

Distribución normal multidimensional: generación de muestras

© 2020, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Presentación en vídeo: <http://personales.upv.es/asala/YT/V/dadonorm2d.html>

Este código ejecutó con Matlab R2020b

Objetivo: usar la función `randn` de Matlab que genera variables normales de media cero y varianza 1 (distribución normal estándar), para generar distribuciones de probabilidad normal multivariable de matriz de varianzas-covarianzas Σ dada.

Tabla de Contenidos

Diagonalización de la matriz de varianzas-covarianzas, componentes principales.....	1
Generación de muestras.....	2
RESUMEN/Conclusiones.....	4

Diagonalización de la matriz de varianzas-covarianzas, componentes principales

La distribución normal n -dimensional tiene función de densidad:

$$f_X(\psi_1, \dots, \psi_n) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left(-\frac{1}{2} (\boldsymbol{\psi} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{\psi} - \boldsymbol{\mu}) \right)$$

Consideremos que $\boldsymbol{\psi}$ es una variable aleatoria normal bidimensional $\boldsymbol{\psi} = \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} \in \mathbb{R}^2$ de media cero,

con matriz VC dada por $E[\boldsymbol{\psi}\boldsymbol{\psi}^T] = \Sigma$:

```
Sigma=[4 -6.6;-6.6 16]
```

```
Sigma = 2x2  
    4.0000    -6.6000  
   -6.6000    16.0000
```

```
N=size(Sigma,1)
```

```
N = 2
```

Si diagonalizamos la matriz Σ

```
[V,D]=eig(Sigma)
```

```
V = 2x2  
   -0.9145   -0.4046
```

```

-0.4046    0.9145
D = 2x2
  1.0804    0
    0    18.9196

```

Se verifica que V es ortogonal (Σ es simétrica definida positiva), o sea $V^T = V^{-1}$, $V^T V = I$, y $VDV^T = \Sigma$

```
V*D*V'
```

```

ans = 2x2
  4.0000   -6.6000
 -6.6000   16.0000

```

```
V'*Sigma*V
```

```

ans = 2x2
  1.0804    0.0000
 -0.0000   18.9196

```

Por tanto si definimos $\phi = V^T \psi$, tenemos

$$E[\phi \phi^T] = E[V^T \psi \psi^T V] = V^T E[\psi \psi^T] V = V^T \Sigma V = D$$

con lo que las variables ϕ son "no correladas" ($\equiv$ independientes, en el caso distrib. normal), cada componente con varianza D_{ii} y desviación típica $\sigma_i = \sqrt{D_{ii}}$. Estas variables ϕ se denominan "componentes principales" de la distribución de probabilidad de ψ .

```
desvtip_componentesprinc=sqrt(diag(D))
```

```

desvtip_componentesprinc = 2x1
  1.0394
  4.3497

```

El cambio inverso es $\psi = V\phi$. En efecto, su matriz de vzas-covarianzas será

$$E[\psi \psi^T] = E[V\phi \cdot \phi^T V^T] = V \cdot E[\phi \phi^T] \cdot V^T = VDV^T = \Sigma$$

Generación de muestras

Tirar el dado en los componentes principales es "fácil": son variables independientes, con lo cual se trata de usar la función randn (desv. típica 1) y multiplicarla por la desv. típica de cada componente.

```
unatirada_componentesprinc=diag(desvtip_componentesprinc)*randn(N,1)
```

```

unatirada_componentesprinc = 2x1
  0.5589
  7.9768

```

```
unatirada_psi=V*unatirada_componentesprinc
```

```
unatirada_psi = 2×1
-3.7381
 7.0688
```

Vamos a repetir miles de veces estas tiradas, para calcular varianzas-covarianzas y comprobar si son las especificadas:

```
m=10000;
muchastiradas_componentesprinc=diag(desvtip_componentesprinc)*randn(N,m);
```

*Nosotros considerábamos nuestras variables aleatorias como vector columna... La statistics toolbox lo considera como vector FILA, hay que transponer:

```
cov(muchastiradas_componentesprinc') %diagonal, son independientes
```

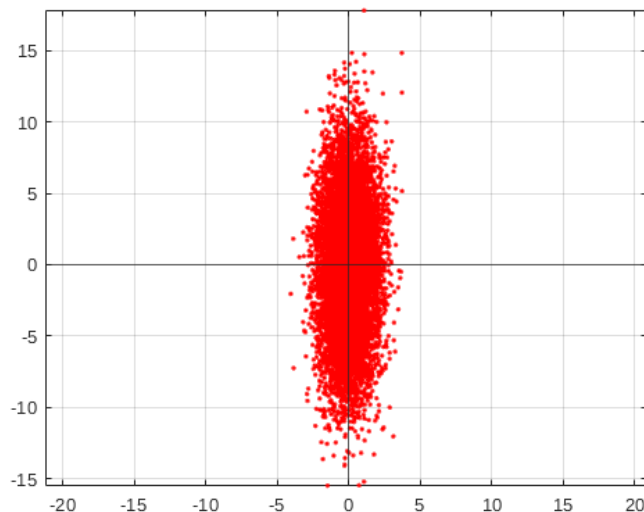
```
ans = 2×2
 1.0719    0.1390
 0.1390   18.3591
```

```
std(muchastiradas_componentesprinc') %coincide (muestras finitas) con lo previsto
```

```
ans = 1×2
 1.0353    4.2848
```

Los componentes principales se distribuyen en una elipse alineada con los ejes:

```
plot(muchastiradas_componentesprinc(1,:),muchastiradas_componentesprinc(2,:),'.r'), grid on
xline(0), yline(0)
```



Tras el cambio de variable, tenemos las tiradas de las variables buscadas:

```
muchastiradas_psi=V*muchastiradas_componentesprinc;
size(muchastiradas_psi)
```

```
ans = 1x2
      2      10000
```

Transponemos para pasar a una fila por tirada:

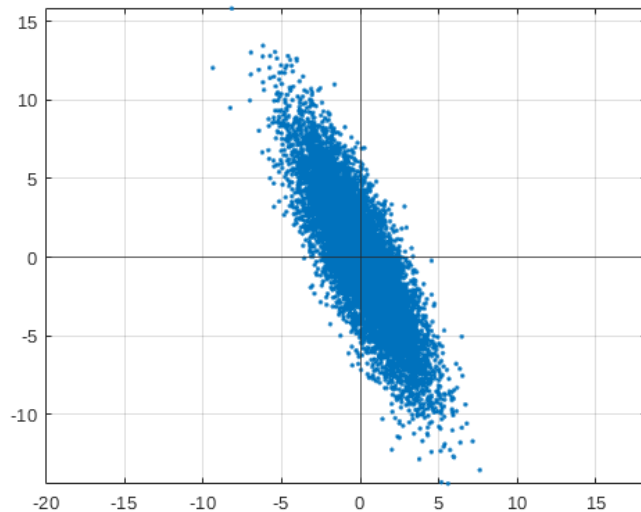
```
cov(muchastiradas_psi')
```

```
ans = 2x2
      4.0041    -6.4893
     -6.4893    15.4270
```

Las covarianzas coinciden con Σ cuando el número de muestras $\rightarrow \infty$.

El cambio "V" ortogonal es un "giro" del elipsoide principal a otras coordenadas.

```
plot(muchastiradas_psi(1,:),muchastiradas_psi(2,:),'.'), grid on, axis equal
xline(0), yline(0)
```



RESUMEN/Conclusiones

Para tirar "m veces" el dado de una distribución normal N-dimensional (columna) de media cero y matriz de Varianzas-covarianzas $\Sigma = VDV^T$, podemos ejecutar:

```
tiradas_columna=V*sqrt(D)*randn(N,m);
size(tiradas_columna)
```

```
ans = 1x2
      2      10000
```

```
cov(tiradas_columna')
```

```
ans = 2x2
      3.8808    -6.3653
```

-6.3653 15.5570

Para tirar "m veces" el dado de una distribución normal N-dimensional (FILA) de media cero y matriz de Varianzas-covarianzas $\Sigma = VDV^T$, podemos ejecutar:

```
tiradas_fila=randn(m,N)*sqrt(D)*V';  
size(tiradas_fila)
```

```
ans = 1x2  
      10000      2
```

```
cov(tiradas_fila)
```

```
ans = 2x2  
      4.0718    -6.6674  
     -6.6674    16.0974
```

Obviamente, si la media de la distribución normal a generar muestras no fuera cero, habría que sumarla a estos resultados (tiradas_columna o tiradas_fila).

Nota: La diagonalización calcula la "raíz cuadrada" de la matriz de varianzas-covarianzas $Q = V \cdot \sqrt{D}$, de modo que $\Sigma = QQ^T$, que es por lo que hay que multiplicar `\texttt{randn}`; esta raíz cuadrada de la matriz tiene la interpretación de "componentes principales" muy útil en muchas aplicaciones, pero existen otras raíces cuadradas (descomposición alternativa $\Sigma = QQ^T$), más rápidas computacionalmente como la descomposición de Cholesky, con Q triangular, que también son válidas para generar las muestras: si $\psi = Q \cdot s$ con $\text{cov}(s) = I$ entonces $E[\psi\psi^T] = Q \cdot I \cdot Q^T = \Sigma$. Esta descomposición es la usada en la documentación de Matlab como ejemplo (<https://www.mathworks.com/help/matlab/ref/randn.html>) para generar distribuciones normales multidimensionales.