estimados (matriz de varianzas-covarianzas a posteriori de $\hat{\theta}$), y considera los casos particulares de identificación “kernel” (proceso gaussiano) pura ($\Sigma_\theta = 0$) y los mínimos cuadrados recursivos ($\kappa(x_1, x_2) = 0$). Por último, enfatiza el caso particular de proceso gaussiano con media μ no nula que debe ser estimada (conocido como *ordinary kriging* en la literatura), que corresponde a $\phi(x) = 1$ de modo que $y(x) = z(x) + \mu + \epsilon$, siendo μ constante y ϵ ruido (blanco) de medida de una cierta varianza.

[34: semipcls]

Modelos semiparamétricos (Kernel+regresores): clase Matlab para ejemplos

07:39

Materiales:

[CÓD.: KernelRegressionClass.m] [PDF]

[YouTube ▶]

Este vídeo describe la estructura de una clase Matlab para ejecutar la regresión semiparamétrica (kernel de covarianza + componente paramétrico) cuya teoría se discute en los vídeos [32] y [33].

Esta clase será utilizada en el vídeo [35].

[35: semipcaso]

Modelos semiparamétricos (Kernel+regresores): caso de estudio Matlab

14:40

Materiales:

[CÓD.: semiparametricKernelRegressionTestsV2.mlx] [PDF]

[YouTube ▶]

Este vídeo presenta un ejemplo de regresión semiparamétrica (kernel de covarianza + componente paramétrico) cuya teoría se discute en los vídeos [32] y [33]. Se utiliza la clase Matlab cuyo código se analiza en el vídeo [34]. Es aconsejable, por tanto, el visionar dichos vídeos previos para poder comprender el ejemplo aquí desarrollado.

Nota: sin descargar el fichero `.m` asociado al vídeo [34] en el mismo directorio que el `.mlx` aquí, el código `mlx` no funcionará correctamente.

En concreto, se genera una serie de datos y se comparan:

- Regresión lineal (recta de regresión, mínimos cuadrados)
- Regresión no paramétrica (kernel $\kappa(x_1, x_2)$ de media cero), a veces denominado *simple kriging*.
- Regresión semiparamétrica estimando media ($\kappa(x_1, x_2)$ y componente constante μ), denominado *ordinary kriging* en la literatura (bueno, cierto caso límite).
- Regresión semiparamétrica estimando un componente lineal $\theta_1 x + \theta_2$, adicional al componente de covarianza $\kappa(x_1, x_2)$.

Se comparan los estimados en media y varianza, así como los intervalos de confianza de los parámetros explícitos θ estimados.

Nota: la selección de hiperparámetros (parámetros de la función κ) ha sido más o menos “manual” para que salgan gráficas “bonitas”. Un enfoque más adecuado debería, posiblemente, incluir una optimización de hiperparámetros tal y como se considera en el vídeo [31].