Opinionated Lessons in Statistics

by Bill Press

#17 The Multivariate Normal Distribution

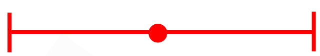
Multivariate Normal Distributions

Generalizes Normal (Gaussian) to M-dimensions
Like 1-d Gaussian, completely defined by its mean and (co-)variance
Mean is a M-vector, covariance is a M x M matrix

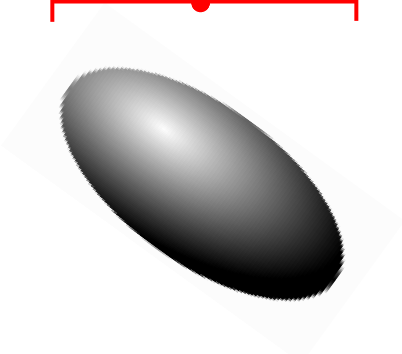
$$N(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{M/2}\det(\boldsymbol{\Sigma})^{1/2}} \exp[-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})]$$

The mean and covariance of r.v.'s from this distribution are*

$$\boldsymbol{\mu} = \langle \mathbf{x} \rangle \qquad \boldsymbol{\Sigma} = \left\langle (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T \right\rangle$$



In the one-dimensional case σ is the standard deviation, which can be visualized as "error bars" around the mean.



In more than one dimension Σ can be visualized as an error ellipsoid around the mean in a similar way.

$$1 = (\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

*shouldn't we prove this?

Because mean and covariance are easy to estimate from a data set, it is easy – perhaps too easy – to fit a multivariate normal distribution to data.

$$\boldsymbol{\mu} = \langle \mathbf{x} \rangle \approx \frac{1}{N} \sum_i \mathbf{x}_i \qquad \boldsymbol{\Sigma} = \left\langle (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T \right\rangle \approx \frac{1}{N} \sum_i (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

Estimate by sample averages (turns out to be the maximum likelihood estimate)

But back to moments. The mean follows from the symmetry argument

$$0 = \int \cdots \int (\mathbf{x}-\boldsymbol{\mu}) \frac{1}{(2\pi)^{M/2} \det(\boldsymbol{\Sigma})^{1/2}} \exp[-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})]\, d^M\mathbf{x}$$

It's *not* obvious that the covariance in fact obtains from the definition of the multivariate Normal. One has to do the multidimensional (and tensor) integral:

$$\mathbf{M}_2 = \int \cdots \int (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T \frac{1}{(2\pi)^{M/2} \det(\boldsymbol{\Sigma})^{1/2}} \exp[-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})]\, d^M\mathbf{x}$$

The only way I know how to do this integral is by trickery involving the Cholesky decomposition ("square root of a positive definite matrix"):

$$\mathbf{\Sigma} = \mathbf{L}\mathbf{L}^T \text{ (Cholesky)}, \quad \mathbf{\Sigma}^{-1} = (\mathbf{L}^T)^{-1}\mathbf{L}^{-1}, \quad \mathbf{L}\mathbf{y} \equiv \mathbf{x}$$

we're setting μ to 0 for convenience

$$p(\mathbf{y}) = p(\mathbf{x}) \left| \frac{\partial \mathbf{x}}{\partial \mathbf{y}} \right|$$

Jacobian determinant. The transformation law for multivariate probability distributions.

$$= \frac{\det(\mathbf{L})}{(2\pi)^{N/2}\det(\mathbf{\Sigma})^{1/2}} \exp[-\tfrac{1}{2}(\mathbf{y}^T\mathbf{L}^T)(\mathbf{L}^{T-1}\mathbf{L}^{-1})(\mathbf{L}\mathbf{y})]$$

$$= \prod_i (2\pi)^{-1/2} \exp(-\tfrac{1}{2}y_i^2)$$

This is the distribution of N independent univariate Normals N(0,1)!

$$\left\langle \mathbf{x}\mathbf{x}^T \right\rangle = \left\langle \mathbf{L}\mathbf{y}\mathbf{y}^T\mathbf{L}^T \right\rangle = \mathbf{L}\left\langle \mathbf{y}\mathbf{y}^T \right\rangle \mathbf{L}^T = \mathbf{L}\mathbf{L}^T = \mathbf{\Sigma}$$

Ha!

(I don't know an elementary proof, i.e., without some matrix decomposition. Can you find one?)

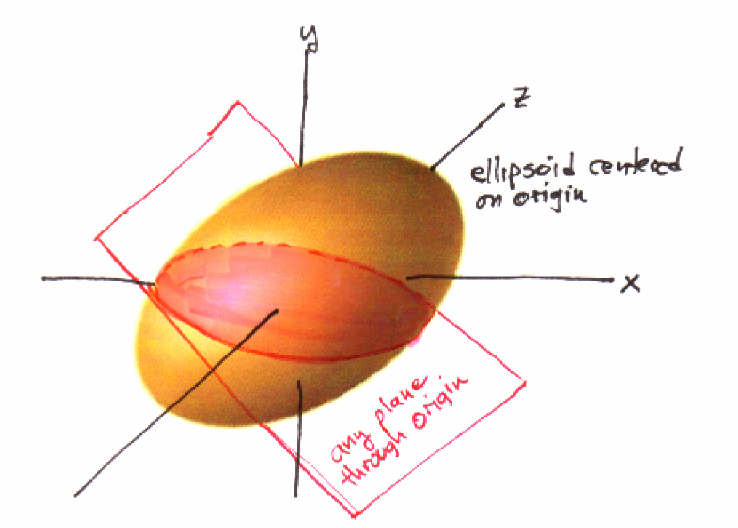
Reduced dimension properties of multivariate normal

$$N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{M/2} \det(\boldsymbol{\Sigma})^{1/2}} \exp[-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})]$$

1. Any slice through a m.v.n. is a m.v.n ("constraint" or "conditioning")

2. Any projection of a m.v.n. is a m.v.n ("marginalization")

You can prove both assertions by "completing the square" in the exponential, producing an exponential in (only) the reduced dimension times an exponential in (only) the lost dimensions. Then the second exponential is either constant (slice case) or can be integrated over (projection case).



How to generate multivariate normal deviates N(μ,Σ):

Cholesky: $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$

Fill $\mathbf{y}$ with independent Normals: $\mathbf{y} = \{y_i\} \sim \mathrm{N}(0,1)$

Transform: $\mathbf{x} = \mathbf{L}\mathbf{y} + \boldsymbol{\mu}$ That's it! x is the desired m.v.n.

Proof: $\left\langle \mathbf{y}\mathbf{y}^T \right\rangle = \mathbf{1}$

$$\begin{aligned}
\left\langle (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T \right\rangle &= \left\langle (\mathbf{L}\mathbf{y})(\mathbf{L}\mathbf{y})^T \right\rangle \\
&= \left\langle \mathbf{L}(\mathbf{y}\mathbf{y}^T)\mathbf{L}^T \right\rangle = \mathbf{L}\left\langle \mathbf{y}\mathbf{y}^T \right\rangle \mathbf{L}^T \\
&= \mathbf{L}\mathbf{L}^T = \boldsymbol{\Sigma}
\end{aligned}$$

Even easier: MATLAB has a built-in function mvnrnd(MU,SIGMA).

Notice that the proof never used Normality. You can fill **y** with anything with zero mean and variance one, and you'll reproduce Σ. But the result won't be Normal!

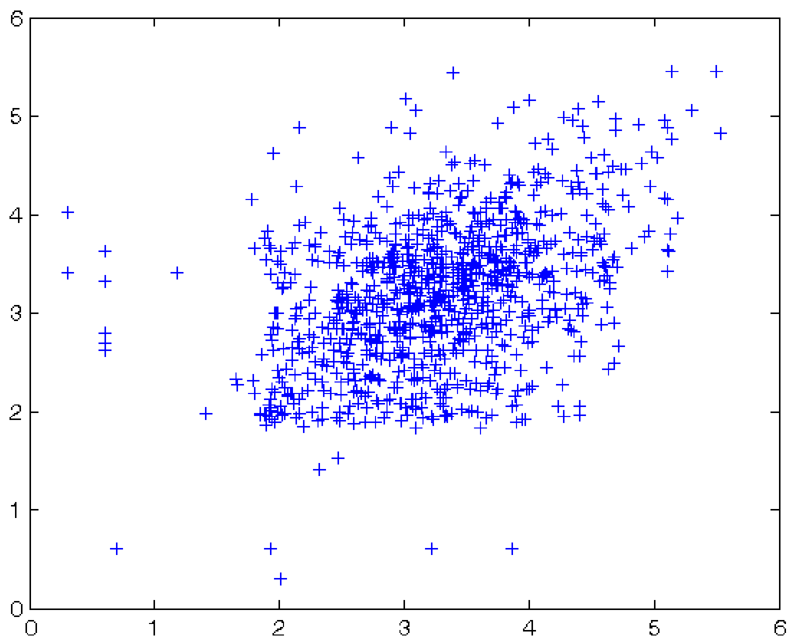
So, easy operations are:

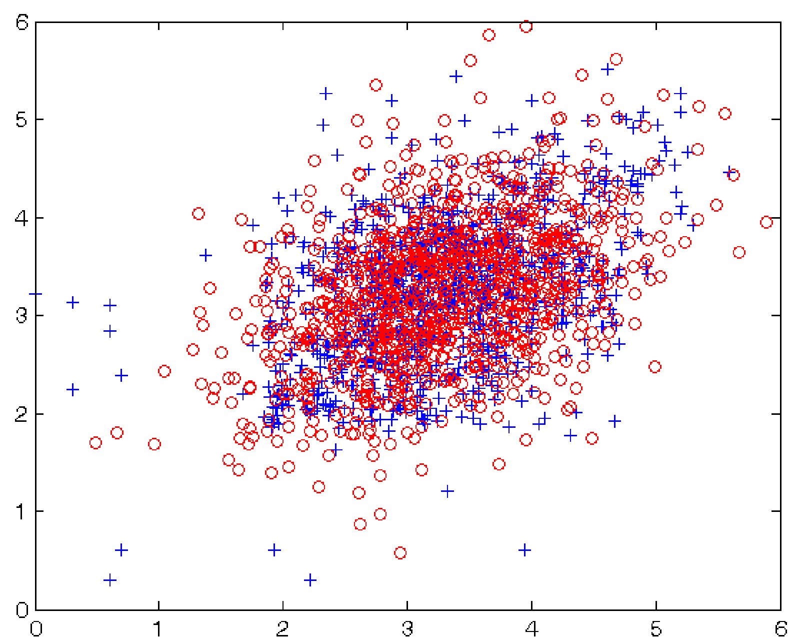
1. Fitting a multivariate normal to a set of points (just compute the sample mean and covariance!)
2. Sampling from the fitted m.v.n.

```
mu = mean([len1 len2])
sig = cov(len1,len2)
mu =
    3.2844
    3.2483
sig =
    0.6125    0.2476
    0.2476    0.5458
rsamp = mvnrnd(mu,sig,1000);
```

In MATLAB, for example, these are one-line operations.

Example:





A related, useful, Cholesky trick is to draw error ellipses (ellipsoids, …)

$$\mathbf{\Sigma} = \mathbf{L}\mathbf{L}^T$$

So, locus of points at 1 standard deviation is

$$1 = (\mathbf{x}-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}) \quad \Rightarrow \quad \left|\mathbf{L}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right| = 1$$

So, if z is on the unit circle (sphere, …) then

$$\mathbf{x} = \mathbf{L}\mathbf{z} + \boldsymbol{\mu}$$

will be on the error ellipse.

my coding of this idea looks like this

```
function [x y] = errorellipse(mu,sigma,stdev,n)
L = chol(sigma,'lower');
circle =
 [cos(2*pi*(0:n)/n); sin(2*pi*(0:n)/n)].*stdev;
ellipse = L*circle + repmat(mu,[1,n+1]);
x = ellipse(1,:);
y = ellipse(2,:);
```