The Secret Life of the Covariance Matrix

Raúl Rojas
Computer Science Department
Freie Universität Berlin
January 2009

Abstract

This note reviews some interesting properties of the covariance matrix and its use in the multivariate Gaussian distribution, especially for pattern recognition. Usually, the covariance matrix is taken as a given, and some concepts, such as the Mahalanobis distance, are not motivated well enough. Here we show that we can think of the covariance matrix as a handy storage of the variance of a distribution in all projection or cut directions.

1 Motivation

In pattern recognition problems, we usually have a training set T of labelled samples (x_i, y_i), for $x_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$ and label y_i. In a two class problem, the labels can be $y_i = 1$ or $y_i = -1$. We usually expect a class to *cluster* in a certain region of space, and the other classes to concentrate in separate regions. The individual components of the vectors x_i are called the *features* of the classification problem. Good features produce well-separated clusters. Usually, though, there is some interpenetration of the data clouds. Otherwise classification problems would be fairly easy to solve.

A k-nearest neighbors classifier determines, for a given x to be classified, the set of Euclidean distances from x to each element x_i in the training set. The k-nearest neighbors *vote* for the label of point x. Therefore N distance calculations are necessary for each classification.

In a problem with C classes, we can reduce the number of operations by referring to clusters instead of to all data points. If we compute the mean μ_j of each class j of data points, for $j = 1, \ldots, C$, we can simplify the pattern recognition problem: point x is assigned the class of the nearest cluster mean μ_ℓ, $\ell \in \{1, \ldots, C\}$. The computation is simple and fast. The number of distance calculations is now proportional to C, and not to N.

The problem with this simple approach ist that it disregards the spread of the data point distribution. Consider Figure 1. Here, the point x is exactly in the middle of the line connecting the means μ_1 and μ_2 of two classes. Although the point has the same distance to both means, Class 2 seems more appropriate for x, because there is a large spread in the direction from μ_2 to x, and less spread for class 1 in the direction from μ_1 to x. Therefore, we need some way of incorporating the spread of the distribution in the classification process. We need the covariance matrix.

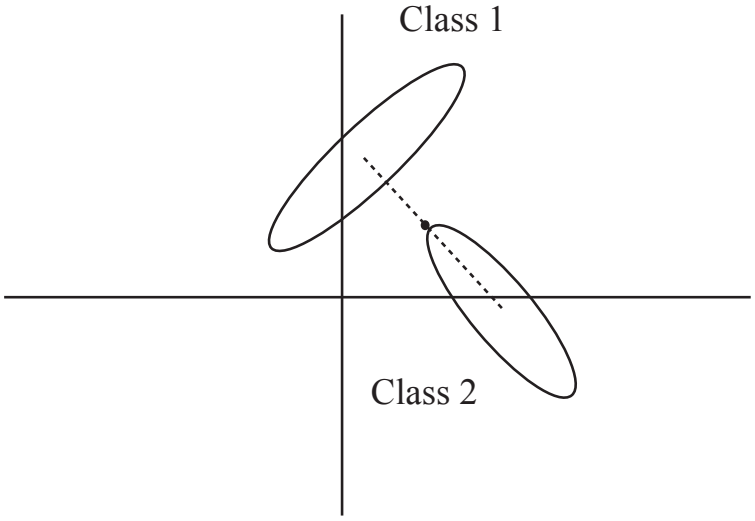

Figure 1: A point in the middle between two class means.

2 The one-dimensional case

In the one-dimensional case, the covariance matrix is just the variance of the distribution of data points.

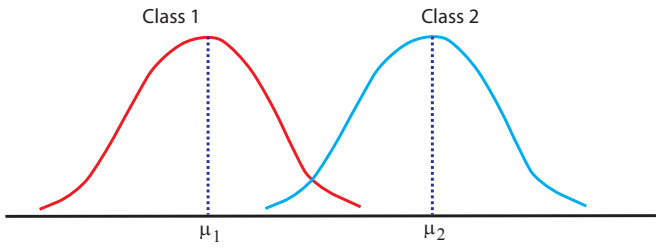

Figure 2: Probability distribution of two one-dimensional classes

Consider Figure 2. It shows histograms of the distribution of two classes of one-dimensional data points. Points from class 1 cluster around μ_1, while class 2 points cluster around μ_2. By definition, the variance of points in class 1 is the averaged squared deviation from the mean:

$$\sigma_1^2 = \frac{1}{N_1} \sum_i^{N_1} (x_i - \mu_1)^2$$

The variance of class 2 is:

$$\sigma_2^2 = \frac{1}{N_2} \sum_i^{N_2} (x_i - \mu_2)^2$$

The 1×1 covariance matrix $[\sigma_1^2]$ of class 1 is just the mean square distance of each point in class 1 to the mean, and similarly for class 2.

Enter the normal distribution

The Gaussian or normal distribution is used frequently in pattern recognition problems. The probability density at the point x is given by the expression:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

Here μ is the mean of the data cloud, and σ^2 the mean square distance of points in the data cloud to the mean. The Gaussian distribution is a model that we can apply in many interesting situations. In the case of two classes, we can select class 1 for point x in case that

$$\frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{1}{2}\frac{(x-\mu_1)^2}{\sigma_1^2}} > \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{1}{2}\frac{(x-\mu_2)^2}{\sigma_2^2}}$$

The inequality means that it is more likely that x belongs to class 1 than to class 2.

Now, in the case of multidimensional variables, we need something similar to the mean squared distance to the mean. If a point x is to be classified, we can measure how similar it is to a point x_i (in the cluster centered around μ)by computing the square of the scalar product of the vectors relative to the center of the cluster μ.

$$d(x, x_i) = \left((x-\mu)^{\mathrm{T}}(x_i-\mu)\right)^2$$

We can repeat this computation for each data point x_i, $i = 1, \ldots, N$, and averaging the results:

$$\begin{aligned} D(x,\mu) &= \tfrac{1}{N}\textstyle\sum_1^N \left((x-\mu)^{\mathrm{T}}(x_i-\mu)\right)^2 \\ &= \tfrac{1}{N}\textstyle\sum (x-\mu)^T(x_i-\mu)(x_i-\mu)^T(x-\mu) \\ &= (x-\mu)^T\left(\tfrac{1}{N}\textstyle\sum (x_i-\mu)(x_i-\mu)^T\right)(x-\mu) \\ &= (x-\mu)^T\Sigma(x-\mu) \end{aligned}$$

where the matrix Σ is defined as

$$\Sigma = \frac{1}{N}\sum_i^N (x_i-\mu)(x_i-\mu)^T$$

This is the much celebrated covariance matrix. We can now investigate its usefulness.

Tomography of the normal distribution

In tomography, slices of the human body are x-rayed from different angles, in order to get a projection on a line of the permeability of the body to x-rays. The line is

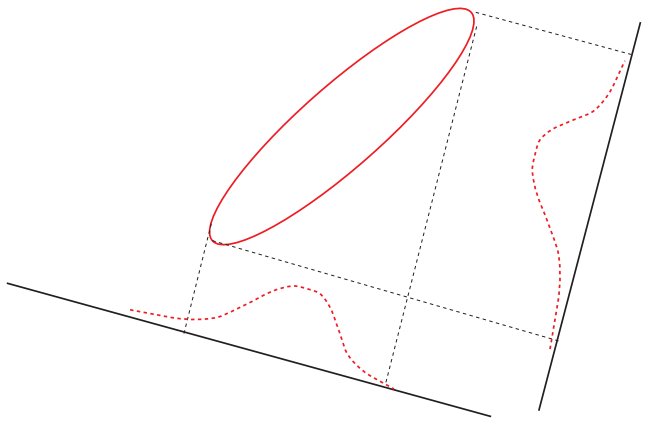

Figure 3: Two one-dimensional projections of the data cloud. The Gaussians represent the respective one-dimensional distribution of the projections.

rotated around the body and the set of projections can be used to reconstruct the two -dimensional distribution of the body's tissues.

In statistics we can also take a tomography of a probability distribution, by projecting the probability density of all data points onto a single direction (see Fig. 3). Given a set of points $x_1, \ldots, x_N$ in R^n with distribution centered around μ and with covariance matrix Σ, the projection of all data points onto a vector u in R^n, where u is of unit length for convenience, is given by $x_i \cdot u$, for $i = 1, \ldots, N$. The mean of the one-dimensional distribution is

$$\hat{u} = \frac{1}{N} \sum_{1}^{N} x_i \cdot u = \left(\frac{1}{N} \sum_{1}^{N} x_i \right) \cdot u = \mu \cdot u$$

That is, the mean of the one dimensional distribution is the projection of μ onto u.

Now, the variance of the one-dimensional projected distribution is given by

$$\sigma^2 = \frac{1}{N} \sum_{i}^{N} (x_i \cdot u - \hat{u})^{\mathrm{T}} (x_i \cdot u - \hat{u})$$

which is the same as

$$\sigma^2 = \frac{1}{N} \sum_{i}^{N} (x_i^{\mathrm{T}} u - \mu^{\mathrm{T}} u)^{\mathrm{T}} (x_i^{\mathrm{T}} u - \mu^{\mathrm{T}} u)$$

and from this we obtain

$$\sigma^2 = \frac{1}{N}\sum_i^N u^{\mathrm{T}}(x_i - \mu)(x_i - \mu)^{\mathrm{T}}u = u^{\mathrm{T}}\Sigma u$$

This is wonderful result: it says that if we want to know the variance of the projection of the whole data distribution with covariance matrix Σ on the unit-length vector u, all we have to do is compute $u^{\mathrm{T}}\Sigma u$. That is, the covariance matrix makes easier to find the variance of the one dimensional projections on any given direction. It is as if the covariance matrix stored all possible projection variances in all directions. All we have to do to "decode" that stored variance, for the direction u, is to compute $u^{\mathrm{T}}\Sigma u$.

3 Eigenvalues of the covariance matrix

Assume that the matrix Σ is of full rank. Since it is a symmetric matrix, it has eigenvectors $e_1, \ldots, e_n$ with respective positive eigenvalues $\lambda_1, \ldots, \lambda_n$. Assume that the eigenvalues have been sorted from largest to smallest.

The eigenvectors (of unit lenght) represent the principal axis of the linear transformation defined by the matrix Σ. They provide us with an orthonormal basis in which all other vectors can be represented.

Let us write the vector u as a linear combination of the eigenvectors $e_1, \ldots, e_n$:

$$u = \alpha_1 e_1 + \cdots + \alpha_n e_n$$

The quadratic form mentioned above $u^{\mathrm{T}}\Sigma u$, that is the variance of the projection of all data points on the line u, reduces to

$$u^{\mathrm{T}}\Sigma u = \alpha_1^2\lambda_1^2 + \cdots + \alpha_n^2\lambda_n^2$$

Since the vector u is of unit length, the α's are constrained by

$$\alpha_1^2 + \cdots + \alpha_n^2 = 1$$

The direction u of maximum spread is therefore that in which we put all the weight on the largest eigenvalue, i.e.,$\alpha_1 = 1$ and therefore $u = e_1$. In this case $u^{\mathrm{T}}\Sigma u = \lambda_1^2$. The direction e_1 is called the first principal component of Σ. The direction e_2 is the second principal component, and so on. The eigenvalues $\lambda_1, \ldots, \lambda_n$ represent the standard deviation of the distribution in each of the respective directions $e_1, \ldots, e_n$.

4 Variance of cuts of the distribution

Now, we could be interested not in the variance of the projection of the whole distribution, but on the variance of the distribution along a certain "cut". Fig. 4 shows the two cases. Given a two dimensional Gaussian distribution we can be interested in the one dimensional projection along the line a or in the cut along the line b. Both have different variances. For judging if the point P fits the distribution well, it is more important to consider the variance along the cut than along the projection, as can be seen in the figure.

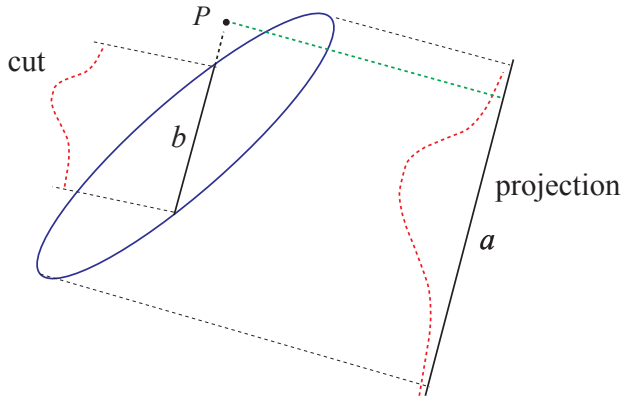

Figure 4: Projection of the whole distribution on line a. Cut of the normal distribution along line b. Line b goes through the center of the distribution.

Given the covariance matrix it is easy to compute the variance along a cut. Let u be the vector represented by

$$u = \alpha_1 e_1 + \cdots + \alpha_n e_n$$

where e_i, for $i = 1, \ldots, n$, represent the eigenvectors of the covariance matrix. In that case

$$u^{\mathrm{T}} \Sigma^{-1} u = \frac{\alpha_1^2}{\lambda_1^2} + \cdots + \frac{\alpha_1^n}{\lambda_1^n}$$

since the eigenvalues of Σ^{-1} are the reciprocals of the eigenvalues of Σ.

The expression $u^T \Sigma u$ is different to $u^T \Sigma^{-1} u$ because the first computes the variance of the projection of the *complete data set* on the line with direction vector

u, while the second computes the *variance of a cut* of the data distribution in the direction u. The cut goes through the middle of the distribution.

To make this explicit, that the cut goes through the middle of the distribution, we refer to the vector x displaced to the new origin μ as $x - \mu$ Therefore, given a point x in the original coordinate center, we are interested in expressions of the form

$$(x - \mu)^{\mathrm{T}} \Sigma^{-1} (x - \mu)$$

This is the variance of the distribution along the line $x - \mu$.

5 Mahalanobis distance

Let us investigate those regions of space for which the variance in a cut is the same, that is, those vectors u for which $u^{\mathrm{T}} \Sigma^{-1} u$ is a constant c. In that case

$$u^{\mathrm{T}} \Sigma^{-1} u = \frac{\alpha_1^2}{\lambda_1} + \cdots + \frac{\alpha_n^2}{\lambda_n} = c$$

This is the equation of the hypersurface of a paraboloid with n axis of respective length $\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_n}$. In the two dimensional case, we have ellipses of constant quadratic variance c. Along each ellipse the spread of the distribution is the same. The constant c is called the Mahalanobis distance of the points along the ellipses defined above to the center of the distribution. The idea of this metric is to measure the variance along all directions around μ.

6 The multivariate normal distribution

While it is not always straightforward to generalize probability distributions to n dimensions, this is easy to do with the normal distribution. For given n-dimensional mean μ and given $n \times n$ covariance matrix Σ the multivariate normal distribution for a point x is a one-dimensional Gaussian in the direction $x - \mu$, that is

$$p(x) = \frac{1}{\sqrt{|2\pi\Sigma|}} \, e^{-\frac{1}{2}(x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu)}$$

It is exactly the same formula as we had before, regarding the exponent of e, since the variance σ^2 in the direction $x - \mu$ is now $(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)$. The normalization factor uses the determinant of the matrix Σ. This factor can be derived by integrating the exponential expression in $\mathbb{R}^n$.

7 Fisher discriminant

Another example of the usefulness of the expressions for the variance of the projections of a distribution is the Fisher linear discriminant. Assume that we want to separate two classes (with respective means μ_1 and μ_2, and covariance matrices Σ_1 and Σ_2) as well as possible, through a projection in one dimension. We are looking for a line such that the sum of the variances of two distributions is as low as possible, while the distance between the means is as high as possible. The expression for S given below tries to achieve this: it grows when the distance between the class means is large along the line u and when the sum of the variance of the projections of the two classes is low:

$$S(u) = \frac{|\mu_1 \cdot u - \mu_2 \cdot u|^2}{u^{\mathrm{T}}\Sigma_1 u + u^{\mathrm{T}}\Sigma_2 u}$$

We want to maximize S for the direction u. This is called the Fisher criterion. To maximize, we move the right-hand side denominator to the left-hand side of the expression. We obtain:

$$(u^{\mathrm{T}}\Sigma_1 u + u^{\mathrm{T}}\Sigma_2 u)S(u) = u^{\mathrm{T}}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^{\mathrm{T}}u$$

Differentiating

$$(2(\Sigma_1 + \Sigma_2)u)S(u) + (u^{\mathrm{T}}\Sigma_1 u + u^{\mathrm{T}}\Sigma_2 u)\frac{dS}{du} = 2(\mu_1 - \mu_2)(\mu_1 - \mu_2)^{\mathrm{T}}u$$

where we have used the fact the Σ_1 and Σ_2 are symmetric matrices.

Since we are maximizing, we set dS/du to zero, and we obtain:

$$((\Sigma_1 + \Sigma_2)u)S(u_0) = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^{\mathrm{T}}u$$

For given u, $(\mu_1 - \mu_2)^{\mathrm{T}}u$ is an scalar γ, and therefore

$$((\Sigma_1 + \Sigma_2)u)S(u_0) = \gamma(\mu_1 - \mu_2)$$

and finally

$$u = \frac{\gamma}{S(u_0)}(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2)$$

This direction of maximum separation (according to the Fisher criterion) is the celebrated Fisher linear discriminant. This approach allows us to project two multidimensional classes on a line and proceed finding a classifier for the simpler one-dimensional problem. Projecting multidimensional data sets on lines is a usual approach when trying to reduce the dimensionality of a problem, as is done, for example, in random trees [1]. The Fisher linear discriminant gives us a heuristic for a fast computation of a good projection direction.

References

[1] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer-Verlag, New York, 2001.