Multivariate Gaussians

[This note assumes that you know the background material on expectations of random variables.]

We're going to use Gaussian distributions as parts of models of data, and to represent beliefs about models. Most models and algorithms in machine learning involve more than one scalar variable however. (A scalar meaning a single number, rather than a vector of values.) Multivariate Gaussians generalize the univariate Gaussian distribution to multiple variables, which can be dependent.

1 Independent Standard Normals

We could sample a vector $\mathbf{x}$ by independently sampling each element from a standard normal distribution, $x_d \sim \mathcal{N}(0,1)$. Because the variables are independent, the joint probability is the product of the individual or marginal probabilities:

$$p(\mathbf{x}) = \prod_{d=1}^{D} p(x_d) = \prod_{d=1}^{D} \mathcal{N}(x_d; 0, 1). \tag{1}$$

Usually I recommend that you write any Gaussian PDFs in your maths using the $\mathcal{N}(x; \mu, \sigma^2)$ notation unless you have to expand them. It will be less writing, and clearer. Here, I want to combine the PDFs, so will substitute in the standard equation:

$$p(\mathbf{x}) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi}} e^{-x_d^2/2} = \frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2}\sum_d x_d^2} \tag{2}$$

$$= \frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2}\mathbf{x}^\top \mathbf{x}}. \tag{3}$$

The PDF is proportional to the Radial Basis Functions (RBFs) we've used previously. Here the normalizer $1/(2\pi)^{D/2}$ means that the PDF integrates to one.

Like an RBF centred at the origin, this density function only depends on the square-distance or radius of $\mathbf{x}$ from the origin. Any point in a spherical shell (or a circular shell in 2-dimensions) is equally probable. Therefore if we simulate points in 2-dimensions and draw a scatter plot:

```
# Python
N = int(1e4); D = 2
X = np.random.randn(N, D)
plt.plot(X[:,0], X[:,1], '.')
plt.axis('square')
plt.show()
```

We will see a diffuse circular spray of points. The spherical symmetry is a special property of Gaussians. If you were to draw independent samples from, say, a Laplace distribution you would see a non-circular distribution that has more density close to the axes.

2 Covariance

The multivariate generalization of variance, is *covariance*, which is represented with a matrix. While a variance is often denoted σ^2, a covariance matrix is often denoted Σ—not to be confused with a summation $\sum_{d=1}^{D} \ldots$

The elements of the covariance matrix for a random vector $\mathbf{x}$ are:

$$\mathrm{cov}[\mathbf{x}]_{ij} = \mathbb{E}[x_i x_j] - \mathbb{E}[x_i]\mathbb{E}[x_j]. \tag{4}$$

On the diagonal, where $i = j$, you will see that this definition gives the scalar variances $\text{var}[x_i]$ for each of the elements of the vector. We can write the whole matrix with a linear algebra expression:

$$\text{cov}[\mathbf{x}] = \mathbb{E}[\mathbf{x}\mathbf{x}^\top] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^\top. \tag{5}$$

Question: What is the covariance S of the spherical distribution of the previous section? (We will reserve Σ for the covariance of the general Gaussian in the next section.)

The video will take you through the answer, so we're not making you type it in. But by considering each element S_{ij}, you should be able to derive the answer yourself using the results in the review notes on expectations.

Answer: The first term is $\mathbb{E}[x_i x_j] = \mathbb{E}[x_i]\mathbb{E}[x_j]$ if x_i and x_j are independent, which they are if $i \neq j$. Thus $S_{i\neq j} = 0$. The diagonal elements S_{ii} are equal to the variances of the individual variables, which are all equal to one. Therefore, $S_{ij} = \delta_{ij}$, where δ_{ij} is a Kronecker delta. Or as a matrix, $S = \mathbb{I}$, the identity matrix.

2.1 Empirical covariance

The covariance above is a formal property of a distribution.

An *empirical covariance* (or *sample covariance*) is where the expectations in the definition of covariance, are replaced with averages over samples[1]. In NumPy, `np.cov` computes a covariance using expectations under a uniform distribution over N samples. Annoyingly this function requires an input of shape `(D,N)`, so an `(N,D)` design matrix must be transposed.

If you have any doubt how covariances are computed, you should write your own version of `cov` from primitive matrix operations, and check agreement with `np.cov`.

3 Transforming and Rotating: general Gaussians

As with one-dimensional Gaussians, we can generalize the standard zero-mean, unit-variance Gaussian by a linear transformation and a shift. If any of the steps here are unclear, make sure you are comfortable with the univariate Gaussian note first.

If we generated the elements of $\mathbf{x}$ from independent $\mathcal{N}(0,1)$ draws as above, we could form a linear combination of these outcomes:

$$\mathbf{y} = A\mathbf{x}. \tag{6}$$

To keep the discussion simpler, I will assume that A is square and invertible, so $\mathbf{y}$ has the same dimensionality as $\mathbf{x}$.

Question: What is the covariance Σ of the new variable $\mathbf{y}$?

Answer: Simply substitute $\mathbf{y}$ into the definition:

$$\begin{aligned}
\text{cov}[\mathbf{y}] &= \mathbb{E}[\mathbf{y}\mathbf{y}^\top] - \mathbb{E}[\mathbf{y}]\mathbb{E}[\mathbf{y}]^\top && (7)\\
&= \mathbb{E}[A\mathbf{x}\mathbf{x}^\top A^\top] - \mathbb{E}[A\mathbf{x}]\mathbb{E}[A\mathbf{x}]^\top && (8)\\
&= A\mathbb{E}[\mathbf{x}\mathbf{x}^\top]A^\top - A\mathbb{E}[\mathbf{x}](A\mathbb{E}[\mathbf{x}])^\top. && (9)
\end{aligned}$$

Because $\mathbb{E}[\mathbf{x}]$ is zero, the second term is zero, and the expectation in the first term is equal to $\text{cov}[\mathbf{x}] = \mathbb{I}$. Therefore,

$$\text{cov}[\mathbf{y}] = \Sigma = AA^\top. \tag{10}$$

1. There is also an "$(N-1)$" version of the estimator, just as there is for estimating variances.

Because we're assuming A is invertible, we can compute the original vector from the transformed one: $\mathbf{x} = A^{-1}\mathbf{y}$. Substituting that expression into the PDF for $\mathbf{x}$ we can see the shape of the new PDF:

$$p(\mathbf{y}) \propto e^{-\frac{1}{2}(A^{-1}\mathbf{y})^\top (A^{-1}\mathbf{y})} \tag{11}$$

$$\propto e^{-\frac{1}{2}\mathbf{y}^\top A^{-\top} A^{-1}\mathbf{y}}. \tag{12}$$

As we saw in the univariate Gaussian note, if we stretch out a PDF, we must scale it down so that the distribution remains normalized. If we apply a linear transformation A to a volume of points, then the volume is multiplied by $|A|$, the determinant of the matrix.[2] Therefore,

$$p(\mathbf{y}) = \frac{1}{|A|(2\pi)^{D/2}}\, e^{-\frac{1}{2}\mathbf{y}^\top A^{-\top} A^{-1}\mathbf{y}}. \tag{13}$$

Usually this expression is re-written in terms of the covariance of the vector. Noticing that

$$\Sigma^{-1} = A^{-\top} A^{-1}, \text{ and} \tag{14}$$

$$|\Sigma| = |AA^\top| = |A||A^\top| = |A|^2, \tag{15}$$

we can write:

$$p(\mathbf{y}) = \frac{1}{|\Sigma|^{1/2}(2\pi)^{D/2}}\, e^{-\frac{1}{2}\mathbf{y}^\top \Sigma^{-1}\mathbf{y}} = |2\pi\Sigma|^{-1/2}\, e^{-\frac{1}{2}\mathbf{y}^\top \Sigma^{-1}\mathbf{y}}. \tag{16}$$

As demonstrated above, there are different equivalent ways to write the normalizing constant, and different books will choose different forms.[3]

Finally, we can shift the distribution to have non-zero mean:

$$\mathbf{z} = \mathbf{y} + \boldsymbol{\mu}. \tag{17}$$

Shifting the PDF does not change its normalization, so we can simply substitute $\mathbf{y} = \mathbf{z} - \boldsymbol{\mu}$ into the PDF for $\mathbf{y}$:

$$\boxed{p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \Sigma) = \frac{1}{|\Sigma|^{1/2}(2\pi)^{D/2}}\, e^{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{z}-\boldsymbol{\mu})}.} \tag{18}$$

Here we've generalized the $\mathcal{N}$ notation for Gaussian distributions to take a mean vector and a matrix of covariances. In one-dimension, these quantities still correspond to the scalar mean, and the variance.

It's a common mistake to forget the matrix inverse inside the exponential. The inverse covariance matrix Σ^{-1}, is also known as the precision matrix.

4 Covariances are positive (semi-)definite

[This section may be tough going on first reading. If so, that's ok: just keep going to the "check your understanding" section, which you should work through.]

---

2. Here $|A|$ is the "Jacobian of the transformation", although confusingly "Jacobian" can refer to both a matrix and its determinant. The change of variables might be clearer if we label the different probability density functions: $p_Y(\mathbf{y}) = p_X(\mathbf{x})/|A| = p_X(A^{-1}\mathbf{y})/|A|$. See also the further reading section.

3. Over the years, many students have questioned whether and why $1/(|\Sigma|^{1/2}(2\pi)^{D/2}) = |2\pi\Sigma|^{-1/2}$. The first thing I do when unsure, is check an example numerically. For example:

```
D = 5; Sigma = np.cov(np.random.randn(D, 10*D))
lhs = 1 / (np.linalg.det(Sigma)**0.5 * (2*np.pi)**(D/2))
rhs = np.linalg.det(2*np.pi*Sigma)**-0.5
```

To understand why they're equal: for a scalar c and a matrix A, we can write $|cA| = |c\mathbb{I}A| = |c\mathbb{I}||A| = c^D|A|$. The transformation $c\mathbb{I}$ stretches an object in each of D directions by c. The determinant gives the resulting volume change of c^D.

Covariance matrices are always symmetric: in the definition of covariance $\text{cov}[x_i, x_j] = \text{cov}[x_j, x_i]$ or $\Sigma_{ij} = \Sigma_{ji}$. Moreover, just as variances must be positive — or zero if we are careful — there is a positive-like constraint on covariance matrices.

A real[4] symmetric matrix Σ is *positive definite* iff[5] it satisfies:

$$\mathbf{z}^\top \Sigma \mathbf{z} > 0, \quad \text{for all real vectors } \mathbf{z} \neq \mathbf{0}. \tag{19}$$

these matrices are always invertible, and the inverse is also positive definite:

$$\mathbf{z}^\top \Sigma^{-1} \mathbf{z} > 0, \quad \text{for all real vectors } \mathbf{z} \neq \mathbf{0}. \tag{20}$$

Therefore, the exponential term in the probability density of a Gaussian falls as we move away from the mean in any direction iff the covariance is positive definite. If the density didn't fall in some direction then it wouldn't be normalized (integrate to one), and so wouldn't be a valid density.

Edge case: In general, covariances can be *positive semi-definite*, which means:

$$\mathbf{z}^\top \Sigma \mathbf{z} \geq 0, \quad \text{for all real vectors } \mathbf{z}. \tag{21}$$

However, if $\mathbf{z}^\top \Sigma \mathbf{z} = 0$ for some $\mathbf{z} \neq 0$, then the determinant $|\Sigma|$ will be zero, and the covariance won't be invertible. Therefore the expression we gave for the probability density is only valid for strictly positive definite covariances.

An example of a Gaussian distribution where the covariance isn't strictly positive definite can be simulated by drawing $x_1 \sim \mathcal{N}(0, 1)$ and deterministically setting $x_2 = x_1$. You should be able to show that the theoretical covariance of such vectors is:

$$\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \tag{22}$$

and you should be able to simulate this process to confirm numerically[6].

In this example, the probability density is zero "almost everywhere", for any $\mathbf{x}$ where $x_1 \neq x_2$. The only way to make $\int p(\mathbf{x})\, \mathrm{d}\mathbf{x} = 1$ is to make the density infinite along the line $x_1 = x_2$. Gaussian distributions with zero-determinant covariances generalize the Dirac delta function, where the distribution is constrained to a surface with zero volume, rather than just a point. Care is required with such distributions, both analytically and numerically. We will stick to strictly positive definite covariances whenever we can.

Given a real-valued matrix A, $\Sigma = AA^\top$ is always positive semi-definite. Moreover, if Σ is symmetric and positive semi-definite, it can always be written in this form. Allowing non-symmetric Σ wouldn't expand the set of probability densities that can be expressed[7]. Therefore, the process for sampling from a Gaussian that was described in this document is general: we can sample from any Gaussian by transforming draws from a standard normal, and such a process always generates points from a distribution with a well-defined covariance.

5 Computing A to sample from $\mathcal{N}(\mathbf{0}, \Sigma)$

Given a covariance matrix Σ, the transformation A is not uniquely defined. For example

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad \text{and} \quad A = \begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix}, \tag{23}$$

4. I often forget such distinctions, because I rarely deal with complex numbers. Although machine learning systems that use complex numbers have been proposed.
5. "iff" means "if and only if".
6. For example: `x1 = np.random.randn(10**6); X = np.hstack((x1[:,None], x1[:,None])); np.cov(X.T)`
7. Because $\mathbf{x}^\top M \mathbf{x} = \mathbf{x}^\top (\frac{1}{2}M + \frac{1}{2}M^\top)\mathbf{x}$, for any $\mathbf{x}$ and square matrix M. So we can replace a non-symmetric precision Σ^{-1} with the symmetric matrix $(\frac{1}{2}\Sigma^{-1} + \frac{1}{2}\Sigma^{-\top})$, and the covariance will be symmetric too.

both give the same product $AA^\top$. However, *any* A such that $\Sigma = AA^\top$, can be used to transform points to sample from $\mathcal{N}(\mathbf{0}, \Sigma)$.

It's common to use a fast and standard matrix routine known as the (lower-triangular) *Cholesky decomposition*, which is easy to call in NumPy:

```
D = 3; Sigma = np.cov(np.random.randn(D, 3*D))
A = np.linalg.cholesky(Sigma)
Sigma_from_A = A @ A.T  # up to round-off error, matches Sigma
```

Do try this out, and look at the matrices; this checking step is not just for beginners! I still routinely check that functions do what I expect, because I am still frequently bitten by nasty surprises. For example, `SciPy` also has a `cholesky` function:

```
import scipy.linalg as sla
A = sla.cholesky(Sigma)
Sigma_wrong = A @ A.T  # doesn't match Sigma!
```

Unlike NumPy, the SciPy version gives the *upper*-triangular Cholesky decomposition by default — a difference you notice if you look at the matrices for a small example. You need to transpose the result, or use `A = sla.cholesky(Sigma, lower=True)`.

6 Check your understanding

Diagonal transformation: A special case of a general transformation A, is a diagonal matrix, that simply stretches each variable independently: $\Lambda_{ij} = \delta_{ij}\sigma_i$. (This Kronecker delta notation was used earlier in the note.) For three dimensions the transformation would be:

$$\Lambda = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}. \tag{24}$$

[The website version of this note has a question here.]

Investigate a special case: You could use Python to sample some points from different multivariate Gaussians, and see how the covariance affects the cloud of points.

For example, you could use a family of transformations parameterized by a:

$$A = \begin{bmatrix} 1 & 0 \\ a & 1-a \end{bmatrix}, \tag{25}$$

What does this transformation do? Is it clear why the variables are dependent for $a \neq 0$? When are the variables maximally dependent[8]? What happens to the PDF as $a \to 1$ and why? Does the covariance matrix have an inverse when $a = 1$?

[The website version of this note has a question here.]

Contours: The shape of a two-dimensional Gaussian is often sketched using a contour of its PDF. Just like the Radial Basis Function (RBF) discussed last week, the contours of a radially symmetric Gaussian are circular. So if you compute the (x_1, x_2) coordinates of some points on a circle, these can be joined up to plot a contour of the Gaussian with identity covariance. You can then transform these points, just like sample positions, with a matrix A, to plot a contour of $\mathcal{N}(\mathbf{x}; \mathbf{0}, AA^\top)$.

[The website version of this note has a question here.]

8. We don't need a formal definition of dependence here. If you first understand the transformation, this question has a clear answer and is not ambiguous.

7 Further reading

https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Both Bishop Section 2.3 and Barber Section 8.4 start with the definition that this note builds up to, and then works in the reverse direction from there to build up an interpretation. These sections then go further than this note, and both these books have some further exercises. The treatment by Murphy, Section 2.5.2, is rather more terse!

Transforming the PDF of the spherical distribution required getting the normalization correct due to the change of variables. If you would like a more rigorous treatment, or to understand what to do if the transformation is non-linear, I'll defer to the text books. The maths for transforming a PDF due to a change of variables is quickly reviewed in Barber Section 8.2, Result 8.1. Murphy's treatment is longer this time, in Section 2.6.