4 *Gaussian models*

4.1 Introduction

In this chapter, we discuss the **multivariate Gaussian** or **multivariate normal** (**MVN**), which is the most widely used joint probability density function for continuous variables. It will form the basis for many of the models we will encounter in later chapters.

Unfortunately, the level of mathematics in this chapter is higher than in many other chapters. In particular, we rely heavily on linear algebra and matrix calculus. This is the price one must pay in order to deal with high-dimensional data. Beginners may choose to skip sections marked with a *. In addition, since there are so many equations in this chapter, we have put a box around those that are particularly important.

4.1.1 Notation

Let us briefly say a few words about notation. We denote vectors by boldface lower case letters, such as $\mathbf{x}$. We denote matrices by boldface upper case letters, such as $\mathbf{X}$. We denote entries in a matrix by non-bold upper case letters, such as X_{ij}.

All vectors are assumed to be column vectors unless noted otherwise. We use $[x_1, \ldots, x_D]$ to denote a column vector created by stacking D scalars. Similarly, if we write $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_D]$, where the left hand side is a tall column vector, we mean to stack the $\mathbf{x}_i$ along the rows; this is usually written as $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_D^T)^T$, but that is rather ugly. If we write $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_D]$, where the left hand side is a matrix, we mean to stack the $\mathbf{x}_i$ along the columns, creating a matrix.

4.1.2 Basics

Recall from Section 2.5.2 that the pdf for an MVN in D dimensions is defined by the following:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] \tag{4.1}$$

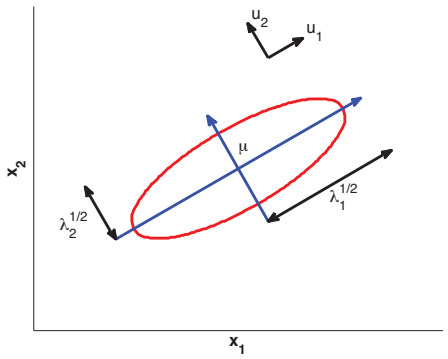


Figure 4.1 Visualization of a 2 dimensional Gaussian density. The major and minor axes of the ellipse are defined by the first two eigenvectors of the covariance matrix, namely $\mathbf{u}_1$ and $\mathbf{u}_2$. Based on Figure 2.7 of (Bishop 2006a).

The expression inside the exponent is the Mahalanobis distance between a data vector $\mathbf{x}$ and the mean vector $\boldsymbol{\mu}$, We can gain a better understanding of this quantity by performing an **eigendecomposition** of $\boldsymbol{\Sigma}$. That is, we write $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is an orthonormal matrix of eigenvectors satsifying $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues.

Using the eigendecomposition, we have that

$$\boldsymbol{\Sigma}^{-1} = \mathbf{U}^{-T}\boldsymbol{\Lambda}^{-1}\mathbf{U}^{-1} = \mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^T = \sum_{i=1}^{D}\frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^T \tag{4.2}$$

where $\mathbf{u}_i$ is the i'th column of $\mathbf{U}$, containing the i'th eigenvector. Hence we can rewrite the Mahalanobis distance as follows:

$$(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = (\mathbf{x}-\boldsymbol{\mu})^T\left(\sum_{i=1}^{D}\frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^T\right)(\mathbf{x}-\boldsymbol{\mu}) \tag{4.3}$$

$$= \sum_{i=1}^{D}\frac{1}{\lambda_i}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{u}_i\mathbf{u}_i^T(\mathbf{x}-\boldsymbol{\mu}) = \sum_{i=1}^{D}\frac{y_i^2}{\lambda_i} \tag{4.4}$$

where $y_i \triangleq \mathbf{u}_i^T(\mathbf{x}-\boldsymbol{\mu})$. Recall that the equation for an ellipse in 2d is

$$\frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} = 1 \tag{4.5}$$

Hence we see that the contours of equal probability density of a Gaussian lie along ellipses. This is illustrated in Figure 4.1. The eigenvectors determine the orientation of the ellipse, and the eigenvalues determine how elogonated it is.

In general, we see that the Mahalanobis distance corresponds to Euclidean distance in a transformed coordinate system, where we shift by $\boldsymbol{\mu}$ and rotate by $\mathbf{U}$.

4.1.3 MLE for an MVN

We now describe one way to estimate the parameters of an MVN, using MLE. In later sections, we will discuss Bayesian inference for the parameters, which can mitigate overfitting, and can provide a measure of confidence in our estimates.

Theorem 4.1.1 (MLE for a Gaussian)**.** *If we have N iid samples $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then the MLE for the parameters is given by*

$$\hat{\boldsymbol{\mu}}_{mle} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \triangleq \overline{\mathbf{x}} \tag{4.6}$$

$$\hat{\boldsymbol{\Sigma}}_{mle} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T = \frac{1}{N}(\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T) - \overline{\mathbf{x}}\,\overline{\mathbf{x}}^T \tag{4.7}$$

That is, the MLE is just the empirical mean and empirical covariance. In the univariate case, we get the following familiar results:

$$\hat{\mu} = \frac{1}{N}\sum_{i} x_i = \overline{x} \tag{4.8}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i}(x_i - \overline{x})^2 = \left(\frac{1}{N}\sum_{i} x_i^2\right) - (\overline{x})^2 \tag{4.9}$$

4.1.3.1 Proof *

To prove this result, we will need several results from matrix algebra, which we summarize below. In the equations, $\mathbf{a}$ and $\mathbf{b}$ are vectors, and $\mathbf{A}$ and $\mathbf{B}$ are matrices. Also, the notation $\mathrm{tr}(\mathbf{A})$ refers to the **trace** of a matrix, which is the sum of its diagonals: $\mathrm{tr}(\mathbf{A}) = \sum_i A_{ii}$.

$$\begin{aligned}
\frac{\partial(\mathbf{b}^T\mathbf{a})}{\partial\mathbf{a}} &= \mathbf{b} \\
\frac{\partial(\mathbf{a}^T\mathbf{A}\mathbf{a})}{\partial\mathbf{a}} &= (\mathbf{A} + \mathbf{A}^T)\mathbf{a} \\
\frac{\partial}{\partial\mathbf{A}}\mathrm{tr}(\mathbf{B}\mathbf{A}) &= \mathbf{B}^T \\
\frac{\partial}{\partial\mathbf{A}}\log|\mathbf{A}| &= \mathbf{A}^{-T} \triangleq (\mathbf{A}^{-1})^T \\
\mathrm{tr}(\mathbf{A}\mathbf{B}\mathbf{C}) &= \mathrm{tr}(\mathbf{C}\mathbf{A}\mathbf{B}) = \mathrm{tr}(\mathbf{B}\mathbf{C}\mathbf{A})
\end{aligned} \tag{4.10}$$

The last equation is called the **cyclic permutation property** of the trace operator. Using this, we can derive the widely used **trace trick**, which reorders the scalar inner product $\mathbf{x}^T\mathbf{A}\mathbf{x}$ as follows

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = \mathrm{tr}(\mathbf{x}^T\mathbf{A}\mathbf{x}) = \mathrm{tr}(\mathbf{x}\mathbf{x}^T\mathbf{A}) = \mathrm{tr}(\mathbf{A}\mathbf{x}\mathbf{x}^T) \tag{4.11}$$

Proof. We can now begin with the proof. The log-likelihood is

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{N}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^T\boldsymbol{\Lambda}(\mathbf{x}_i - \boldsymbol{\mu}) \tag{4.12}$$

where $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ is the precision matrix.

Using the substitution $\mathbf{y}_i = \mathbf{x}_i - \boldsymbol{\mu}$ and the chain rule of calculus, we have

$$\frac{\partial}{\partial \boldsymbol{\mu}}(\mathbf{x}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) = \frac{\partial}{\partial \mathbf{y}_i}\mathbf{y}_i^T\boldsymbol{\Sigma}^{-1}\mathbf{y}_i\frac{\partial \mathbf{y}_i}{\partial \boldsymbol{\mu}} \tag{4.13}$$

$$= -1(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-T})\mathbf{y}_i \tag{4.14}$$

Hence

$$\frac{\partial}{\partial \boldsymbol{\mu}}\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{i=1}^{N} -2\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-1}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu}) = 0 \tag{4.15}$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i = \overline{\mathbf{x}} \tag{4.16}$$

So the MLE of $\boldsymbol{\mu}$ is just the empirical mean.

Now we can use the trace-trick to rewrite the log-likelihood for $\boldsymbol{\Lambda}$ as follows:

$$\ell(\boldsymbol{\Lambda}) = \frac{N}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\sum_i \mathrm{tr}[(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T\boldsymbol{\Lambda}] \tag{4.17}$$

$$= \frac{N}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\mathrm{tr}\left[\mathbf{S}_\mu\boldsymbol{\Lambda}\right] \tag{4.18}$$

$$\tag{4.19}$$

where

$$\mathbf{S}_\mu \triangleq \sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \tag{4.20}$$

is the scatter matrix centered on $\boldsymbol{\mu}$. Taking derivatives of this expression with respect to $\boldsymbol{\Lambda}$ yields

$$\frac{\partial \ell(\boldsymbol{\Lambda})}{\partial \boldsymbol{\Lambda}} = \frac{N}{2}\boldsymbol{\Lambda}^{-T} - \frac{1}{2}\mathbf{S}_\mu^T = 0 \tag{4.21}$$

$$\boldsymbol{\Lambda}^{-T} = \boldsymbol{\Lambda}^{-1} = \boldsymbol{\Sigma} = \frac{1}{N}\mathbf{S}_\mu \tag{4.22}$$

so

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \tag{4.23}$$

which is just the empirical covariance matrix centered on $\boldsymbol{\mu}$. If we plug-in the MLE $\boldsymbol{\mu} = \overline{\mathbf{x}}$ (since both parameters must be simultaneously optimized), we get the standard equation for the MLE of a covariance matrix. □

4.1.4 Maximum entropy derivation of the Gaussian *

In this section, we show that the multivariate Gaussian is the distribution with maximum entropy subject to having a specified mean and covariance (see also Section 9.2.6). This is one reason the Gaussian is so widely used: the first two moments are usually all that we can reliably estimate from data, so we want a distribution that captures these properties, but otherwise makes as few addtional assumptions as possible.

To simplify notation, we will assume the mean is zero. The pdf has the form

$$p(\mathbf{x}) = \frac{1}{Z}\exp(-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}) \tag{4.24}$$

If we define $f_{ij}(\mathbf{x}) = x_i x_j$ and $\lambda_{ij} = \frac{1}{2}(\Sigma^{-1})_{ij}$, for $i, j \in \{1, \ldots, D\}$, we see that this is in the same form as Equation 9.74. The (differential) entropy of this distribution (using log base e) is given by

$$h(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \frac{1}{2}\ln\left[(2\pi e)^D|\boldsymbol{\Sigma}|\right] \tag{4.25}$$

We now show the MVN has maximum entropy amongst all distributions with a specified covariance $\boldsymbol{\Sigma}$.

Theorem 4.1.2. *Let $q(\mathbf{x})$ be any density satisfying $\int q(\mathbf{x})x_i x_j = \Sigma_{ij}$. Let $p = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Then $h(q) \leq h(p)$.*

Proof. (From (Cover and Thomas 1991, p234).) We have

$$\begin{aligned}
0 \quad &\leq \quad \mathbb{KL}\left(q||p\right) = \int q(\mathbf{x})\log\frac{q(\mathbf{x})}{p(\mathbf{x})}d\mathbf{x} && (4.26)\\
&= \quad -h(q) - \int q(\mathbf{x})\log p(\mathbf{x})d\mathbf{x} && (4.27)\\
&=^* \quad -h(q) - \int p(\mathbf{x})\log p(\mathbf{x})d\mathbf{x} && (4.28)\\
&= \quad -h(q) + h(p) && (4.29)
\end{aligned}$$

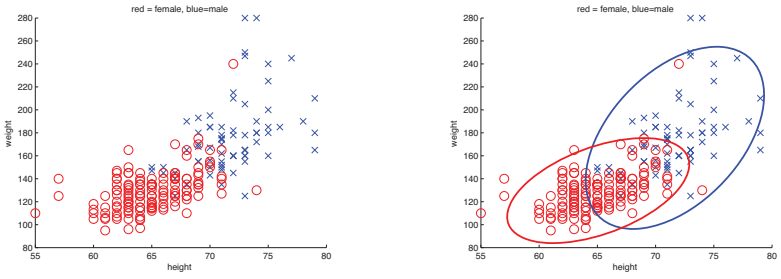
where the key step in Equation 4.28 (marked with a *) follows since q and p yield the same moments for the quadratic form encoded by $\log p(\mathbf{x})$. □

4.2 Gaussian discriminant analysis

One important application of MVNs is to define the the class conditional densities in a generative classifier, i.e.,

$$p(\mathbf{x}|y = c, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \tag{4.30}$$

The resulting technique is called (Gaussian) **discriminant analysis** or GDA (even though it is a generative, not discriminative, classifier — see Section 8.6 for more on this distinction). If $\boldsymbol{\Sigma}_c$ is diagonal, this is equivalent to naive Bayes.



(a) (b)

Figure 4.2 (a) Height/weight data. (b) Visualization of 2d Gaussians fit to each class. 95% of the probability mass is inside the ellipse. Figure generated by `gaussHeightWeight`.

We can classify a feature vector using the following decision rule, derived from Equation 2.13:

$$\hat{y}(\mathbf{x}) = \underset{c}{\operatorname{argmax}} \left[\log p(y=c|\boldsymbol{\pi}) + \log p(\mathbf{x}|\boldsymbol{\theta}_c)\right] \tag{4.31}$$

When we compute the probability of $\mathbf{x}$ under each class conditional density, we are measuring the distance from $\mathbf{x}$ to the center of each class, $\boldsymbol{\mu}_c$, using Mahalanobis distance. This can be thought of as a **nearest centroids classifier**.

As an example, Figure 4.2 shows two Gaussian class-conditional densities in 2d, representing the height and weight of men and women. We can see that the features are correlated, as is to be expected (tall people tend to weigh more). The ellipses for each class contain 95% of the probability mass. If we have a uniform prior over classes, we can classify a new test vector as follows:

$$\hat{y}(\mathbf{x}) = \underset{c}{\operatorname{argmin}} (\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (\mathbf{x} - \boldsymbol{\mu}_c) \tag{4.32}$$

4.2.1 Quadratic discriminant analysis (QDA)

The posterior over class labels is given by Equation 2.13. We can gain further insight into this model by plugging in the definition of the Gaussian density, as follows:

$$p(y=c|\mathbf{x},\boldsymbol{\theta}) = \frac{\pi_c |2\pi\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c)\right]}{\sum_{c'} \pi_{c'} |2\pi\boldsymbol{\Sigma}_{c'}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{c'})^T\boldsymbol{\Sigma}_{c'}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{c'})\right]} \tag{4.33}$$

Thresholding this results in a quadratic function of $\mathbf{x}$. The result is known as **quadratic discriminant analysis** (QDA). Figure 4.3 gives some examples of what the decision boundaries look like in 2D.

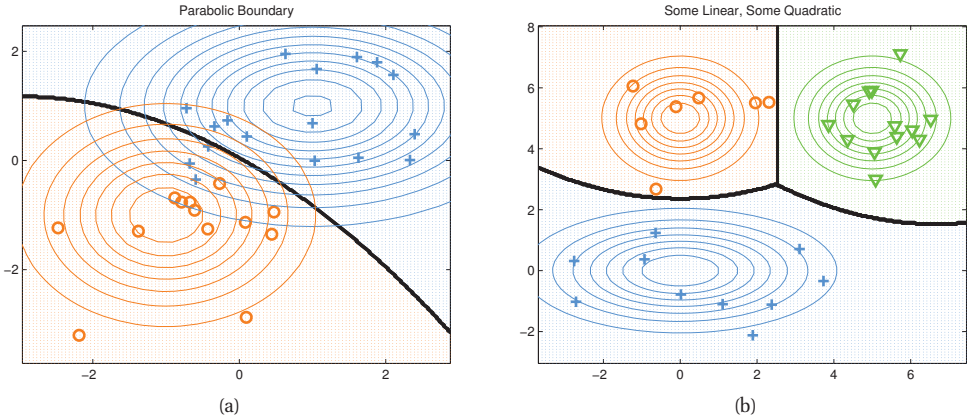


Figure 4.3 Quadratic decision boundaries in 2D for the 2 and 3 class case. Figure generated by `discrimAnalysisDboundariesDemo`.

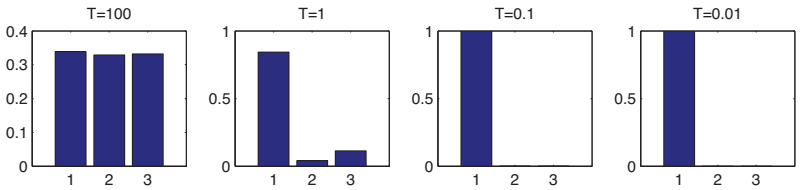


Figure 4.4 Softmax distribution $\mathcal{S}(\boldsymbol{\eta}/T)$, where $\boldsymbol{\eta} = (3, 0, 1)$, at different temperatures T. When the temperature is high (left), the distribution is uniform, whereas when the temperature is low (right), the distribution is "spiky", with all its mass on the largest element. Figure generated by `softmaxDemo2`.

4.2.2 Linear discriminant analysis (LDA)

We now consider a special case in which the covariance matrices are **tied** or **shared** across classes, $\boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}$. In this case, we can simplify Equation 4.33 as follows:

$$p(y=c|\mathbf{x},\boldsymbol{\theta}) \propto \pi_c \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\right] \tag{4.34}$$

$$= \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c\right]\exp[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}] \tag{4.35}$$

Since the quadratic term $\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}$ is independent of c, it will cancel out in the numerator and denominator. If we define

$$\gamma_c = -\frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c \tag{4.36}$$

$$\boldsymbol{\beta}_c = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c \tag{4.37}$$

then we can write

$$p(y=c|\mathbf{x},\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_c^T\mathbf{x}+\gamma_c}}{\sum_{c'} e^{\boldsymbol{\beta}_{c'}^T\mathbf{x}+\gamma_{c'}}} = \mathcal{S}(\boldsymbol{\eta})_c \tag{4.38}$$

where $\boldsymbol{\eta} = [\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1, \ldots, \boldsymbol{\beta}_C^T\mathbf{x}+\gamma_C]$, and $\mathcal{S}$ is the **softmax** function, defined as follows:

$$\mathcal{S}(\boldsymbol{\eta})_c = \frac{e^{\eta_c}}{\sum_{c'=1}^{C} e^{\eta_{c'}}} \tag{4.39}$$

The softmax function is so-called since it acts a bit like the max function. To see this, let us divide each η_c by a constant T called the **temperature**. Then as $T \to 0$, we find

$$\mathcal{S}(\boldsymbol{\eta}/T)_c = \begin{cases} 1.0 & \text{if } c = \text{argmax}_{c'}\, \eta_{c'} \\ 0.0 & \text{otherwise} \end{cases} \tag{4.40}$$

In other words, at low temperatures, the distribution spends essentially all of its time in the most probable state, whereas at high temperatures, it visits all states uniformly. See Figure 4.4 for an illustration. Note that this terminology comes from the area of statistical physics, where it is common to use the **Boltzmann distribution**, which has the same form as the softmax function.

An interesting property of Equation 4.38 is that, if we take logs, we end up with a linear function of $\mathbf{x}$. (The reason it is linear is because the $\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}$ cancels from the numerator and denominator.) Thus the decision boundary between any two classes, say c and c', will be a straight line. Hence this technique is called **linear discriminant analysis** or **LDA**. [1] We can derive the form of this line as follows:

$$p(y=c|\mathbf{x},\boldsymbol{\theta}) = p(y=c'|\mathbf{x},\boldsymbol{\theta}) \tag{4.41}$$

$$\boldsymbol{\beta}_c^T\mathbf{x}+\gamma_c = \boldsymbol{\beta}_{c'}^T\mathbf{x}+\gamma_{c'} \tag{4.42}$$

$$\mathbf{x}^T(\boldsymbol{\beta}_{c'}-\boldsymbol{\beta}) = \gamma_{c'}-\gamma_c \tag{4.43}$$

See Figure 4.5 for some examples.

An alternative to fitting an LDA model and then deriving the class posterior is to directly fit $p(y|\mathbf{x},\mathbf{W}) = \text{Cat}(y|\mathbf{W}\mathbf{x})$ for some $C \times D$ weight matrix $\mathbf{W}$. This is called **multi-class logistic regression**, or **multinomial logistic regression**.[2] We will discuss this model in detail in Section 8.2. The difference between the two approaches is explained in Section 8.6.

4.2.3 Two-class LDA

To gain further insight into the meaning of these equations, let us consider the binary case. In this case, the posterior is given by

$$p(y=1|\mathbf{x},\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1}}{e^{\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1} + e^{\boldsymbol{\beta}_0^T\mathbf{x}+\gamma_0}} \tag{4.44}$$

$$= \frac{1}{1+e^{(\boldsymbol{\beta}_0-\boldsymbol{\beta}_1)^T\mathbf{x}+(\gamma_0-\gamma_1)}} = \text{sigm}\left((\boldsymbol{\beta}_1-\boldsymbol{\beta}_0)^T\mathbf{x}+(\gamma_1-\gamma_0)\right) \tag{4.45}$$

1. The abbreviation "LDA", could either stand for "linear discriminant analysis" or "latent Dirichlet allocation" (Section 27.3). We hope the meaning is clear from text.

2. In the language modeling community, this model is called a **maximum entropy** model, for reasons explained in Section 9.2.6.

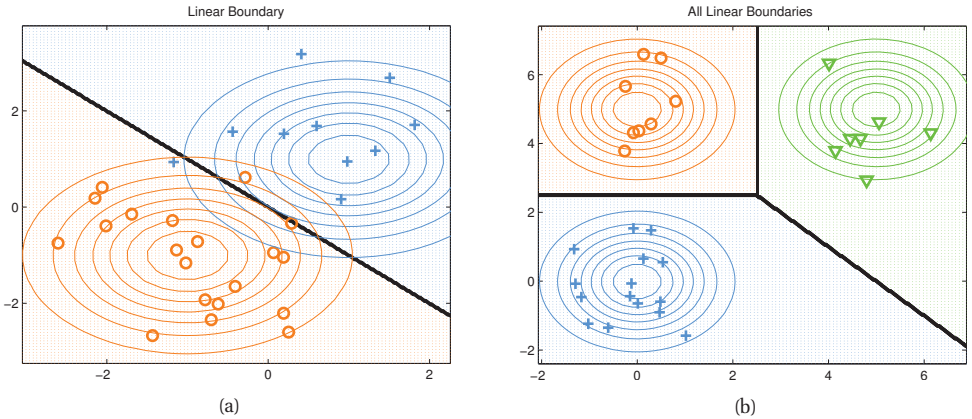


Figure 4.5 Linear decision boundaries in 2D for the 2 and 3 class case. Figure generated by `discrimAnalysisDboundariesDemo`.

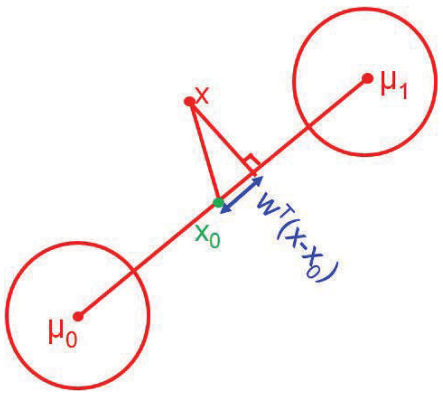


Figure 4.6 Geometry of LDA in the 2 class case where $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbf{I}$.

where $\text{sigm}(\eta)$ refers to the sigmoid function (Equation 1.10).

Now

$$\gamma_1 - \gamma_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 + \log(\pi_1/\pi_0) \tag{4.46}$$

$$= -\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) + \log(\pi_1/\pi_0) \tag{4.47}$$

So if we define

$$\mathbf{w} = \boldsymbol{\beta}_1 - \boldsymbol{\beta}_0 = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \tag{4.48}$$

$$\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\frac{\log(\pi_1/\pi_0)}{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)} \tag{4.49}$$

then we have $\mathbf{w}^T\mathbf{x}_0 = -(\gamma_1 - \gamma_0)$, and hence

$$p(y=1|\mathbf{x},\boldsymbol{\theta}) \quad = \quad \text{sigm}(\mathbf{w}^T(\mathbf{x}-\mathbf{x}_0)) \tag{4.50}$$

(This is closely related to logistic regression, which we will discuss in Section 8.2.) So the final decision rule is as follows: shift $\mathbf{x}$ by $\mathbf{x}_0$, project onto the line $\mathbf{w}$, and see if the result is positive or negative.

If $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, then $\mathbf{w}$ is in the direction of $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. So we classify the point based on whether its projection is closer to $\boldsymbol{\mu}_0$ or $\boldsymbol{\mu}_1$. This is illustrated in Figure 4.6. Furthemore, if $\pi_1 = \pi_0$, then $\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)$, which is half way between the means. If we make $\pi_1 > \pi_0$, then $\mathbf{x}_0$ gets closer to $\boldsymbol{\mu}_0$, so more of the line belongs to class 1 *a priori*. Conversely if $\pi_1 < \pi_0$, the boundary shifts right. Thus we see that the class prior, π_c, just changes the decision threshold, and not the overall geometry, as we claimed above. (A similar argument applies in the multi-class case.)

The magnitude of $\mathbf{w}$ determines the steepness of the logistic function, and depends on how well-separated the means are, relative to the variance. In psychology and signal detection theory, it is common to define the **discriminability** of a signal from the background noise using a quantity called **d-prime**:

$$d' \triangleq \frac{\mu_1 - \mu_0}{\sigma} \tag{4.51}$$

where μ_1 is the mean of the signal and μ_0 is the mean of the noise, and σ is the standard deviation of the noise. If d' is large, the signal will be easier to discriminate from the noise.

4.2.4 MLE for discriminant analysis

We now discuss how to fit a discriminant analysis model. The simplest way is to use maximum likelihood. The log-likelihood function is as follows:

$$\log p(\mathcal{D}|\boldsymbol{\theta}) \quad = \quad \left[\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbb{I}(y_i = c)\log\pi_c\right] + \sum_{c=1}^{C}\left[\sum_{i:y_i=c}\log\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c)\right] \tag{4.52}$$

We see that this factorizes into a term for $\boldsymbol{\pi}$, and C terms for each $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$. Hence we can estimate these parameters separately. For the class prior, we have $\hat{\pi}_c = \frac{N_c}{N}$, as with naive Bayes. For the class-conditional densities, we just partition the data based on its class label, and compute the MLE for each Gaussian:

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{N_c}\sum_{i:y_i=c}\mathbf{x}_i, \quad \hat{\boldsymbol{\Sigma}}_c = \frac{1}{N_c}\sum_{i:y_i=c}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^T \tag{4.53}$$

See `discrimAnalysisFit` for a Matlab implementation. Once the model has been fit, you can make predictions using `discrimAnalysisPredict`, which uses a plug-in approximation.

4.2.5 Strategies for preventing overfitting

The speed and simplicity of the MLE method is one of its greatest appeals. However, the MLE can badly overfit in high dimensions. In particular, the MLE for a full covariance matrix is singular if $N_c < D$. And even when $N_c > D$, the MLE can be ill-conditioned, meaning it is close to singular. There are several possible solutions to this problem:

- Use a diagonal covariance matrix for each class, which assumes the features are conditionally independent; this is equivalent to using a naive Bayes classifier (Section 3.5).
- Use a full covariance matrix, but force it to be the same for all classes, $\boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}$. This is an example of **parameter tying** or **parameter sharing**, and is equivalent to LDA (Section 4.2.2).
- Use a diagonal covariance matrix *and* forced it to be shared. This is called diagonal covariance LDA, and is discussed in Section 4.2.7.
- Use a full covariance matrix, but impose a prior and then integrate it out. If we use a conjugate prior, this can be done in closed form, using the results from Section 4.6.3; this is analogous to the "Bayesian naive Bayes" method in Section 3.5.1.2. See (Minka 2000f) for details.
- Fit a full or diagonal covariance matrix by MAP estimation. We discuss two different kinds of prior below.
- Project the data into a low dimensional subspace and fit the Gaussians there. See Section 8.6.3.3 for a way to find the best (most discriminative) linear projection.

We discuss some of these options below.

4.2.6 Regularized LDA *

Suppose we tie the covariance matrices, so $\boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}$, as in LDA, and furthermore we perform MAP estimation of $\boldsymbol{\Sigma}$ using an inverse Wishart prior of the form $\text{IW}(\text{diag}(\hat{\boldsymbol{\Sigma}}_{mle}), \nu_0)$ (see Section 4.5.1). Then we have

$$\hat{\boldsymbol{\Sigma}} = \lambda \text{diag}(\hat{\boldsymbol{\Sigma}}_{mle}) + (1-\lambda)\hat{\boldsymbol{\Sigma}}_{mle} \tag{4.54}$$

where λ controls the amount of regularization, which is related to the strength of the prior, ν_0 (see Section 4.6.2.1 for details). This technique is known as **regularized discriminant analysis** or RDA (Hastie et al. 2009, p656).

When we evaluate the class conditional densities, we need to compute $\hat{\boldsymbol{\Sigma}}^{-1}$, and hence $\hat{\boldsymbol{\Sigma}}_{mle}^{-1}$, which is impossible to compute if $D > N$. However, we can use the SVD of $\mathbf{X}$ (Section 12.2.3) to get around this, as we show below. (Note that this trick cannot be applied to QDA, which is a nonlinear function of $\mathbf{x}$.)

Let $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ be the SVD of the design matrix, where $\mathbf{V}$ is $D \times N$, $\mathbf{U}$ is an $N \times N$ orthogonal matrix, and $\mathbf{D}$ is a diagonal matrix of size N. Furthermore, define the $N \times N$ matrix $\mathbf{Z} = \mathbf{U}\mathbf{D}$; this is like a design matrix in a lower dimensional space (since we assume $N < D$). Also, define $\boldsymbol{\mu}_z = \mathbf{V}^T\boldsymbol{\mu}$ as the mean of the data in this reduced space; we can recover the original mean using $\boldsymbol{\mu} = \mathbf{V}\boldsymbol{\mu}_z$, since $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$. With these definitions, we can

rewrite the MLE as follows:

$$\begin{aligned}
\hat{\boldsymbol{\Sigma}}_{mle} &= \frac{1}{N}\mathbf{X}^T\mathbf{X} - \boldsymbol{\mu}\boldsymbol{\mu}^T && (4.55)\\
&= \frac{1}{N}(\mathbf{Z}\mathbf{V}^T)^T(\mathbf{Z}\mathbf{V}^T) - (\mathbf{V}\boldsymbol{\mu}_z)(\mathbf{V}\boldsymbol{\mu}_z)^T && (4.56)\\
&= \frac{1}{N}\mathbf{V}\mathbf{Z}^T\mathbf{Z}\mathbf{V}^T - \mathbf{V}\boldsymbol{\mu}_z\boldsymbol{\mu}_z^T\mathbf{V}^T && (4.57)\\
&= \mathbf{V}(\frac{1}{N}\mathbf{Z}^T\mathbf{Z} - \boldsymbol{\mu}_z\boldsymbol{\mu}_z^T)\mathbf{V}^T && (4.58)\\
&= \mathbf{V}\hat{\boldsymbol{\Sigma}}_z\mathbf{V}^T && (4.59)
\end{aligned}$$

where $\hat{\boldsymbol{\Sigma}}_z$ is the empirical covariance of $\mathbf{Z}$. Hence we can rewrite the MAP estimate as

$$\begin{aligned}
\hat{\boldsymbol{\Sigma}}_{map} &= \mathbf{V}\tilde{\boldsymbol{\Sigma}}_z\mathbf{V}^T && (4.60)\\
\tilde{\boldsymbol{\Sigma}}_z &= \lambda\mathrm{diag}(\hat{\boldsymbol{\Sigma}}_z) + (1-\lambda)\hat{\boldsymbol{\Sigma}}_z && (4.61)
\end{aligned}$$

Note, however, that we never need to actually compute the $D \times D$ matrix $\hat{\boldsymbol{\Sigma}}_{map}$. This is because Equation 4.38 tells us that to classify using LDA, all we need to compute is $p(y = c|\mathbf{x}, \boldsymbol{\theta}) \propto \exp(\delta_c)$, where

$$\delta_c = -\mathbf{x}^T\boldsymbol{\beta}_c + \gamma_c, \quad \boldsymbol{\beta}_c = \hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\mu}_c, \quad \gamma_c - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\beta}_c + \log\pi_c \tag{4.62}$$

We can compute the crucial $\boldsymbol{\beta}_c$ term for RDA without inverting the $D \times D$ matrix as follows:

$$\boldsymbol{\beta}_c = \hat{\boldsymbol{\Sigma}}_{map}^{-1}\boldsymbol{\mu}_c = (\mathbf{V}\tilde{\boldsymbol{\Sigma}}_z\mathbf{V}^T)^{-1}\boldsymbol{\mu}_c = \mathbf{V}\tilde{\boldsymbol{\Sigma}}_z^{-1}\mathbf{V}^T\boldsymbol{\mu}_c = \mathbf{V}\tilde{\boldsymbol{\Sigma}}_z^{-1}\boldsymbol{\mu}_{z,c} \tag{4.63}$$

where $\boldsymbol{\mu}_{z,c} = \mathbf{V}^T\boldsymbol{\mu}_c$ is the mean of the $\mathbf{Z}$ matrix for data belonging to class c. See `rdaFit` for the code.

4.2.7 Diagonal LDA

A simple alternative to RDA is to tie the covariance matrices, so $\boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}$ as in LDA, and then to use a diagonal covariance matrix for each class. This is called the **diagonal LDA** model, and is equivalent to RDA with $\lambda = 1$. The corresponding discriminant function is as follows (compare to Equation 4.33):

$$\delta_c(\mathbf{x}) = \log p(\mathbf{x}, y = c|\boldsymbol{\theta}) = -\sum_{j=1}^{D}\frac{(x_j - \mu_{cj})^2}{2\sigma_j^2} + \log\pi_c \tag{4.64}$$

Typically we set $\hat{\mu}_{cj} = \overline{x}_{cj}$ and $\hat{\sigma}_j^2 = s_j^2$, which is the **pooled empirical variance** of feature j (pooled across classes) defined by

$$s_j^2 = \frac{\sum_{c=1}^{C}\sum_{i:y_i=c}(x_{ij} - \overline{x}_{cj})^2}{N - C} \tag{4.65}$$

In high dimensional settings, this model can work much better than LDA and RDA (Bickel and Levina 2004).

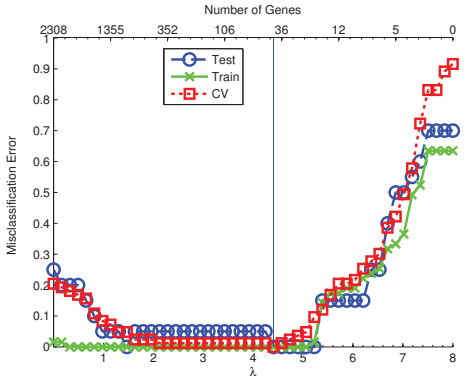


Figure 4.7 Error versus amount of shrinkage for nearest shrunken centroid classifier applied to the SRBCT gene expression data. Based on Figure 18.4 of (Hastie et al. 2009). Figure generated by `shrunkenCentroidsSRBCTdemo`.

4.2.8 Nearest shrunken centroids classifier *

One drawback of diagonal LDA is that it depends on all of the features. In high dimensional problems, we might prefer a method that only depends on a subset of the features, for reasons of accuracy and interpretability. One approach is to use a screening method, perhaps based on mutual information, as in Section 3.5.4. We now discuss another approach to this problem known as the **nearest shrunken centroids** classifier (Hastie et al. 2009, p652).

The basic idea is to perform MAP estimation for diagonal LDA with a sparsity-promoting (Laplace) prior (see Section 13.3). More precisely, define the class-specific feature mean, μ_{cj}, in terms of the class-independent feature mean, m_j, and a class-specific offset, Δ_{cj}. Thus we have

$$\mu_{cj} = m_j + \Delta_{cj} \tag{4.66}$$

We will then put a prior on the Δ_{cj} terms to encourage them to be strictly zero and compute a MAP estimate. If, for feature j, we find that $\Delta_{cj} = 0$ for all c, then feature j will play no role in the classification decision (since μ_{cj} will be independent of c). Thus features that are not discriminative are automatically ignored. The details can be found in (Hastie et al. 2009, p652) and (Greenshtein and Park 2009). See `shrunkenCentroidsFit` for some code.

Let us give an example of the method in action, based on (Hastie et al. 2009, p652). Consider the problem of classifying a gene expression dataset, which 2308 genes, 4 classes, 63 training samples and 20 test samples. Using a diagonal LDA classifier produces 5 errors on the test set. Using the nearest shrunken centroids classifier produced 0 errors on the test set, for a range of λ values: see Figure 4.7. More importantly, the model is sparse and hence more interpretable: Figure 4.8 plots an unpenalized estimate of the difference, d_{cj}, in gray, as well as the shrunken estimates Δ_{cj} in blue. (These estimates are computed using the value of λ estimated by CV.) We see that only 39 genes are used, out of the original 2308.

Now consider an even harder problem, with 16,603 genes, a training set of 144 patients, a test set of 54 patients, and 14 different types of cancer (Ramaswamy et al. 2001). Hastie et al. (Hastie et al. 2009, p656) report that nearest shrunken centroids produced 17 errors on the test

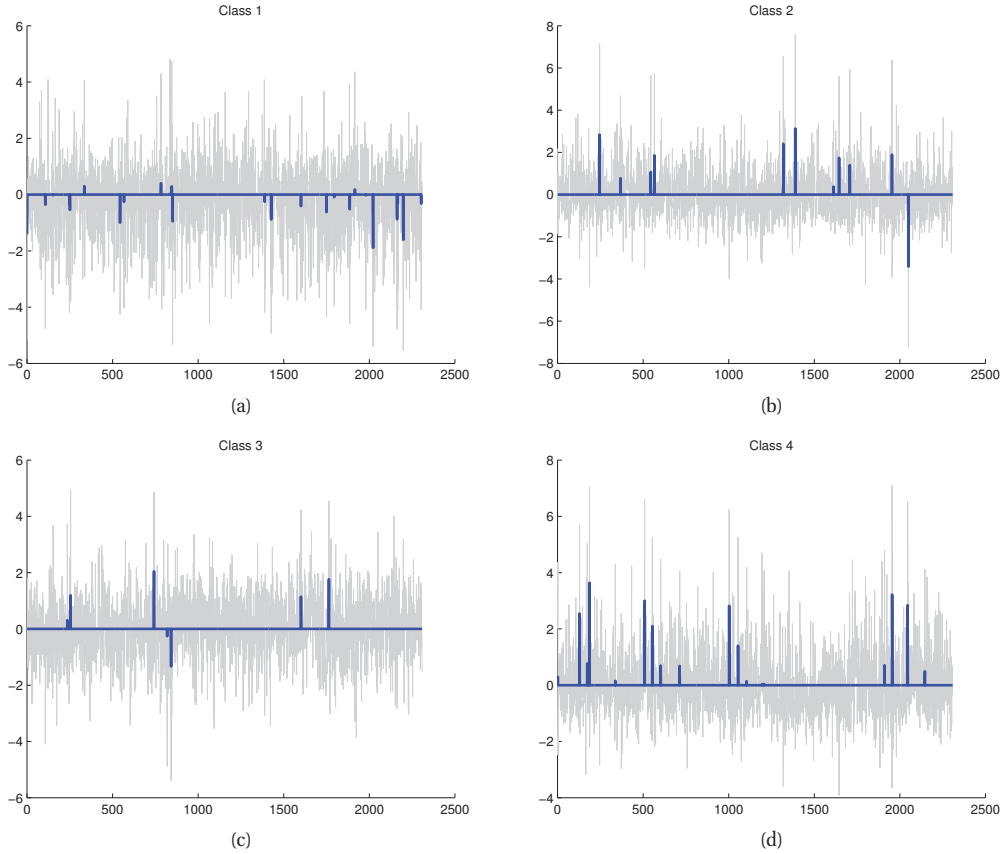


Figure 4.8 Profile of the shrunken centroids corresponding to $\lambda = 4.4$ (CV optimal in Figure 4.7). This selects 39 genes. Based on Figure 18.4 of (Hastie et al. 2009). Figure generated by `shrunkenCentroidsSRBCTdemo`.

set, using 6,520 genes, and that RDA (Section 4.2.6) produced 12 errors on the test set, using all 16,603 genes. The PMTK function `cancerHighDimClassifDemo` can be used to reproduce these numbers.

4.3 Inference in jointly Gaussian distributions

Given a joint distribution, $p(\mathbf{x}_1, \mathbf{x}_2)$, it is useful to be able to compute marginals $p(\mathbf{x}_1)$ and conditionals $p(\mathbf{x}_1|\mathbf{x}_2)$. We discuss how to do this below, and then give some applications. These operations take $O(D^3)$ time in the worst case. See Section 20.4.3 for faster methods.

4.3.1 Statement of the result

Theorem 4.3.1 (Marginals and conditionals of an MVN)**.** *Suppose* $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ *is jointly Gaussian with parameters*

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix} \tag{4.67}$$

Then the marginals are given by

$$\begin{aligned} p(\mathbf{x}_1) &= \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ p(\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_2|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \end{aligned} \tag{4.68}$$

and the posterior conditional is given by

$$\begin{aligned} p(\mathbf{x}_1|\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\Sigma}_{1|2}\left(\boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2)\right) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1} \end{aligned} \tag{4.69}$$

Equation 4.69 is of such crucial importance in this book that we have put a box around it, so you can easily find it. For the proof, see Section 4.3.4.

We see that both the marginal and conditional distributions are themselves Gaussian. For the marginals, we just extract the rows and columns corresponding to $\mathbf{x}_1$ or $\mathbf{x}_2$. For the conditional, we have to do a bit more work. However, it is not that complicated: the conditional mean is just a linear function of $\mathbf{x}_2$, and the conditional covariance is just a constant matrix that is independent of $\mathbf{x}_2$. We give three different (but equivalent) expressions for the posterior mean, and two different (but equivalent) expressions for the posterior covariance; each one is useful in different circumstances.

4.3.2 Examples

Below we give some examples of these equations in action, which will make them seem more intuitive.

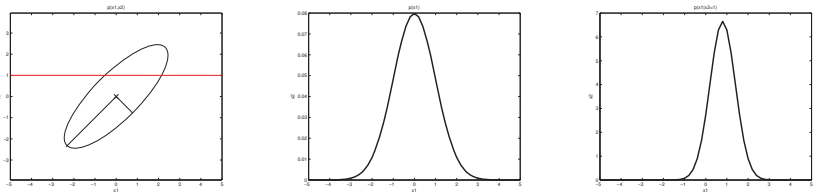
4.3.2.1 Marginals and conditionals of a 2d Gaussian

Let us consider a 2d example. The covariance matrix is

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{4.70}$$

The marginal $p(x_1)$ is a 1D Gaussian, obtained by projecting the joint distribution onto the x_1 line:

$$p(x_1) = \mathcal{N}(x_1|\mu_1, \sigma_1^2) \tag{4.71}$$



(a) (b) (c)

Figure 4.9 (a) A joint Gaussian distribution $p(x_1, x_2)$ with a correlation coefficient of 0.8. We plot the 95% contour and the principal axes. (b) The unconditional marginal $p(x_1)$. (c) The conditional $p(x_1|x_2) = \mathcal{N}(x_1|0.8, 0.36)$, obtained by slicing (a) at height $x_2 = 1$. Figure generated by `gaussCondition2Ddemo2`.

Suppose we observe $X_2 = x_2$; the conditional $p(x_1|x_2)$ is obtained by "slicing" the joint distribution through the $X_2 = x_2$ line (see Figure 4.9):

$$p(x_1|x_2) = \mathcal{N}\left(x_1|\mu_1 + \frac{\rho\sigma_1\sigma_2}{\sigma_2^2}(x_2 - \mu_2),\ \sigma_1^2 - \frac{(\rho\sigma_1\sigma_2)^2}{\sigma_2^2}\right) \tag{4.72}$$

If $\sigma_1 = \sigma_2 = \sigma$, we get

$$p(x_1|x_2) = \mathcal{N}\left(x_1|\mu_1 + \rho(x_2 - \mu_2),\ \sigma^2(1 - \rho^2)\right) \tag{4.73}$$

In Figure 4.9 we show an example where $\rho = 0.8$, $\sigma_1 = \sigma_2 = 1$, $\boldsymbol{\mu} = \mathbf{0}$ and $x_2 = 1$. We see that $\mathbb{E}[x_1|x_2 = 1] = 0.8$, which makes sense, since $\rho = 0.8$ means that we believe that if x_2 increases by 1 (beyond its mean), then x_1 increases by 0.8. We also see $\text{var}[x_1|x_2 = 1] = 1 - 0.8^2 = 0.36$. This also makes sense: our uncertainty about x_1 has gone down, since we have learned something about x_1 (indirectly) by observing x_2. If $\rho = 0$, we get $p(x_1|x_2) = \mathcal{N}\left(x_1|\mu_1,\ \sigma_1^2\right)$, since x_2 conveys no information about x_1 if they are uncorrelated (and hence independent).

4.3.2.2 Interpolating noise-free data

Suppose we want to estimate a 1d function, defined on the interval $[0, T]$, such that $y_i = f(t_i)$ for N observed points t_i. We assume for now that the data is noise-free, so we want to **interpolate** it, that is, fit a function that goes exactly through the data. (See Section 4.4.2.3 for the noisy data case.) The question is: how does the function behave in between the observed data points? It is often reasonable to assume that the unknown function is smooth. In Chapter 15, we shall see how to encode *priors over functions*, and how to update such a prior with observed values to get a posterior over functions. But in this section, we take a simpler approach, which is adequate for MAP estimation of functions defined on 1d inputs. We follow the presentation of (Calvetti and Somersalo 2007, p135).

We start by discretizing the problem. First we divide the support of the function into D equal subintervals. We then define

$$x_j = f(s_j), \quad s_j = jh, \quad h = \frac{T}{D},\ 1 \le j \le D \tag{4.74}$$

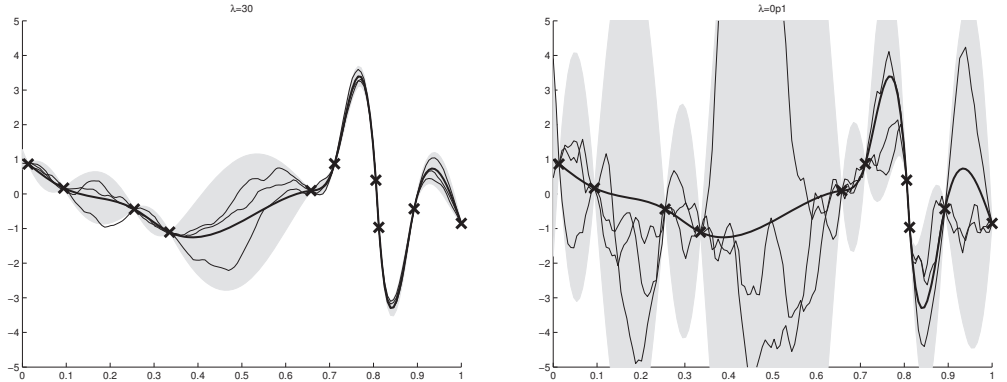


Figure 4.10 Interpolating noise-free data using a Gaussian with prior precision λ. (a) $\lambda = 30$. (b) $\lambda = 0.01$. See also Figure 4.15. Based on Figure 7.1 of (Calvetti and Somersalo 2007). Figure generated by `gaussInterpDemo`.

We can encode our smoothness prior by assuming that x_j is an average of its neighbors, x_{j-1} and x_{j+1}, plus some Gaussian noise:

$$x_j = \frac{1}{2}(x_{j-1} + x_{j+1}) + \epsilon_j, \quad 2 \leq j \leq D-2 \tag{4.75}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, (1/\lambda)\mathbf{I})$. The precision term λ controls how much we think the function will vary: a large λ corresponds to a belief that the function is very smooth, a small λ corresponds to a belief that the function is quite "wiggly". In vector form, the above equation can be written as follows:

$$\mathbf{L}\mathbf{x} = \boldsymbol{\epsilon} \tag{4.76}$$

where $\mathbf{L}$ is the $(D-2) \times D$ second order **finite difference matrix**

$$\mathbf{L} = \frac{1}{2}\begin{pmatrix} -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & & & \\ & & & -1 & 2 & -1 \end{pmatrix} \tag{4.77}$$

The corresponding prior has the form

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, (\lambda^2 \mathbf{L}^T\mathbf{L})^{-1}) \propto \exp\left(-\frac{\lambda^2}{2}||\mathbf{L}\mathbf{x}||_2^2\right) \tag{4.78}$$

We will henceforth assume we have scaled $\mathbf{L}$ by λ so we can ignore the λ term, and just write $\boldsymbol{\Lambda} = \mathbf{L}^T\mathbf{L}$ for the precision matrix.

Note that although $\mathbf{x}$ is D-dimensional, the precision matrix $\boldsymbol{\Lambda}$ only has rank $D-2$. Thus this is an improper prior, known as an intrinsic Gaussian random field (see Section 19.4.4 for

more information). However, providing we observe $N \geq 2$ data points, the posterior will be proper.

Now let $\mathbf{x}_2$ be the N noise-free observations of the function, and $\mathbf{x}_1$ be the $D-N$ unknown function values. Without loss of generality, assume that the unknown variables are ordered first, then the known variables. Then we can partition the $\mathbf{L}$ matrix as follows:

$$\mathbf{L} = [\mathbf{L}_1,\ \mathbf{L}_2],\ \mathbf{L}_1 \in \mathbb{R}^{(D-2)\times(D-N)},\ \mathbf{L}_2 \in \mathbb{R}^{(D-2)\times(N)} \tag{4.79}$$

We can also partition the precision matrix of the joint distribution:

$$\mathbf{\Lambda} = \mathbf{L}^T\mathbf{L} = \begin{pmatrix} \mathbf{\Lambda}_{11} & \mathbf{\Lambda}_{12} \\ \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{L}_1^T\mathbf{L}_1 & \mathbf{L}_1^T\mathbf{L}_2 \\ \mathbf{L}_2^T\mathbf{L}_1 & \mathbf{L}_2^T\mathbf{L}_2 \end{pmatrix} \tag{4.80}$$

Using Equation 4.69, we can write the conditional distribution as follows:

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\boldsymbol{\mu}_{1|2}, \mathbf{\Sigma}_{1|2}) \tag{4.81}$$

$$\boldsymbol{\mu}_{1|2} = -\mathbf{\Lambda}_{11}^{-1}\mathbf{\Lambda}_{12}\mathbf{x}_2 = -\mathbf{L}_1^T\mathbf{L}_2\mathbf{x}_2 \tag{4.82}$$

$$\Sigma_{1|2} = \mathbf{\Lambda}_{11}^{-1} \tag{4.83}$$

Note that we can compute the mean by solving the following system of linear equations:

$$\mathbf{L}_1\boldsymbol{\mu}_{1|2} = -\mathbf{L}_2\mathbf{x}_2 \tag{4.84}$$

This is efficient since $\mathbf{L}_1$ is tridiagonal. Figure 4.10 gives an illustration of these equations. We see that the posterior mean $\boldsymbol{\mu}_{1|2}$ equals the observed data at the specified points, and smoothly interpolates in between, as desired.

It is also interesting to plot the 95% **pointwise marginal credibility intervals**, $\mu_j \pm 2\sqrt{\Sigma_{1|2,jj}}$, shown in grey. We see that the variance goes up as we move away from the data. We also see that the variance goes up as we decrease the precision of the prior, λ. Interestingly, λ has no effect on the posterior mean, since it cancels out when multiplying $\mathbf{\Lambda}_{11}$ and $\mathbf{\Lambda}_{12}$. By contrast, when we consider noisy data in Section 4.4.2.3, we will see that the prior precision affects the smoothness of posterior mean estimate.

The marginal credibility intervals do not capture the fact that neighboring locations are correlated. We can represent that by drawing complete functions (i.e., vectors $\mathbf{x}$) from the posterior, and plotting them. These are shown by the thin lines in Figure 4.10. These are not quite as smooth as the posterior mean itself. This is because the prior only penalizes first-order differences. See Section 4.4.2.3 for further discussion of this point.

4.3.2.3 Data imputation

Suppose we are missing some entries in a design matrix. If the columns are correlated, we can use the observed entries to predict the missing entries. Figure 4.11 shows a simple example. We sampled some data from a 20 dimensional Gaussian, and then deliberately "hid" 50% of the data in each row. We then inferred the missing entries given the observed entries, using the true (generating) model. More precisely, for each row i, we compute $p(\mathbf{x}_{\mathbf{h}_i}|\mathbf{x}_{\mathbf{v}_i}, \boldsymbol{\theta})$, where $\mathbf{h}_i$ and $\mathbf{v}_i$ are the indices of the hidden and visible entries in case i. From this, we compute the marginal distribution of each missing variable, $p(x_{h_{ij}}|\mathbf{x}_{\mathbf{v}_i}, \boldsymbol{\theta})$. We then plot the mean of this distribution, $\hat{x}_{ij} = \mathbb{E}[x_j|\mathbf{x}_{\mathbf{v}_i}, \boldsymbol{\theta}]$; this represents our "best guess" about the true value of that entry, in the

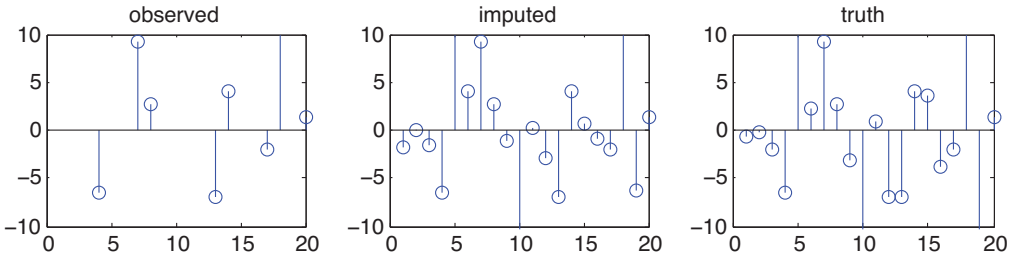


Figure 4.11 Illustration of data imputation. Left column: visualization of three rows of the data matrix with missing entries. Middle column: mean of the posterior predictive, based on partially observed data in that row, but the true model parameters. Right column: true values. Figure generated by `gaussImputationDemo`.

sense that it minimizes our expected squared error (see Section 5.7 for details). Figure 4.11 shows that the estimates are quite close to the truth. (Of course, if $j \in \mathbf{v}_i$, the expected value is equal to the observed value, $\hat{x}_{ij} = x_{ij}$.)

We can use $\text{var}\left[x_{h_{ij}}|\mathbf{x}_{\mathbf{v}_i}, \boldsymbol{\theta}\right]$ as a measure of confidence in this guess, although this is not shown. Alternatively, we could draw multiple samples from $p(\mathbf{x}_{\mathbf{h}_i}|\mathbf{x}_{\mathbf{v}_i}, \boldsymbol{\theta})$; this is called **multiple imputation**.

In addition to imputing the missing entries, we may be interested in computing the likelihood of each partially observed row in the table, $p(\mathbf{x}_{\mathbf{v}_i}|\boldsymbol{\theta})$, which can be computed using Equation 4.68. This is useful for detecting outliers (atypical observations).

4.3.3 Information form

Suppose $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. One can show that $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$ is the mean vector, and $\text{cov}[\mathbf{x}] = \boldsymbol{\Sigma}$ is the covariance matrix. These are called the **moment parameters** of the distribution. However, it is sometimes useful to use the **canonical parameters** or **natural parameters**, defined as

$$\boldsymbol{\Lambda} \triangleq \boldsymbol{\Sigma}^{-1}, \quad \boldsymbol{\xi} \triangleq \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \tag{4.85}$$

We can convert back to the moment parameters using

$$\boldsymbol{\mu} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\xi}, \quad \boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1} \tag{4.86}$$

Using the canonical parameters, we can write the MVN in **information form** (i.e., in exponential family form, defined in Section 9.2):

$$\mathcal{N}_c(\mathbf{x}|\boldsymbol{\xi}, \boldsymbol{\Lambda}) = (2\pi)^{-D/2}|\boldsymbol{\Lambda}|^{\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}^T\boldsymbol{\Lambda}\mathbf{x} + \boldsymbol{\xi}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{\xi} - 2\mathbf{x}^T\boldsymbol{\xi})\right] \tag{4.87}$$

where we use the notation $\mathcal{N}_c()$ to distinguish from the moment parameterization $\mathcal{N}()$.

It is also possible to derive the marginalization and conditioning formulas in information form. We find

$$p(\mathbf{x}_2) = \mathcal{N}_c(\mathbf{x}_2|\boldsymbol{\xi}_2 - \boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\xi}_1, \boldsymbol{\Lambda}_{22} - \boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}) \tag{4.88}$$

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}_c(\mathbf{x}_1|\boldsymbol{\xi}_1 - \boldsymbol{\Lambda}_{12}\mathbf{x}_2, \boldsymbol{\Lambda}_{11}) \tag{4.89}$$

Thus we see that marginalization is easier in moment form, and conditioning is easier in information form.

Another operation that is significantly easier in information form is multiplying two Gaussians. One can show that

$$\mathcal{N}_c(\xi_f, \lambda_f)\mathcal{N}_c(\xi_g, \lambda_g) \quad = \quad \mathcal{N}_c(\xi_f + \xi_g, \lambda_f + \lambda_g) \tag{4.90}$$

However, in moment form, things are much messier:

$$\mathcal{N}(\mu_f, \sigma_f^2)\mathcal{N}(\mu_g, \sigma_g^2) = \mathcal{N}\left(\frac{\mu_f\sigma_g^2 + \mu_g\sigma_f^2}{\sigma_g^2 + \sigma_g^2}, \frac{\sigma_f^2\sigma_g^2}{\sigma_g^2 + \sigma_g^2}\right) \tag{4.91}$$

4.3.4 Proof of the result *

We now prove Theorem 4.3.1. Readers who are intimidated by heavy matrix algebra can safely skip this section. We first derive some results that we will need here and elsewhere in the book. We will return to the proof at the end.

4.3.4.1 Inverse of a partitioned matrix using Schur complements

The key tool we need is a way to invert a partitioned matrix. This can be done using the following result.

Theorem 4.3.2 (Inverse of a partitioned matrix)**.** *Consider a general partitioned matrix*

$$\mathbf{M} = \begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix} \tag{4.92}$$

where we assume $\mathbf{E}$ *and* $\mathbf{H}$ *are invertible. We have*

$$\mathbf{M}^{-1} = \begin{pmatrix} (\mathbf{M}/\mathbf{H})^{-1} & -(\mathbf{M}/\mathbf{H})^{-1}\mathbf{F}\mathbf{H}^{-1} \\ -\mathbf{H}^{-1}\mathbf{G}(\mathbf{M}/\mathbf{H})^{-1} & \mathbf{H}^{-1} + \mathbf{H}^{-1}\mathbf{G}(\mathbf{M}/\mathbf{H})^{-1}\mathbf{F}\mathbf{H}^{-1} \end{pmatrix} \tag{4.93}$$

$$= \begin{pmatrix} \mathbf{E}^{-1} + \mathbf{E}^{-1}\mathbf{F}(\mathbf{M}/\mathbf{E})^{-1}\mathbf{G}\mathbf{E}^{-1} & -\mathbf{E}^{-1}\mathbf{F}(\mathbf{M}/\mathbf{E})^{-1} \\ -(\mathbf{M}/\mathbf{E})^{-1}\mathbf{G}\mathbf{E}^{-1} & (\mathbf{M}/\mathbf{E})^{-1} \end{pmatrix} \tag{4.94}$$

where

$$\mathbf{M}/\mathbf{H} \triangleq \mathbf{E} - \mathbf{F}\mathbf{H}^{-1}\mathbf{G} \tag{4.95}$$

$$\mathbf{M}/\mathbf{E} \triangleq \mathbf{H} - \mathbf{G}\mathbf{E}^{-1}\mathbf{F} \tag{4.96}$$

We say that $\mathbf{M}/\mathbf{H}$ *is the* **Schur complement** *of* $\mathbf{M}$ *wrt* $\mathbf{H}$*. Equation 4.93 is called the* **partitioned inverse formula**.

Proof. If we could block diagonalize $\mathbf{M}$, it would be easier to invert. To zero out the top right block of $\mathbf{M}$ we can pre-multiply as follows

$$\begin{pmatrix} \mathbf{I} & -\mathbf{F}\mathbf{H}^{-1} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix} = \begin{pmatrix} \mathbf{E} - \mathbf{F}\mathbf{H}^{-1}\mathbf{G} & \mathbf{0} \\ \mathbf{G} & \mathbf{H} \end{pmatrix} \tag{4.97}$$

Similarly, to zero out the bottom left we can post-multiply as follows

$$\begin{pmatrix} \mathbf{E}-\mathbf{F}\mathbf{H}^{-1}\mathbf{G} & \mathbf{0} \\ \mathbf{G} & \mathbf{H} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{H}^{-1}\mathbf{G} & \mathbf{I} \end{pmatrix} = \begin{pmatrix} \mathbf{E}-\mathbf{F}\mathbf{H}^{-1}\mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{pmatrix} \tag{4.98}$$

Putting it all together we get

$$\underbrace{\begin{pmatrix} \mathbf{I} & -\mathbf{F}\mathbf{H}^{-1} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}}_{\mathbf{X}} \underbrace{\begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix}}_{\mathbf{M}} \underbrace{\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{H}^{-1}\mathbf{G} & \mathbf{I} \end{pmatrix}}_{\mathbf{Z}} = \underbrace{\begin{pmatrix} \mathbf{E}-\mathbf{F}\mathbf{H}^{-1}\mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{pmatrix}}_{\mathbf{W}} \tag{4.99}$$

Taking the inverse of both sides yields

$$\mathbf{Z}^{-1}\mathbf{M}^{-1}\mathbf{X}^{-1} = \mathbf{W}^{-1} \tag{4.100}$$

and hence

$$\mathbf{M}^{-1} = \mathbf{Z}\mathbf{W}^{-1}\mathbf{X} \tag{4.101}$$

Substituting in the definitions we get

$$\begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{H}^{-1}\mathbf{G} & \mathbf{I} \end{pmatrix} \begin{pmatrix} (\mathbf{M}/\mathbf{H})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\mathbf{F}\mathbf{H}^{-1} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{4.102}$$

$$= \begin{pmatrix} (\mathbf{M}/\mathbf{H})^{-1} & \mathbf{0} \\ -\mathbf{H}^{-1}\mathbf{G}(\mathbf{M}/\mathbf{H})^{-1} & \mathbf{H}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\mathbf{F}\mathbf{H}^{-1} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{4.103}$$

$$= \begin{pmatrix} (\mathbf{M}/\mathbf{H})^{-1} & -(\mathbf{M}/\mathbf{H})^{-1}\mathbf{F}\mathbf{H}^{-1} \\ -\mathbf{H}^{-1}\mathbf{G}(\mathbf{M}/\mathbf{H})^{-1} & \mathbf{H}^{-1}+\mathbf{H}^{-1}\mathbf{G}(\mathbf{M}/\mathbf{H})^{-1}\mathbf{F}\mathbf{H}^{-1} \end{pmatrix} \tag{4.104}$$

Alternatively, we could have decomposed the matrix $\mathbf{M}$ in terms of $\mathbf{E}$ and $\mathbf{M}/\mathbf{E} = (\mathbf{H} - \mathbf{G}\mathbf{E}^{-1}\mathbf{F})$, yielding

$$\begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{E}^{-1}+\mathbf{E}^{-1}\mathbf{F}(\mathbf{M}/\mathbf{E})^{-1}\mathbf{G}\mathbf{E}^{-1} & -\mathbf{E}^{-1}\mathbf{F}(\mathbf{M}/\mathbf{E})^{-1} \\ -(\mathbf{M}/\mathbf{E})^{-1}\mathbf{G}\mathbf{E}^{-1} & (\mathbf{M}/\mathbf{E})^{-1} \end{pmatrix} \tag{4.105}$$

□

4.3.4.2 The matrix inversion lemma

We now derive some useful corollaries of the above result.

Corollary 4.3.1 (Matrix inversion lemma). *Consider a general partitioned matrix* $\mathbf{M} = \begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix}$, *where we assume* $\mathbf{E}$ *and* $\mathbf{H}$ *are invertible. We have*

$$(\mathbf{E}-\mathbf{F}\mathbf{H}^{-1}\mathbf{G})^{-1} = \mathbf{E}^{-1}+\mathbf{E}^{-1}\mathbf{F}(\mathbf{H}-\mathbf{G}\mathbf{E}^{-1}\mathbf{F})^{-1}\mathbf{G}\mathbf{E}^{-1} \tag{4.106}$$

$$(\mathbf{E}-\mathbf{F}\mathbf{H}^{-1}\mathbf{G})^{-1}\mathbf{F}\mathbf{H}^{-1} = \mathbf{E}^{-1}\mathbf{F}(\mathbf{H}-\mathbf{G}\mathbf{E}^{-1}\mathbf{F})^{-1} \tag{4.107}$$

$$|\mathbf{E}-\mathbf{F}\mathbf{H}^{-1}\mathbf{G}| = |\mathbf{H}-\mathbf{G}\mathbf{E}^{-1}\mathbf{F}||\mathbf{H}^{-1}||\mathbf{E}| \tag{4.108}$$

The first two equations are s known as the **matrix inversion lemma** or the **Sherman-Morrison-Woodbury formula**. The third equation is known as the **matrix determinant lemma**. A typical application in machine learning/ statistics is the following. Let $\mathbf{E} = \mathbf{\Sigma}$ be a $N \times N$ diagonal matrix, let $\mathbf{F} = \mathbf{G}^T = \mathbf{X}$ of size $N \times D$, where $N \gg D$, and let $\mathbf{H}^{-1} = -\mathbf{I}$. Then we have

$$(\mathbf{\Sigma} + \mathbf{X}\mathbf{X}^T)^{-1} = \mathbf{\Sigma}^{-1} - \mathbf{\Sigma}^{-1}\mathbf{X}(\mathbf{I} + \mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{\Sigma}^{-1} \tag{4.109}$$

The LHS takes $O(N^3)$ time to compute, the RHS takes time $O(D^3)$ to compute.

Another application concerns computing a **rank one update** of an inverse matrix. Let $H = -1$ (a scalar), $\mathbf{F} = \mathbf{u}$ (a column vector), and $\mathbf{G} = \mathbf{v}^T$ (a row vector). Then we have

$$(\mathbf{E} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{E}^{-1} + \mathbf{E}^{-1}\mathbf{u}(-1 - \mathbf{v}^T\mathbf{E}^{-1}\mathbf{u})^{-1}\mathbf{v}^T\mathbf{E}^{-1} \tag{4.110}$$

$$= \mathbf{E}^{-1} - \frac{\mathbf{E}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{E}^{-1}}{1 + \mathbf{v}^T\mathbf{E}^{-1}\mathbf{u}} \tag{4.111}$$

This is useful when we incrementally add a data vector to a design matrix, and want to update our sufficient statistics. (One can derive an analogous formula for removing a data vector.)

Proof. To prove Equation 4.106, we simply equate the top left block of Equation 4.93 and Equation 4.94. To prove Equation 4.107, we simple equate the top right blocks of Equations 4.93 and 4.94. The proof of Equation 4.108 is left as an exercise. □

4.3.4.3 Proof of Gaussian conditioning formulas

We can now return to our original goal, which is to derive Equation 4.69. Let us factor the joint $p(\mathbf{x}_1, \mathbf{x}_2)$ as $p(\mathbf{x}_2)p(\mathbf{x}_1|\mathbf{x}_2)$ as follows:

$$E = \exp\left\{-\frac{1}{2}\begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1 \\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}^T \begin{pmatrix}\mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22}\end{pmatrix}^{-1} \begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1 \\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}\right\} \tag{4.112}$$

Using Equation 4.102 the above exponent becomes

$$E = \exp\left\{-\frac{1}{2}\begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1 \\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}^T \begin{pmatrix}\mathbf{I} & \mathbf{0} \\ -\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21} & \mathbf{I}\end{pmatrix} \begin{pmatrix}(\mathbf{\Sigma}/\mathbf{\Sigma}_{22})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{22}^{-1}\end{pmatrix}\right. \tag{4.113}$$

$$\left. \times \begin{pmatrix}\mathbf{I} & -\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1} \\ \mathbf{0} & \mathbf{I}\end{pmatrix} \begin{pmatrix}\mathbf{x}_1 - \boldsymbol{\mu}_1 \\ \mathbf{x}_2 - \boldsymbol{\mu}_2\end{pmatrix}\right\} \tag{4.114}$$

$$= \exp\left\{-\frac{1}{2}(\mathbf{x}_1 - \boldsymbol{\mu}_1 - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2))^T(\mathbf{\Sigma}/\mathbf{\Sigma}_{22})^{-1}\right. \tag{4.115}$$

$$\left.(\mathbf{x}_1 - \boldsymbol{\mu}_1 - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2))\right\} \times \exp\left\{-\frac{1}{2}(\mathbf{x}_2 - \boldsymbol{\mu}_2)^T\mathbf{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2)\right\} \tag{4.116}$$

This is of the form

$$\exp(\text{quadratic form in } \mathbf{x}_1, \mathbf{x}_2) \times \exp(\text{quadratic form in } \mathbf{x}_2) \tag{4.117}$$

Hence we have successfully factorized the joint as

$$p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_1|\mathbf{x}_2)p(\mathbf{x}_2) \tag{4.118}$$
$$= \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2})\mathcal{N}(\mathbf{x}_2|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \tag{4.119}$$

where the parameters of the conditional distribution can be read off from the above equations using

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \tag{4.120}$$
$$\boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{22} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \tag{4.121}$$

We can also use the fact that $|\mathbf{M}| = |\mathbf{M}/\mathbf{H}||\mathbf{H}|$ to check the normalization constants are correct:

$$(2\pi)^{(d_1+d_2)/2}|\boldsymbol{\Sigma}|^{\frac{1}{2}} = (2\pi)^{(d_1+d_2)/2}(|\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{22}|\ |\boldsymbol{\Sigma}_{22}|)^{\frac{1}{2}} \tag{4.122}$$
$$= (2\pi)^{d_1/2}|\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{22}|^{\frac{1}{2}}\ (2\pi)^{d_2/2}|\boldsymbol{\Sigma}_{22}|^{\frac{1}{2}} \tag{4.123}$$

where $d_1 = \dim(\mathbf{x}_1)$ and $d_2 = \dim(\mathbf{x}_2)$.

We leave the proof of the other forms of the result in Equation 4.69 as an exercise.

4.4 Linear Gaussian systems

Suppose we have two variables, $\mathbf{x}$ and $\mathbf{y}$. Let $\mathbf{x} \in \mathbb{R}^{D_x}$ be a hidden variable, and $\mathbf{y} \in \mathbb{R}^{D_y}$ be a noisy observation of $\mathbf{x}$. Let us assume we have the following prior and likelihood:

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \\ p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \boldsymbol{\Sigma}_y) \end{aligned} \tag{4.124}$$

where $\mathbf{A}$ is a matrix of size $D_y \times D_x$. This is an example of a **linear Gaussian system**. We can represent this schematically as $\mathbf{x} \rightarrow \mathbf{y}$, meaning $\mathbf{x}$ generates $\mathbf{y}$. In this section, we show how to "invert the arrow", that is, how to infer $\mathbf{x}$ from $\mathbf{y}$. We state the result below, then give several examples, and finally we derive the result. We will see many more applications of these results in later chapters.

4.4.1 Statement of the result

Theorem 4.4.1 (Bayes rule for linear Gaussian systems)**.** *Given a linear Gaussian system, as in Equation 4.124, the posterior* $p(\mathbf{x}|\mathbf{y})$ *is given by the following:*

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y}) \\ \boldsymbol{\Sigma}_{x|y}^{-1} &= \boldsymbol{\Sigma}_x^{-1} + \mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{A} \\ \boldsymbol{\mu}_{x|y} &= \boldsymbol{\Sigma}_{x|y}[\mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}\ (\mathbf{y} - \mathbf{b}) + \boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x] \end{aligned} \tag{4.125}$$

In addition, the normalization constant $p(\mathbf{y})$ is given by

$$\boxed{p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}_x + \mathbf{b}, \boldsymbol{\Sigma}_y + \mathbf{A}\boldsymbol{\Sigma}_x\mathbf{A}^T)} \tag{4.126}$$

For the proof, see Section 4.4.3.

4.4.2 Examples

In this section, we give some example applications of the above result.

4.4.2.1 Inferring an unknown scalar from noisy measurements

Suppose we make N noisy measurements y_i of some underlying quantity x; let us assume the measurement noise has fixed precision $\lambda_y = 1/\sigma^2$, so the likelihood is

$$p(y_i|x) = \mathcal{N}(y_i|x, \lambda_y^{-1}) \tag{4.127}$$

Now let us use a Gaussian prior for the value of the unknown source:

$$p(x) = \mathcal{N}(x|\mu_0, \lambda_0^{-1}) \tag{4.128}$$

We want to compute $p(x|y_1, \ldots, y_N, \sigma^2)$. We can convert this to a form that lets us apply Bayes rule for Gaussians by defining $\mathbf{y} = (y_1, \ldots, y_N)$, $\mathbf{A} = \mathbf{1}_N^T$ (an $1 \times N$ row vector of 1's), and $\boldsymbol{\Sigma}_y^{-1} = \mathrm{diag}(\lambda_y \mathbf{I})$. Then we get

$$p(x|\mathbf{y}) = \mathcal{N}(x|\mu_N, \lambda_N^{-1}) \tag{4.129}$$

$$\lambda_N = \lambda_0 + N\lambda_y \tag{4.130}$$

$$\mu_N = \frac{N\lambda_y \overline{y} + \lambda_0 \mu_0}{\lambda_N} = \frac{N\lambda_y}{N\lambda_y + \lambda_0}\overline{y} + \frac{\lambda_0}{N\lambda_y + \lambda_0}\mu_0 \tag{4.131}$$

These equations are quite intuitive: the posterior precision λ_N is the prior precision λ_0 plus N units of measurement precision λ_y. Also, the posterior mean μ_N is a convex combination of the MLE $\overline{y}$ and the prior mean μ_0. This makes it clear that the posterior mean is a compromise between the MLE and the prior. If the prior is weak relative to the signal strength (λ_0 is small relative to λ_y), we put more weight on the MLE. If the prior is strong relative to the signal strength (λ_0 is large relative to λ_y), we put more weight on the prior. This is illustrated in Figure 4.12, which is very similar to the analogous results for the beta-binomial model in Figure 3.6.

Note that the posterior mean is written in terms of $N\lambda_y\overline{y}$, so having N measurements each of precision λ_y is like having one measurement with value $\overline{y}$ and precision $N\lambda_y$.

We can rewrite the results in terms of the posterior variance, rather than posterior precision,

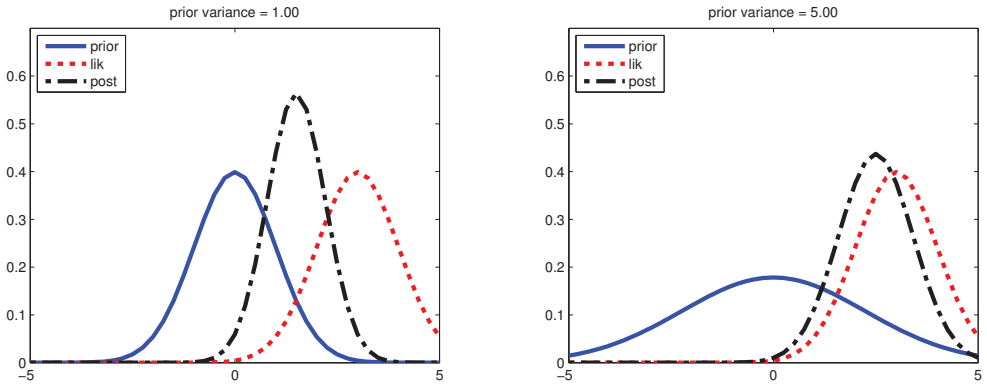


Figure 4.12 Inference about x given a noisy observation $y = 3$. (a) Strong prior $\mathcal{N}(0, 1)$. The posterior mean is "shrunk" towards the prior mean, which is 0. (a) Weak prior $\mathcal{N}(0, 5)$. The posterior mean is similar to the MLE. Figure generated by `gaussInferParamsMean1d`.

as follows:

$$p(x|\mathcal{D}, \sigma^2) = \mathcal{N}(x|\mu_N, \tau_N^2) \tag{4.132}$$

$$\tau_N^2 = \frac{1}{\frac{N}{\sigma^2} + \frac{1}{\tau_0^2}} = \frac{\sigma^2\tau_0^2}{N\tau_0^2 + \sigma^2} \tag{4.133}$$

$$\mu_N = \tau_N^2\left(\frac{\mu_0}{\tau_0^2} + \frac{N\overline{y}}{\sigma^2}\right) = \frac{\sigma^2}{N\tau_0^2 + \sigma^2}\mu_0 + \frac{N\tau_0^2}{N\tau_0^2 + \sigma^2}\overline{y} \tag{4.134}$$

where $\tau_0^2 = 1/\lambda_0$ is the prior variance and $\tau_N^2 = 1/\lambda_N$ is the posterior variance.

We can also compute the posterior sequentially, by updating after each observation. If $N = 1$, we can rewrite the posterior after seeing a single observation as follows (where we define $\Sigma_y = \sigma^2$, $\Sigma_0 = \tau_0^2$ and $\Sigma_1 = \tau_1^2$ to be the variances of the likelihood, prior and posterior):

$$p(x|y) = \mathcal{N}(x|\mu_1, \Sigma_1) \tag{4.135}$$

$$\Sigma_1 = \left(\frac{1}{\Sigma_0} + \frac{1}{\Sigma_y}\right)^{-1} = \frac{\Sigma_y\Sigma_0}{\Sigma_0 + \Sigma_y} \tag{4.136}$$

$$\mu_1 = \Sigma_1\left(\frac{\mu_0}{\Sigma_0} + \frac{y}{\Sigma_y}\right) \tag{4.137}$$

We can rewrite the posterior mean in 3 different ways:

$$\mu_1 = \frac{\Sigma_y}{\Sigma_y + \Sigma_0}\mu_0 + \frac{\Sigma_0}{\Sigma_y + \Sigma_0}y \tag{4.138}$$

$$= \mu_0 + (y - \mu_0)\frac{\Sigma_0}{\Sigma_y + \Sigma_0} \tag{4.139}$$

$$= y - (y - \mu_0)\frac{\Sigma_y}{\Sigma_y + \Sigma_0} \tag{4.140}$$

The first equation is a convex combination of the prior and the data. The second equation is the prior mean adjusted towards the data. The third equation is the data adjusted towards the prior mean; this is called **shrinkage**. These are all equivalent ways of expressing the tradeoff between likelihood and prior. If Σ_0 is small relative to Σ_Y, corresponding to a strong prior, the amount of shrinkage is large (see Figure 4.12(a)), whereas if Σ_0 is large relative to Σ_y, corresponding to a weak prior, the amount of shrinkage is small (see Figure 4.12(b)).

Another way to quantify the amount of shrinkage is in terms of the **signal-to-noise ratio**, which is defined as follows:

$$\text{SNR} \triangleq \frac{\mathbb{E}\left[X^2\right]}{\mathbb{E}\left[\epsilon^2\right]} = \frac{\Sigma_0 + \mu_0^2}{\Sigma_y} \tag{4.141}$$

where $x \sim \mathcal{N}(\mu_0, \Sigma_0)$ is the true signal, $y = x + \epsilon$ is the observed signal, and $\epsilon \sim \mathcal{N}(0, \Sigma_y)$ is the noise term.

4.4.2.2 Inferring an unknown vector from noisy measurements

Now consider N vector-valued observations, $\mathbf{y}_i \sim \mathcal{N}(\mathbf{x}, \mathbf{\Sigma}_y)$, and a Gaussian prior, $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0)$. Setting $\mathbf{A} = \mathbf{I}$, $\mathbf{b} = \mathbf{0}$, and using $\overline{\mathbf{y}}$ for the effective observation with precision $N\mathbf{\Sigma}_y^{-1}$, we have

$$p(\mathbf{x}|\mathbf{y}_1, \ldots, \mathbf{y}_N) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_N, \mathbf{\Sigma}_N) \tag{4.142}$$

$$\mathbf{\Sigma}_N^{-1} = \mathbf{\Sigma}_0^{-1} + N\mathbf{\Sigma}_y^{-1} \tag{4.143}$$

$$\boldsymbol{\mu}_N = \mathbf{\Sigma}_N(\mathbf{\Sigma}_y^{-1}(N\overline{\mathbf{y}}) + \mathbf{\Sigma}_0^{-1}\boldsymbol{\mu}_0) \tag{4.144}$$

See Figure 4.13 for a 2d example. We can think of $\mathbf{x}$ as representing the true, but unknown, location of an object in 2d space, such as a missile or airplane, and the $\mathbf{y}_i$ as being noisy observations, such as radar "blips". As we receive more blips, we are better able to localize the source. In Section 18.3.1, we will see how to extend this example to track moving objects using the famous Kalman filter algorithm.

Now suppose we have multiple measuring devices, and we want to combine them together; this is known as **sensor fusion**. If we have multiple observations with different covariances (corresponding to sensors with different reliabilities), the posterior will be an appropriate weighted average of the data. Consider the example in Figure 4.14. We use an uninformative prior on $\mathbf{x}$, namely $p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0) = \mathcal{N}(\mathbf{0}, 10^{10}\mathbf{I}_2)$. We get 2 noisy observations, $\mathbf{y}_1 \sim \mathcal{N}(\mathbf{x}, \mathbf{\Sigma}_{y,1})$ and $\mathbf{y}_2 \sim \mathcal{N}(\mathbf{x}, \mathbf{\Sigma}_{y,2})$. We then compute $p(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2)$.

In Figure 4.14(a), we set $\mathbf{\Sigma}_{y,1} = \mathbf{\Sigma}_{y,2} = 0.01\mathbf{I}_2$, so both sensors are equally reliable. In this case, the posterior mean is half way between the two observations, $\mathbf{y}_1$ and $\mathbf{y}_2$. In Figure 4.14(b), we set $\mathbf{\Sigma}_{y,1} = 0.05\mathbf{I}_2$ and $\mathbf{\Sigma}_{y,2} = 0.01\mathbf{I}_2$, so sensor 2 is more reliable than sensor 1. In this case, the posterior mean is closer to $\mathbf{y}_2$. In Figure 4.14(c), we set

$$\mathbf{\Sigma}_{y,1} = 0.01\begin{pmatrix} 10 & 1 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{\Sigma}_{y,2} = 0.01\begin{pmatrix} 1 & 1 \\ 1 & 10 \end{pmatrix} \tag{4.145}$$

so sensor 1 is more reliable in the y_2 component (vertical direction), and sensor 2 is more reliable in the y_1 component (horizontal direction). In this case, the posterior mean uses $\mathbf{y}_1$'s vertical component and $\mathbf{y}_2$'s horizontal component.

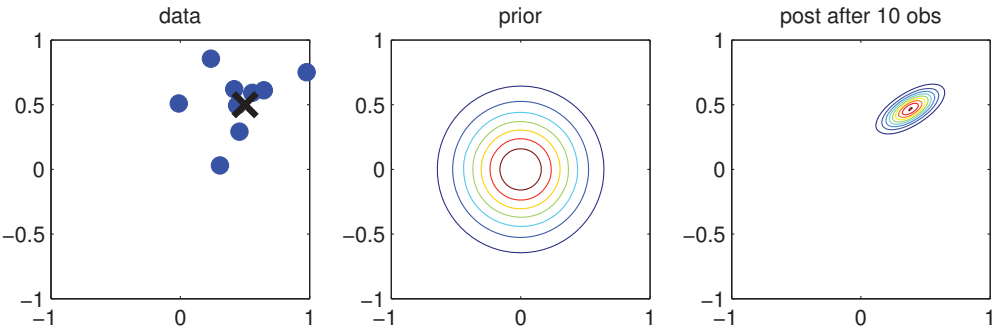


Figure 4.13 Illustration of Bayesian inference for the mean of a 2d Gaussian. (a) The data is generated from $\mathbf{y}_i \sim \mathcal{N}(\mathbf{x}, \mathbf{\Sigma}_y)$, where $\mathbf{x} = [0.5, 0.5]^T$ and $\mathbf{\Sigma}_y = 0.1[2, 1; 1, 1]$). We assume the sensor noise covariance $\mathbf{\Sigma}_y$ is known but $\mathbf{x}$ is unknown. The black cross represents $\mathbf{x}$. (b) The prior is $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, 0.1\mathbf{I}_2)$. (c) We show the posterior after 10 data points have been observed. Figure generated by `gaussInferParamsMean2d`.

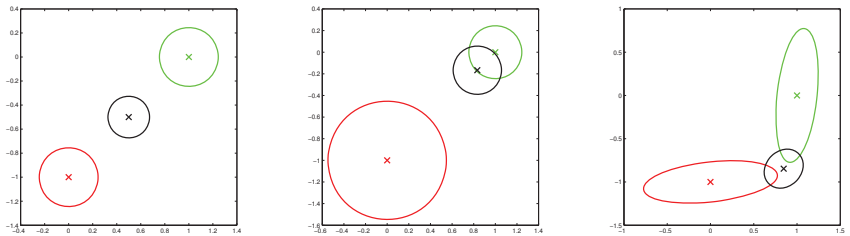


Figure 4.14 We observe $\mathbf{y}_1 = (0, -1)$ (red cross) and $\mathbf{y}_2 = (1, 0)$ (green cross) and infer $E(\boldsymbol{\mu}|\mathbf{y}_1, \mathbf{y}_2, \boldsymbol{\theta})$ (black cross). (a) Equally reliable sensors, so the posterior mean estimate is in between the two circles. (b) Sensor 2 is more reliable, so the estimate shifts more towards the green circle. (c) Sensor 1 is more reliable in the vertical direction, Sensor 2 is more reliable in the horizontal direction. The estimate is an appropriate combination of the two measurements. Figure generated by `sensorFusion2d`.

Note that this technique crucially relies on modeling our uncertainty of each sensor; computing an unweighted average would give the wrong result. However, we have assumed the sensor precisions are known. When they are not, we should model out uncertainty about $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ as well. See Section 4.6.4 for details.

4.4.2.3 Interpolating noisy data

We now revisit the example of Section 4.3.2.2. This time we no longer assume noise-free observations. Instead, let us assume that we obtain N noisy observations y_i; without loss of generality, assume these correspond to $x_1, \ldots, x_N$. We can model this setup as a linear

Gaussian system:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon} \tag{4.146}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_y)$, $\boldsymbol{\Sigma}_y = \sigma^2\mathbf{I}$, σ^2 is the observation noise, and $\mathbf{A}$ is a $N \times D$ projection matrix that selects out the observed elements. For example, if $N = 2$ and $D = 4$ we have

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{4.147}$$

Using the same improper prior as before, $\boldsymbol{\Sigma}_x = (\mathbf{L}^T\mathbf{L})^{-1}$, we can easily compute the posterior mean and variance. In Figure 4.15, we plot the posterior mean, posterior variance, and some posterior samples. Now we see that the prior precision λ effects the posterior mean as well as the posterior variance. In particular, for a strong prior (large λ), the estimate is very smooth, and the uncertainty is low. but for a weak prior (small λ), the estimate is wiggly, and the uncertainty (away from the data) is high.

The posterior mean can also be computed by solving the following optimization problem:

$$\min_{\mathbf{x}} \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - y_i)^2 + \frac{\lambda}{2}\sum_{j=1}^{D}\left[(x_j - x_{j-1})^2 + (x_j - x_{j+1})^2\right] \tag{4.148}$$

where we have defined $x_0 = x_1$ and $x_{D+1} = x_D$ for notational simplicity. We recognize this as a discrete approximation to the following problem:

$$\min_{f} \frac{1}{2\sigma^2}\int (f(t) - y(t))^2 dt + \frac{\lambda}{2}\int [f'(t)]^2 dt \tag{4.149}$$

where $f'(t)$ is the first derivative of f. The first term measures fit to the data, and the second term penalizes functions that are "too wiggly". This is an example of **Tikhonov regularization**, which is a popular approach to **functional data analysis**. See Chapter 15 for more sophisticated approaches, which enforce higher order smoothness (so the resulting samples look less "jagged").

4.4.3 Proof of the result *

We now derive Equation 4.125. The basic idea is to derive the joint distribution, $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x})$, and then to use the results from Section 4.3.1 for computing $p(\mathbf{x}|\mathbf{y})$.

In more detail, we proceed as follows. The log of the joint distribution is as follows (dropping irrelevant constants):

$$\log p(\mathbf{x}, \mathbf{y}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_x)^T\boldsymbol{\Sigma}_x^{-1}(\mathbf{x} - \boldsymbol{\mu}_x) - \frac{1}{2}(\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{b})^T\boldsymbol{\Sigma}_y^{-1}(\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{b}) \tag{4.150}$$

This is clearly a joint Gaussian distribution, since it is the exponential of a quadratic form.

Expanding out the quadratic terms involving $\mathbf{x}$ and $\mathbf{y}$, and ignoring linear and constant terms, we have

$$\begin{aligned} Q &= -\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}_x^{-1}\mathbf{x} - \frac{1}{2}\mathbf{y}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{y} - \frac{1}{2}(\mathbf{A}\mathbf{x})^T\boldsymbol{\Sigma}_y^{-1}(\mathbf{A}\mathbf{x}) + \mathbf{y}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{A}\mathbf{x} && (4.151) \\ &= -\frac{1}{2}\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^T\begin{pmatrix}\boldsymbol{\Sigma}_x^{-1} + \mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{A} & -\mathbf{A}^T\boldsymbol{\Sigma}_y^{-1} \\ -\boldsymbol{\Sigma}_y^{-1}\mathbf{A} & \boldsymbol{\Sigma}_y^{-1}\end{pmatrix}\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} && (4.152) \\ &= -\frac{1}{2}\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^T\boldsymbol{\Sigma}^{-1}\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} && (4.153) \end{aligned}$$

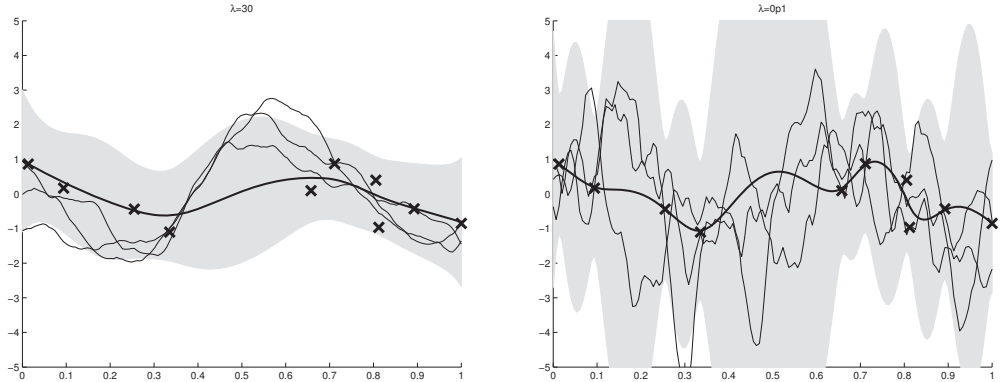


Figure 4.15 Interpolating noisy data (noise variance $\sigma^2 = 1$) using a Gaussian with prior precision λ. (a) $\lambda = 30$. (b) $\lambda = 0.01$. See also Figure 4.10. Based on Figure 7.1 of (Calvetti and Somersalo 2007). Figure generated by `gaussInterpNoisyDemo`. See also `splineBasisDemo`.

where the precision matrix of the joint is defined as

$$\boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Sigma}_x^{-1} + \mathbf{A}^T \boldsymbol{\Sigma}_y^{-1} \mathbf{A} & -\mathbf{A}^T \boldsymbol{\Sigma}_y^{-1} \\ -\boldsymbol{\Sigma}_y^{-1} \mathbf{A} & \boldsymbol{\Sigma}_y^{-1} \end{pmatrix} \triangleq \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{xx} & \boldsymbol{\Lambda}_{xy} \\ \boldsymbol{\Lambda}_{yx} & \boldsymbol{\Lambda}_{yy} \end{pmatrix} \tag{4.154}$$

From Equation 4.69, and using the fact that $\boldsymbol{\mu}_y = \mathbf{A}\boldsymbol{\mu}_x + \mathbf{b}$, we have

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y}) && (4.155) \\ \boldsymbol{\Sigma}_{x|y} &= \boldsymbol{\Lambda}_{xx}^{-1} = (\boldsymbol{\Sigma}_x^{-1} + \mathbf{A}^T \boldsymbol{\Sigma}_y^{-1} \mathbf{A})^{-1} && (4.156) \\ \boldsymbol{\mu}_{x|y} &= \boldsymbol{\Sigma}_{x|y} \left(\boldsymbol{\Lambda}_{xx} \boldsymbol{\mu}_x - \boldsymbol{\Lambda}_{xy} (\mathbf{y} - \boldsymbol{\mu}_y) \right) && (4.157) \\ &= \boldsymbol{\Sigma}_{x|y} \left(\boldsymbol{\Sigma}_x^{-1} \boldsymbol{\mu} + \mathbf{A}^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{y} - \mathbf{b}) \right) && (4.158) \end{aligned}$$

4.5 Digression: The Wishart distribution *

The **Wishart** distribution is the generalization of the Gamma distribution to positive definite matrices. Press (Press 2005, p107) has said "The Wishart distribution ranks next to the (multivariate) normal distribution in order of importance and usefuleness in multivariate statistics". We will mostly use it to model our uncertainty in covariance matrices, $\boldsymbol{\Sigma}$, or their inverses, $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$.

The pdf of the Wishart is defined as follows:

$$\mathrm{Wi}(\boldsymbol{\Lambda}|\mathbf{S}, \nu) = \frac{1}{Z_{\mathrm{Wi}}} |\boldsymbol{\Lambda}|^{(\nu - D - 1)/2} \exp\left(-\frac{1}{2} \mathrm{tr}(\boldsymbol{\Lambda}\mathbf{S}^{-1}) \right) \tag{4.159}$$

Here ν is called the "degrees of freedom" and $\mathbf{S}$ is the "scale matrix". (We shall get more intuition for these parameters shortly.) The normalization constant for this distribution (which

requires integrating over all symmetric pd matrices) is the following formidable expression

$$Z_{\text{Wi}} = 2^{\nu D/2}\Gamma_D(\nu/2)|\mathbf{S}|^{\nu/2} \tag{4.160}$$

where $\Gamma_D(a)$ is the **multivariate gamma function**:

$$\Gamma_D(x) = \pi^{D(D-1)/4}\prod_{i=1}^{D}\Gamma\left(x + (1-i)/2\right) \tag{4.161}$$

Hence $\Gamma_1(a) = \Gamma(a)$ and

$$\Gamma_D(\nu_0/2) = \prod_{i=1}^{D}\Gamma(\frac{\nu_0 + 1 - i}{2}) \tag{4.162}$$

The normalization constant only exists (and hence the pdf is only well defined) if $\nu > D - 1$.

There is a connection between the Wishart distribution and the Gaussian. In particular, let $\mathbf{x}_i \sim \mathcal{N}(0, \mathbf{\Sigma})$. Then the scatter matrix $\mathbf{S} = \sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T$ has a Wishart distribution: $\mathbf{S} \sim \text{Wi}(\mathbf{\Sigma}, 1)$. Hence $\mathbb{E}[\mathbf{S}] = N\mathbf{\Sigma}$. More generally, one can show that the mean and mode of $\text{Wi}(\mathbf{S}, \nu)$ are given by

$$\text{mean} = \nu\mathbf{S}, \ \text{mode} = (\nu - D - 1)\mathbf{S} \tag{4.163}$$

where the mode only exists if $\nu > D + 1$.

If $D = 1$, the Wishart reduces to the Gamma distribution:

$$\text{Wi}(\lambda|s^{-1}, \nu) = \text{Ga}(\lambda|\frac{\nu}{2}, \frac{s}{2}) \tag{4.164}$$

4.5.1 Inverse Wishart distribution

Recall that we showed (Exercise 2.10) that if $\lambda \sim \text{Ga}(a, b)$, then that $\frac{1}{\lambda} \sim \text{IG}(a, b)$. Similarly, if $\mathbf{\Sigma}^{-1} \sim \text{Wi}(\mathbf{S}, \nu)$ then $\mathbf{\Sigma} \sim \text{IW}(\mathbf{S}^{-1}, \nu + D + 1)$, where IW is the **inverse Wishart**, the multidimensional generalization of the inverse Gamma. It is defined as follows, for $\nu > D - 1$ and $\mathbf{S} \succ 0$:

$$\text{IW}(\mathbf{\Sigma}|\mathbf{S}, \nu) = \frac{1}{Z_{IW}}|\mathbf{\Sigma}|^{-(\nu+D+1)/2}\exp\left(-\frac{1}{2}\text{tr}(\mathbf{S}^{-1}\mathbf{\Sigma}^{-1})\right) \tag{4.165}$$

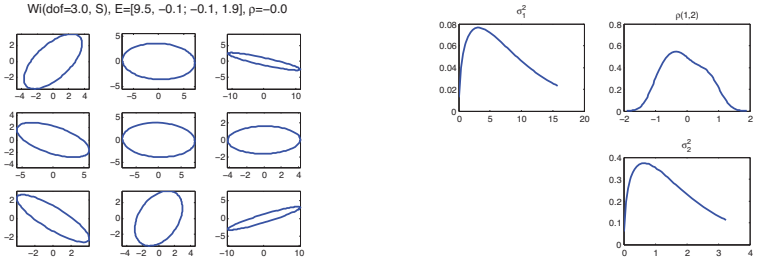
$$Z_{IW} = |\mathbf{S}|^{-\nu/2}2^{\nu D/2}\Gamma_D(\nu/2) \tag{4.166}$$

One can show that the distribution has these properties

$$\text{mean} = \frac{\mathbf{S}^{-1}}{\nu - D - 1}, \quad \text{mode} = \frac{\mathbf{S}^{-1}}{\nu + D + 1} \tag{4.167}$$

If $D = 1$, this reduces to the inverse Gamma:

$$\text{IW}(\sigma^2|S^{-1}, \nu) = \text{IG}(\sigma^2|\nu/2, S/2) \tag{4.168}$$



(a) (b)

Figure 4.16 Visualization of the Wishart distribution. Left: Some samples from the Wishart distribution, $\mathbf{\Sigma} \sim \text{Wi}(\mathbf{S}, \nu)$, where $\mathbf{S} = [3.1653, -0.0262; -0.0262, 0.6477]$ and $\nu = 3$. Right: Plots of the marginals (which are Gamma), and the approximate (sample-based) marginal on the correlation coefficient. If $\nu = 3$ there is a lot of uncertainty about the value of the correlation coefficient ρ (see the almost uniform distribution on $[-1, 1]$). The sampled matrices are highly variable, and some are nearly singular. As ν increases, the sampled matrices are more concentrated on the prior $\mathbf{S}$. Figure generated by `wiPlotDemo`.

4.5.2 Visualizing the Wishart distribution *

Since the Wishart is a distribution over matrices, it is hard to plot as a density function. However, we can easily sample from it, and in the 2d case, we can use the eigenvectors of the resulting matrix to define an ellipse, as explained in Section 4.1.2. See Figure 4.16 for some examples.

For higher dimensional matrices, we can plot marginals of the distribution. The diagonals of a Wishart distributed matrix have Gamma distributions, so are easy to plot. It is hard in general to work out the distribution of the off-diagonal elements, but we can sample matrices from the distribution, and then compute the distribution empirically. In particular, we can convert each sampled matrix to a correlation matrix, and thus compute a Monte Carlo approximation (Section 2.7) to the expected correlation coefficients:

$$\mathbb{E}\left[R_{ij}\right] \approx \frac{1}{S} \sum_{s=1}^{S} \mathbf{R}(\mathbf{\Sigma}^{(s)})_{ij} \tag{4.169}$$

where $\mathbf{\Sigma}^{(s)} \sim \text{Wi}(\mathbf{\Sigma}, \nu)$ and $\mathbf{R}(\mathbf{\Sigma})$ converts matrix $\mathbf{\Sigma}$ into a correlation matrix:

$$R_{ij} = \frac{\mathbf{\Sigma}_{ij}}{\sqrt{\mathbf{\Sigma}_{ii}\mathbf{\Sigma}_{jj}}} \tag{4.170}$$

We can then use kernel density estimation (Section 14.7.2) to produce a smooth approximation to the univariate density $\mathbb{E}\left[R_{ij}\right]$ for plotting purposes. See Figure 4.16 for some examples.

4.6 Inferring the parameters of an MVN

So far, we have discussed inference in a Gaussian assuming the parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \mathbf{\Sigma})$ are known. We now discuss how to infer the parameters themselves. We will assume the data has

the form $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for $i = 1 : N$ and is fully observed, so we have no missing data (see Section 11.6.1 for how to estimate parameters of an MVN in the presence of missing values). To simplify the presentation, we derive the posterior in three parts: first we compute $p(\boldsymbol{\mu}|\mathcal{D}, \boldsymbol{\Sigma})$; then we compute $p(\boldsymbol{\Sigma}|\mathcal{D}, \boldsymbol{\mu})$; finally we compute the joint $p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D})$.

4.6.1 Posterior distribution of $\boldsymbol{\mu}$

We have discussed how to compute the MLE for $\boldsymbol{\mu}$; we now discuss how to compute its posterior, which is useful for modeling our uncertainty about its value.

The likelihood has the form

$$p(\mathcal{D}|\boldsymbol{\mu}) = \mathcal{N}(\bar{\mathbf{x}}|\boldsymbol{\mu}, \frac{1}{N}\boldsymbol{\Sigma}) \tag{4.171}$$

For simplicity, we will use a conjugate prior, which in this case is a Gaussian. In particular, if $p(\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}_0, \mathbf{V}_0)$ then we can derive a Gaussian posterior for $\boldsymbol{\mu}$ based on the results in Section 4.4.2.2. We get

$$\begin{aligned} p(\boldsymbol{\mu}|\mathcal{D}, \boldsymbol{\Sigma}) &= \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}_N, \mathbf{V}_N) && (4.172)\\ \mathbf{V}_N^{-1} &= \mathbf{V}_0^{-1} + N\boldsymbol{\Sigma}^{-1} && (4.173)\\ \mathbf{m}_N &= \mathbf{V}_N(\boldsymbol{\Sigma}^{-1}(N\bar{\mathbf{x}}) + \mathbf{V}_0^{-1}\mathbf{m}_0) && (4.174) \end{aligned}$$

This is exactly the same process as inferring the location of an object based on noisy radar "blips", except now we are inferring the mean of a distribution based on noisy samples. (To a Bayesian, there is no difference between uncertainty about parameters and uncertainty about anything else.)

We can model an uninformative prior by setting $\mathbf{V}_0 = \infty\mathbf{I}$. In this case we have $p(\boldsymbol{\mu}|\mathcal{D}, \boldsymbol{\Sigma}) = \mathcal{N}(\bar{\mathbf{x}}, \frac{1}{N}\boldsymbol{\Sigma})$, so the posterior mean is equal to the MLE. We also see that the posterior variance goes down as $1/N$, which is a standard result from frequentist statistics.

4.6.2 Posterior distribution of $\boldsymbol{\Sigma}$ *

We now discuss how to compute $p(\boldsymbol{\Sigma}|\mathcal{D}, \boldsymbol{\mu})$. The likelihood has the form

$$p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto |\boldsymbol{\Sigma}|^{-\frac{N}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{S}_\mu\boldsymbol{\Sigma}^{-1})\right) \tag{4.175}$$

The corresponding conjugate prior is known as the inverse Wishart distribution (Section 4.5.1). Recall that this has the following pdf:

$$\mathrm{IW}(\boldsymbol{\Sigma}|\mathbf{S}_0^{-1}, \nu_0) \propto |\boldsymbol{\Sigma}|^{-(\nu_0+D+1)/2} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{S}_0\boldsymbol{\Sigma}^{-1})\right) \tag{4.176}$$

Here $\nu_0 > D - 1$ is the degrees of freedom (dof), and $\mathbf{S}_0$ is a symmetric pd matrix. We see that $\mathbf{S}_0^{-1}$ plays the role of the prior scatter matrix, and $N_0 \triangleq \nu_0 + D + 1$ controls the strength of the prior, and hence plays a role analogous to the sample size N.

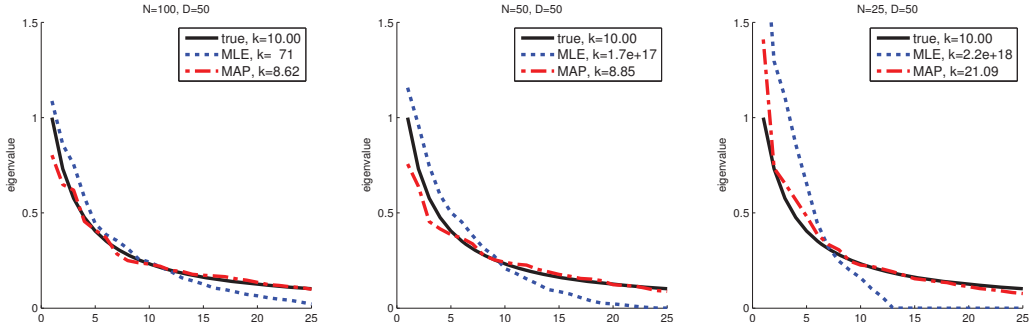


Figure 4.17 Estimating a covariance matrix in $D = 50$ dimensions using $N \in \{100, 50, 25\}$ samples. We plot the eigenvalues in descending order for the true covariance matrix (solid black), the MLE (dotted blue) and the MAP estimate (dashed red), using Equation 4.184 with $\lambda = 0.9$. We also list the condition number of each matrix in the legend. Based on Figure 1 of (Schaefer and Strimmer 2005). Figure generated by `shrinkcovDemo`.

Multiplying the likelihood and prior we find that the posterior is also inverse Wishart:

$$\begin{aligned} p(\boldsymbol{\Sigma}|\mathcal{D}, \boldsymbol{\mu}) &\propto |\boldsymbol{\Sigma}|^{-\frac{N}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_\mu)\right) |\boldsymbol{\Sigma}|^{-(\nu_0+D+1)/2} \\ &\quad \exp\left(-\frac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_0)\right) \quad (4.177) \\ &= |\boldsymbol{\Sigma}|^{-\frac{N+(\nu_0+D+1)}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left[\boldsymbol{\Sigma}^{-1}(\mathbf{S}_\mu + \mathbf{S}_0)\right]\right) \quad (4.178) \\ &= \mathrm{IW}(\boldsymbol{\Sigma}|\mathbf{S}_N, \nu_N) \quad (4.179) \\ \nu_N &= \nu_0 + N \quad (4.180) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0 + \mathbf{S}_\mu \quad (4.181) \end{aligned}$$

In words, this says that the posterior strength ν_N is the prior strength ν_0 plus the number of observations N, and the posterior scatter matrix $\mathbf{S}_N$ is the prior scatter matrix $\mathbf{S}_0$ plus the data scatter matrix $\mathbf{S}_\mu$.

4.6.2.1 MAP estimation

We see from Equation 4.7 that $\hat{\boldsymbol{\Sigma}}_{mle}$ is a rank $\min(N, D)$ matrix. If $N < D$, this is not full rank, and hence will be uninvertible. And even if $N > D$, it may be the case that $\hat{\boldsymbol{\Sigma}}$ is ill-conditioned (meaning it is nearly singular).

To solve these problems, we can use the posterior mode (or mean). One can show (using techniques analogous to the derivation of the MLE) that the MAP estimate is given by

$$\hat{\boldsymbol{\Sigma}}_{map} = \frac{\mathbf{S}_N}{\nu_N + D + 1} = \frac{\mathbf{S}_0 + \mathbf{S}_\mu}{N_0 + N} \quad (4.182)$$

If we use an improper uniform prior, corresponding to $N_0 = 0$ and $\mathbf{S}_0 = \mathbf{0}$, we recover the MLE.

Let us now consider the use of a proper informative prior, which is necessary whenever D/N is large (say bigger than 0.1). Let $\boldsymbol{\mu} = \overline{\mathbf{x}}$, so $\mathbf{S}_\mu = \mathbf{S}_{\overline{x}}$. Then we can rewrite the MAP estimate as a convex combination of the prior mode and the MLE. To see this, let $\boldsymbol{\Sigma}_0 \triangleq \frac{\mathbf{S}_0}{N_0}$ be the prior mode. Then the posterior mode can be rewritten as

$$\hat{\boldsymbol{\Sigma}}_{map} = \frac{\mathbf{S}_0 + \mathbf{S}_{\overline{x}}}{N_0 + N} = \frac{N_0}{N_0 + N}\frac{\mathbf{S}_0}{N_0} + \frac{N}{N_0 + N}\frac{\mathbf{S}}{N} = \lambda\boldsymbol{\Sigma}_0 + (1-\lambda)\hat{\boldsymbol{\Sigma}}_{mle} \tag{4.183}$$

where $\lambda = \frac{N_0}{N_0+N}$, controls the amount of shrinkage towards the prior.

This begs the question: where do the parameters of the prior come from? It is common to set λ by cross validation. Alternatively, we can use the closed-form formula provided in (Ledoit and Wolf 2004b,a; Schaefer and Strimmer 2005), which is the optimal frequentist estimate if we use squared loss. This is arguably not the most natural loss function for covariance matrices (because it ignores the postive definite constraint), but it results in a simple estimator, which is implemented in the PMTK function `shrinkcov`. We discuss Bayesian ways of estimating λ later.

As for the prior covariance matrix, $\mathbf{S}_0$, it is common to use the following (data dependent) prior: $\mathbf{S}_0 = \mathrm{diag}(\hat{\boldsymbol{\Sigma}}_{mle})$. In this case, the MAP estimate is given by

$$\hat{\boldsymbol{\Sigma}}_{map}(i,j) = \begin{cases} \hat{\boldsymbol{\Sigma}}_{mle}(i,j) & \text{if } i = j \\ (1-\lambda)\hat{\boldsymbol{\Sigma}}_{mle}(i,j) & \text{otherwise} \end{cases} \tag{4.184}$$

Thus we see that the diagonal entries are equal to their ML estimates, and the off diagonal elements are "shrunk" somewhat towards 0. This technique is therefore called **shrinkage estimation**, or **regularized estimation**.

The benefits of MAP estimation are illustrated in Figure 4.17. We consider fitting a 50 dimensional Gaussian to $N = 100$, $N = 50$ and $N = 25$ data points. We see that the MAP estimate is always well-conditioned, unlike the MLE. In particular, we see that the **eigenvalue spectrum** of the MAP estimate is much closer to that of the true matrix than the MLE's. The eigenvectors, however, are unaffected.

The importance of regularizing the estimate of $\boldsymbol{\Sigma}$ will become apparent in later chapters, when we consider fitting covariance matrices to high dimensional data.

4.6.2.2 Univariate posterior

In the 1d case, the likelihood has the form

$$p(\mathcal{D}|\sigma^2) \propto (\sigma^2)^{-N/2}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2\right) \tag{4.185}$$

The standard conjugate prior is the inverse Gamma distribution, which is just the scalar version of the inverse Wishart:

$$\mathrm{IG}(\sigma^2|a_0,b_0) \propto (\sigma^2)^{-(a_0+1)}\exp(-\frac{b_0}{\sigma^2}) \tag{4.186}$$

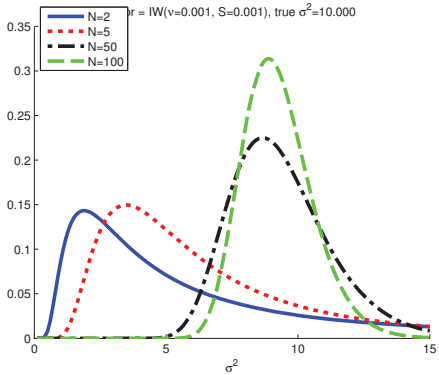


Figure 4.18 Sequential updating of the posterior for σ^2 starting from an uninformative prior. The data was generated from a Gaussian with known mean $\mu = 5$ and unknown variance $\sigma^2 = 10$. Figure generated by `gaussSeqUpdateSigma1D`.

Multiplying the likelihood and the prior, we see that the posterior is also IG:

$$p(\sigma^2|\mathcal{D}) = \text{IG}(\sigma^2|a_N, b_N) \tag{4.187}$$

$$a_N = a_0 + N/2 \tag{4.188}$$

$$b_N = b_0 + \frac{1}{2}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{4.189}$$

See Figure 4.18 for an illustration.

The form of the posterior is not quite as pretty as the multivariate case, because of the factors of $\frac{1}{2}$. This arises because $\text{IW}(\sigma^2|s_0, \nu_0) = \text{IG}(\sigma^2|\frac{s_0}{2}, \frac{\nu_0}{2})$. Another problem with using the $\text{IG}(a_0, b_0)$ distribution is that the strength of the prior is encoded in both a_0 and b_0. To avoid both of these problems, it is common (in the statistics literature) to use an alternative parameterization of the IG distribution, known as the (scaled) **inverse chi-squared distribution**. This is defined as follows:

$$\chi^{-2}(\sigma^2|\nu_0, \sigma_0^2) = \text{IG}(\sigma^2|\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}) \propto (\sigma^2)^{-\nu_0/2-1}\exp(-\frac{\nu_0\sigma_0^2}{2\sigma^2}) \tag{4.190}$$

Here ν_0 controls the strength of the prior, and σ_0^2 encodes the value of the prior. With this prior, the posterior becomes

$$p(\sigma^2|\mathcal{D}, \mu) = \chi^{-2}(\sigma^2|\nu_N, \sigma_N^2) \tag{4.191}$$

$$\nu_N = \nu_0 + N \tag{4.192}$$

$$\sigma_N^2 = \frac{\nu_0\sigma_0^2 + \sum_{i=1}^{N}(x_i - \mu)^2}{\nu_N} \tag{4.193}$$

We see that the posterior dof ν_N is the prior dof ν_0 plus N, and the posterior sum of squares $\nu_N\sigma_N^2$ is the prior sum of squares $\nu_0\sigma_0^2$ plus the data sum of squares.

We can emulate an uninformative prior, $p(\sigma^2) \propto \sigma^{-2}$, by setting $\nu_0 = 0$, which makes intuitive sense (since it corresponds to a zero virtual sample size).

4.6.3 Posterior distribution of μ and Σ *

We now discuss how to compute $p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D})$. These results are a bit complex, but will prove useful later on in this book. Feel free to skip this section on a first reading.

4.6.3.1 Likelihood

The likelihood is given by

$$p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-ND/2}|\Sigma|^{-\frac{N}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right) \tag{4.194}$$

Now one can show that

$$\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) = \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\overline{x}}) + N(\overline{\mathbf{x}} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}} - \boldsymbol{\mu}) \tag{4.195}$$

Hence we can rewrite the likelihood as follows:

$$p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-ND/2}|\Sigma|^{-\frac{N}{2}} \exp\left(-\frac{N}{2}(\boldsymbol{\mu} - \overline{\mathbf{x}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \overline{\mathbf{x}})\right) \tag{4.196}$$

$$\exp\left(-\frac{N}{2}\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_{\overline{x}})\right) \tag{4.197}$$

We will use this form below.

4.6.3.2 Prior

The obvious prior to use is the following

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}_0, \mathbf{V}_0)\mathrm{IW}(\boldsymbol{\Sigma}|\mathbf{S}_0, \nu_0) \tag{4.198}$$

Unfortunately, this is not conjugate to the likelihood. To see why, note that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ appear together in a non-factorized way in the likelihood; hence they will also be coupled together in the posterior.

The above prior is sometimes called **semi-conjugate** or **conditionally conjugate**, since both conditionals, $p(\boldsymbol{\mu}|\boldsymbol{\Sigma})$ and $p(\boldsymbol{\Sigma}|\boldsymbol{\mu})$, are individually conjugate. To create a full conjugate prior, we need to use a prior where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are dependent on each other. We will use a joint distribution of the form

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\boldsymbol{\Sigma})p(\boldsymbol{\mu}|\boldsymbol{\Sigma}) \tag{4.199}$$

Looking at the form of the likelihood equation, Equation 4.197, we see that a natural conjugate

prior has the form of a **Normal-inverse-wishart** or **NIW** distribution, defined as follows:

$$\mathrm{NIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0) \triangleq \tag{4.200}$$

$$\mathcal{N}(\boldsymbol{\mu}|\mathbf{m}_0, \frac{1}{\kappa_0}\boldsymbol{\Sigma}) \times \mathrm{IW}(\boldsymbol{\Sigma}|\mathbf{S}_0, \nu_0) \tag{4.201}$$

$$= \frac{1}{Z_{NIW}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{\kappa_0}{2}(\boldsymbol{\mu}-\mathbf{m}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\mathbf{m}_0)\right) \tag{4.202}$$

$$\times|\boldsymbol{\Sigma}|^{-\frac{\nu_0+D+1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_0)\right) \tag{4.203}$$

$$= \frac{1}{Z_{NIW}}|\boldsymbol{\Sigma}|^{-\frac{\nu_0+D+2}{2}} \tag{4.204}$$

$$\times \exp\left(-\frac{\kappa_0}{2}(\boldsymbol{\mu}-\mathbf{m}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\mathbf{m}_0) - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}_0)\right) \tag{4.205}$$

$$Z_{NIW} = 2^{\nu_0 D/2}\Gamma_D(\nu_0/2)(2\pi/\kappa_0)^{D/2}|\mathbf{S}_0|^{-\nu_0/2} \tag{4.206}$$

where $\Gamma_D(a)$ is the multivariate Gamma function.

The parameters of the NIW can be interpreted as follows: $\mathbf{m}_0$ is our prior mean for $\boldsymbol{\mu}$, and κ_0 is how strongly we believe this prior; and $\mathbf{S}_0$ is (proportional to) our prior mean for $\boldsymbol{\Sigma}$, and ν_0 is how strongly we believe this prior.[3]

One can show (Minka 2000f) that the (improper) uninformative prior has the form

$$\lim_{k\to 0}\mathcal{N}(\mu|\mathbf{m}_0, \boldsymbol{\Sigma}/k)\mathrm{IW}(\boldsymbol{\Sigma}|\mathbf{S}_0, k) \propto |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}}|\boldsymbol{\Sigma}|^{-(D+1)/2} \tag{4.207}$$

$$\propto |\boldsymbol{\Sigma}|^{-(\frac{D}{2}+1)} \propto \mathrm{NIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{0}, 0, 0, 0\mathbf{I}) \tag{4.208}$$

In practice, it is often better to use a weakly informative data-dependent prior. A common choice (see e.g., (Chipman et al. 2001, p81), (Fraley and Raftery 2007, p6)) is to use $\mathbf{S}_0 = \mathrm{diag}(\mathbf{S}_{\overline{x}})/N$, and $\nu_0 = D+2$, to ensure $\mathbb{E}[\boldsymbol{\Sigma}] = \mathbf{S}_0$, and to set $\boldsymbol{\mu}_0 = \overline{\mathbf{x}}$ and κ_0 to some small number, such as 0.01.

3. Although this prior has four parameters, there are really only three free parameters, since our uncertainty in the mean is proportional to the variance. In particular, if we believe that the variance is large, then our uncertainty in μ must be large too. This makes sense intuitively, since if the data has large spread, it may be hard to pin down its mean. See also Exercise 9.1, where we will see the three free parameters more explicitly. If we want separate "control" over our confidence in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we must use a semi-conjugate prior.

4.6.3.3 Posterior

The posterior can be shown (Exercise 4.11) to be NIW with updated parameters:

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) = \text{NIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{m}_N, \kappa_N, \nu_N, \mathbf{S}_N) \tag{4.209}$$

$$\mathbf{m}_N = \frac{\kappa_0 \mathbf{m}_0 + N\overline{\mathbf{x}}}{\kappa_N} = \frac{\kappa_0}{\kappa_0 + N}\mathbf{m}_0 + \frac{N}{\kappa_0 + N}\overline{\mathbf{x}} \tag{4.210}$$

$$\kappa_N = \kappa_0 + N \tag{4.211}$$

$$\nu_N = \nu_0 + N \tag{4.212}$$

$$\mathbf{S}_N = \mathbf{S}_0 + \mathbf{S}_{\overline{x}} + \frac{\kappa_0 N}{\kappa_0 + N}(\overline{\mathbf{x}} - \mathbf{m}_0)(\overline{\mathbf{x}} - \mathbf{m}_0)^T \tag{4.213}$$

$$= \mathbf{S}_0 + \mathbf{S} + \kappa_0 \mathbf{m}_0 \mathbf{m}_0^T - \kappa_N \mathbf{m}_N \mathbf{m}_N^T \tag{4.214}$$

where we have defined $\mathbf{S} \triangleq \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T$ as the uncentered sum-of-squares matrix (this is easier to update incrementally than the centered version).

This result is actually quite intuitive: the posterior mean is a convex combination of the prior mean and the MLE, with "strength" $\kappa_0 + N$; and the posterior scatter matrix $\mathbf{S}_N$ is the prior scatter matrix $\mathbf{S}_0$ plus the empirical scatter matrix $\mathbf{S}_{\overline{x}}$ plus an extra term due to the uncertainty in the mean (which creates its own virtual scatter matrix).

4.6.3.4 Posterior mode

The mode of the joint distribution has the following form:

$$\text{argmax}\, p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) = (\mathbf{m}_N, \frac{\mathbf{S}_N}{\nu_N + D + 2}) \tag{4.215}$$

If we set $\kappa_0 = 0$, this reduces to

$$\text{argmax}\, p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) = (\overline{\mathbf{x}}, \frac{\mathbf{S}_0 + \mathbf{S}_{\overline{x}}}{\nu_0 + N + D + 2}) \tag{4.216}$$

The corresponding estimate $\hat{\boldsymbol{\Sigma}}$ is almost the same as Equation 4.183, but differs by 1 in the denominator, because this is the mode of the joint, not the mode of the marginal.

4.6.3.5 Posterior marginals

The posterior marginal for $\boldsymbol{\Sigma}$ is simply

$$p(\boldsymbol{\Sigma}|\mathcal{D}) = \int p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) d\boldsymbol{\mu} = \text{IW}(\boldsymbol{\Sigma}|\mathbf{S}_N, \nu_N) \tag{4.217}$$

The mode and mean of this marginal are given by

$$\hat{\boldsymbol{\Sigma}}_{map} = \frac{\mathbf{S}_N}{\nu_N + D + 1}, \quad \mathbb{E}[\boldsymbol{\Sigma}] = \frac{\mathbf{S}_N}{\nu_N - D - 1} \tag{4.218}$$

One can show that the posterior marginal for $\boldsymbol{\mu}$ has a multivariate Student T distribution:

$$p(\boldsymbol{\mu}|\mathcal{D}) = \int p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) d\boldsymbol{\Sigma} = \mathcal{T}(\boldsymbol{\mu}|\mathbf{m}_N, \frac{1}{\kappa_N(\nu_N - D + 1)}\mathbf{S}_N, \nu_N - D + 1) \tag{4.219}$$

This follows from the fact that the Student distribution can be represented as a scaled mixture of Gaussians (see Equation 11.61).

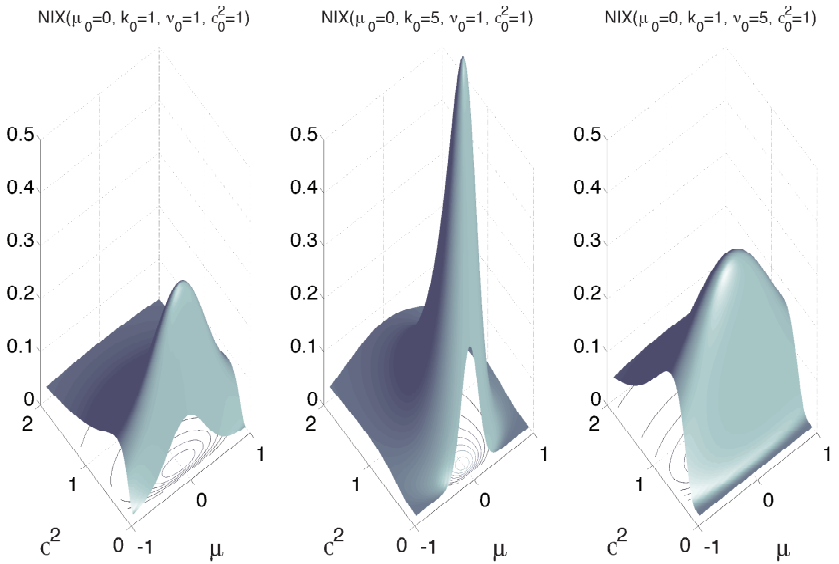


Figure 4.19 The $NI\chi^2(m_0, \kappa_0, \nu_0, \sigma_0^2)$ distribution. m_0 is the prior mean and κ_0 is how strongly we believe this; σ_0^2 is the prior variance and ν_0 is how strongly we believe this. (a) $m_0 = 0, \kappa_0 = 1, \nu_0 = 1, \sigma_0^2 = 1$. Notice that the contour plot (underneath the surface) is shaped like a "squashed egg". (b) We increase the strength of our belief in the mean, so it gets narrower: $m_0 = 0, \kappa_0 = 5, \nu_0 = 1, \sigma_0^2 = 1$. (c) We increase the strength of our belief in the variance, so it gets narrower: $m_0 = 0, \kappa_0 = 1, \nu_0 = 5, \sigma_0^2 = 1$. Figure generated by `NIXdemo2`.

4.6.3.6 Posterior predictive

The posterior predictive is given by

$$p(\mathbf{x}|\mathcal{D}) = \frac{p(\mathbf{x}, \mathcal{D})}{p(\mathcal{D})} \tag{4.220}$$

so it can be easily evaluated in terms of a ratio of marginal likelihoods.

It turns out that this ratio has the form of a multivariate Student-T distribution:

$$p(\mathbf{x}|\mathcal{D}) = \int\int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \mathrm{NIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{m}_N, \kappa_N, \nu_N, \mathbf{S}_N) d\boldsymbol{\mu} d\boldsymbol{\Sigma} \tag{4.221}$$

$$= \mathcal{T}(\mathbf{x}|\mathbf{m}_N, \frac{\kappa_N + 1}{\kappa_N(\nu_N - D + 1)}\mathbf{S}_N, \nu_N - D + 1) \tag{4.222}$$

The Student-T has wider tails than a Gaussian, which takes into account the fact that $\boldsymbol{\Sigma}$ is unknown. However, this rapidly becomes Gaussian-like.

4.6.3.7 Posterior for scalar data

We now specialise the above results to the case where x_i is 1d. These results are widely used in the statistics literature. As in Section 4.6.2.2, it is conventional not to use the normal inverse

Wishart, but to use the **normal inverse chi-squared** or **NIX** distribution, defined by

$$NI\chi^2(\mu, \sigma^2|m_0, \kappa_0, \nu_0, \sigma_0^2) \triangleq \mathcal{N}(\mu|m_0, \sigma^2/\kappa_0)\ \chi^{-2}(\sigma^2|\nu_0, \sigma_0^2) \tag{4.223}$$

$$\propto (\frac{1}{\sigma^2})^{(\nu_0+3)/2} \exp\left(-\frac{\nu_0\sigma_0^2 + \kappa_0(\mu - m_0)^2}{2\sigma^2}\right) \tag{4.224}$$

See Figure 4.19 for some plots. Along the μ axis, the distribution is shaped like a Gaussian, and along the σ^2 axis, the distribution is shaped like a χ^{-2}; the contours of the joint density have a "squashed egg" appearance. Interestingly, we see that the contours for μ are more peaked for small values of σ^2, which makes sense, since if the data is low variance, we will be able to estimate its mean more reliably.

One can show that the posterior is given by

$$p(\mu, \sigma^2|\mathcal{D}) = NI\chi^2(\mu, \sigma^2|m_N, \kappa_N, \nu_N, \sigma_N^2) \tag{4.225}$$

$$m_N = \frac{\kappa_0 m_0 + N\overline{x}}{\kappa_N} \tag{4.226}$$

$$\kappa_N = \kappa_0 + N \tag{4.227}$$

$$\nu_N = \nu_0 + N \tag{4.228}$$

$$\nu_N\sigma_N^2 = \nu_0\sigma_0^2 + \sum_{i=1}^{N}(x_i - \overline{x})^2 + \frac{N\kappa_0}{\kappa_0 + N}(m_0 - \overline{x})^2 \tag{4.229}$$

The posterior marginal for σ^2 is just

$$p(\sigma^2|\mathcal{D}) = \int p(\mu, \sigma^2|\mathcal{D})d\mu = \chi^{-2}(\sigma^2|\nu_N, \sigma_N^2) \tag{4.230}$$

with the posterior mean given by $\mathbb{E}\left[\sigma^2|\mathcal{D}\right] = \frac{\nu_N}{\nu_N - 2}\sigma_N^2$.

The posterior marginal for μ has a Student T distribution, which follows from the scale mixture representation of the student:

$$p(\mu|\mathcal{D}) = \int p(\mu, \sigma^2|D)d\sigma^2 = \mathcal{T}(\mu|m_N, \sigma_N^2/\kappa_N, \nu_N) \tag{4.231}$$

with the posterior mean given by $\mathbb{E}\left[\mu|\mathcal{D}\right] = m_N$.

Let us see how these results look if we use the following uninformative prior:

$$p(\mu, \sigma^2) \propto p(\mu)p(\sigma^2) \propto \sigma^{-2} \propto NI\chi^2(\mu, \sigma^2|\mu_0 = 0, \kappa_0 = 0, \nu_0 = -1, \sigma_0^2 = 0) \tag{4.232}$$

With this prior, the posterior has the form

$$p(\mu, \sigma^2|\mathcal{D}) = NI\chi^2(\mu, \sigma^2|m_N = \overline{x}, \kappa_N = N, \nu_N = N - 1, \sigma_N^2 = s^2) \tag{4.233}$$

where

$$s^2 \triangleq \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 = \frac{N}{N-1}\hat{\sigma}_{mle}^2 \tag{4.234}$$

is the the **sample standard deviation**. (In Section 6.4.2, we show that this is an unbiased estimate of the variance.) Hence the marginal posterior for the mean is given by

$$p(\mu|\mathcal{D}) = \mathcal{T}(\mu|\overline{x}, \frac{s^2}{N}, N - 1) \tag{4.235}$$

and the posterior variance of μ is

$$\text{var}\left[\mu|\mathcal{D}\right] = \frac{\nu_N}{\nu_N - 2}\sigma_N^2 = \frac{N-1}{N-3}\frac{s^2}{N} \rightarrow \frac{s^2}{N} \tag{4.236}$$

The square root of this is called the **standard error of the mean**:

$$\sqrt{\text{var}\left[\mu|\mathcal{D}\right]} \approx \frac{s}{\sqrt{N}} \tag{4.237}$$

Thus an approximate 95% posterior **credible interval** for the mean is

$$I_{.95}(\mu|\mathcal{D}) = \overline{x} \pm 2\frac{s}{\sqrt{N}} \tag{4.238}$$

(Bayesian credible intervals are discussed in more detail in Section 5.2.2; they are contrasted with frequentist confidence intervals in Section 6.6.1.)

4.6.3.8 Bayesian t-test

Suppose we want to test the hypothesis that $\mu \neq \mu_0$ for some known value μ_0 (often 0), given values $x_i \sim \mathcal{N}(\mu, \sigma^2)$. This is called a two-sided, one-sample **t-test**. A simple way to perform such a test is just to check if $\mu_0 \in I_{0.95}(\mu|\mathcal{D})$. If it is not, then we can be 95% sure that $\mu \neq \mu_0$.[4] A more common scenario is when we want to test if two paired samples have the same mean. More precisely, suppose $y_i \sim \mathcal{N}(\mu_1, \sigma^2)$ and $z_i \sim \mathcal{N}(\mu_2, \sigma^2)$. We want to determine if $\mu = \mu_1 - \mu_2 > 0$, using $x_i = y_i - z_i$ as our data. We can evaluate this quantity as follows:

$$p(\mu > \mu_0|\mathcal{D}) = \int_{\mu_0}^{\infty} p(\mu|\mathcal{D})d\mu \tag{4.239}$$

This is called a one-sided, **paired t-test**. (For a similar approach to unpaired tests, comparing the difference in binomial proportions, see Section 5.2.3.)

To calculate the posterior, we must specify a prior. Suppose we use an uninformative prior. As we showed above, we find that the posterior marginal on μ has the form

$$p(\mu|\mathcal{D}) = \mathcal{T}(\mu|\overline{x}, \frac{s^2}{N}, N-1) \tag{4.240}$$

Now let us define the following **t statistic**:

$$t \triangleq \frac{\overline{x} - \mu_0}{s/\sqrt{N}} \tag{4.241}$$

where the denominator is the standard error of the mean. We see that

$$p(\mu|\mathcal{D}) = 1 - F_{N-1}(t) \tag{4.242}$$

where $F_\nu(t)$ is the cdf of the standard Student t distribution $\mathcal{T}(0, 1, \nu)$.

4. A more complex approach is to perform Bayesian model comparison. That is, we compute the Bayes factor (described in Section 5.3.3) $p(\mathcal{D}|H_0)/p(\mathcal{D}|H_1)$, where H_0 is the point null hypothesis that $\mu = \mu_0$, and H_1 is the alternative hypothesis that $\mu \neq \mu_0$. See (Gonen et al. 2005; Rouder et al. 2009) for details.

4.6.3.9 Connection with frequentist statistics *

If we use an uninformative prior, it turns out that the above Bayesian analysis gives the same result as derived using frequentist methods. (We discuss frequentist statistics in Chapter 6.) Specifically, from the above results, we see that

$$\frac{\mu - \overline{x}}{\sqrt{s/N}} | \mathcal{D} \sim t_{N-1} \tag{4.243}$$

This has the same form as the sampling distribution of the MLE:

$$\frac{\mu - \overline{X}}{\sqrt{s/N}} | \mu \sim t_{N-1} \tag{4.244}$$

The reason is that the Student distribution is symmetric in its first two arguments, so $\mathcal{T}(\overline{x}|\mu, \sigma^2, \nu) = \mathcal{T}(\mu|\overline{x}, \sigma^2, \nu)$; hence statements about the posterior for μ have the same form as statements about the sampling distribution of $\overline{x}$. Consequently, the (one-sided) p-value (defined in Section 6.6.2) returned by a frequentist test is the same as $p(\mu > \mu_0|\mathcal{D})$ returned by the Bayesian method. See `bayesTtestDemo` for an example.

Despite the superficial similarity, these two results have a different interpretation: in the Bayesian approach, μ is unknown and $\overline{x}$ is fixed, whereas in the frequentist approach, $\overline{X}$ is unknown and μ is fixed. More equivalences between frequentist and Bayesian inference in simple models using uninformative priors can be found in (Box and Tiao 1973). See also Section 7.6.3.3.

4.6.4 Sensor fusion with unknown precisions *

In this section, we apply the results in Section 4.6.3 to the problem of sensor fusion in the case where the precision of each measurement device is unknown. This generalizes the results of Section 4.4.2.2, where the measurement model was assumed to be Gaussian with known precision. The unknown precision case turns out to give qualitatively different results, yielding a potentially multi-modal posterior as we will see. Our presentation is based on (Minka 2001e).

Suppose we want to pool data from multiple sources to estimate some quantity $\mu \in \mathbb{R}$, but the reliability of the sources is unknown. Specifically, suppose we have two different measurement devices, x and y, with different precisions: $x_i|\mu \sim \mathcal{N}(\mu, \lambda_x^{-1})$ and $y_i|\mu \sim \mathcal{N}(\mu, \lambda_y^{-1})$. We make two independent measurements with each device, which turn out to be

$$x_1 = 1.1, x_2 = 1.9, y_1 = 2.9, y_2 = 4.1 \tag{4.245}$$

We will use a non-informative prior for μ, $p(\mu) \propto 1$, which we can emulate using an infinitely broad Gaussian, $p(\mu) = \mathcal{N}(\mu|m_0 = 0, \lambda_0^{-1} = \infty)$. If the λ_x and λ_y terms were known, then the posterior would be Gaussian:

$$p(\mu|\mathcal{D}, \lambda_x, \lambda_y) = \mathcal{N}(\mu|m_N, \lambda_N^{-1}) \tag{4.246}$$

$$\lambda_N = \lambda_0 + N_x\lambda_x + N_y\lambda_y \tag{4.247}$$

$$m_N = \frac{\lambda_x N_x \overline{x} + \lambda_y N_y \overline{y}}{N_x\lambda_x + N_y\lambda_y} \tag{4.248}$$

where $N_x = 2$ is the number of x measurements, $N_y = 2$ is the number of y measurements, $\overline{x} = \frac{1}{N_x}\sum_{i=1}^{N_x} x_i = 1.5$ and $\overline{y} = \frac{1}{N_y}\sum_{i=1}^{N_y} y_i = 3.5$. This result follows because the posterior precision is the sum of the measurement precisions, and the posterior mean is a weighted sum of the prior mean (which is 0) and the data means.

However, the measurement precisions are not known. Initially we will estimate them by maximum likelihood. The log-likelihood is given by

$$\ell(\mu, \lambda_x, \lambda_y) = \log \lambda_x - \frac{\lambda_x}{2}\sum_i (x_i - \mu)^2 + \log \lambda_y - \frac{\lambda_y}{2}\sum_i (y_i - \mu)^2 \tag{4.249}$$

The MLE is obtained by solving the following simultaneous equations:

$$\frac{\partial \ell}{\partial \mu} = \lambda_x N_x(\overline{x} - \mu) + \lambda_y N_y(\overline{y} - \mu) = 0 \tag{4.250}$$

$$\frac{\partial \ell}{\partial \lambda_x} = \frac{1}{\lambda_x} - \frac{1}{N_x}\sum_{i=1}^{N_x}(x_i - \mu)^2 = 0 \tag{4.251}$$

$$\frac{\partial \ell}{\partial \lambda_y} = \frac{1}{\lambda_y} - \frac{1}{N_y}\sum_{i=1}^{N_y}(y_i - \mu)^2 = 0 \tag{4.252}$$

This gives

$$\hat{\mu} = \frac{N_x \hat{\lambda}_x \overline{x} + N_y \hat{\lambda}_y \overline{y}}{N_x \hat{\lambda}_x + N_y \hat{\lambda}_y} \tag{4.253}$$

$$1/\hat{\lambda}_x = \frac{1}{N_x}\sum_i (x_i - \hat{\mu})^2 \tag{4.254}$$

$$1/\hat{\lambda}_y = \frac{1}{N_y}\sum_i (y_i - \hat{\mu})^2 \tag{4.255}$$

We notice that the MLE for μ has the same form as the posterior mean, m_N.

We can solve these equations by fixed point iteration. Let us initialize by estimating $\lambda_x = 1/s_x^2$ and $\lambda_y = 1/s_y^2$, where $s_x^2 = \frac{1}{N_x}\sum_{i=1}^{N_x}(x_i - \overline{x})^2 = 0.16$ and $s_y^2 = \frac{1}{N_y}\sum_{i=1}^{N_y}(y_i - \overline{y})^2 = 0.36$. Using this, we get $\hat{\mu} = 2.1154$, so $p(\mu|\mathcal{D}, \hat{\lambda}_x, \hat{\lambda}_y) = \mathcal{N}(\mu|2.1154, 0.0554)$. If we now iterate, we converge to $\hat{\lambda}_x = 1/0.1662$, $\hat{\lambda}_y = 1/4.0509$, $p(\mu|\mathcal{D}, \hat{\lambda}_x, \hat{\lambda}_y) = \mathcal{N}(\mu|1.5788, 0.0798)$.

The plug-in approximation to the posterior is plotted in Figure 4.20(a). This weights each sensor according to its estimated precision. Since sensor y was estimated to be much less reliable than sensor x, we have $\mathbb{E}\left[\mu|\mathcal{D}, \hat{\lambda}_x, \hat{\lambda}_y\right] \approx \overline{x}$, so we effectively ignore the y sensor.

Now we will adopt a Bayesian approach and integrate out the unknown precisions, rather than trying to estimate them. That is, we compute

$$p(\mu|\mathcal{D}) \propto p(\mu)\left[\int p(\mathcal{D}_x|\mu, \lambda_x)p(\lambda_x|\mu)d\lambda_x\right]\left[\int p(\mathcal{D}_y|\mu, \lambda_y)p(\lambda_y|\mu)d\lambda_y\right] \tag{4.256}$$

We will use uninformative Jeffrey's priors, $p(\mu) \propto 1$, $p(\lambda_x|\mu) \propto 1/\lambda_x$ and $p(\lambda_y|\mu) \propto 1/\lambda_y$.

Since the x and y terms are symmetric, we will just focus on one of them. The key integral is

$$I = \int p(\mathcal{D}_x|\mu, \lambda_x)p(\lambda_x|\mu)d\lambda_x \quad \propto \quad \int \lambda_x^{-1}(N_x\lambda_x)^{N_x/2} \tag{4.257}$$

$$\exp\left(-\frac{N_x}{2}\lambda_x(\overline{x}-\mu)^2 - \frac{N_x}{2}s_x^2\lambda_x\right)d\lambda_x \tag{4.258}$$

Exploiting the fact that $N_x = 2$ this simplifies to

$$I = \int \lambda_x^{-1}\lambda_x^1 \exp(-\lambda_x[(\overline{x}-\mu)^2 + s_x^2])d\lambda_x \tag{4.259}$$

We recognize this as proportional to the integral of an unnormalized Gamma density

$$\mathrm{Ga}(\lambda|a,b) \propto \lambda^{a-1}e^{-\lambda b} \tag{4.260}$$

where $a = 1$ and $b = (\overline{x}-\mu)^2 + s_x^2$. Hence the integral is proportional to the normalizing constant of the Gamma distribution, $\Gamma(a)b^{-a}$, so we get

$$I \propto \int p(\mathcal{D}_x|\mu, \lambda_x)p(\lambda_x|\mu)d\lambda_x \quad \propto \quad \left((\overline{x}-\mu)^2 + s_x^2\right)^{-1} \tag{4.261}$$

and the posterior becomes

$$p(\mu|\mathcal{D}) \propto \frac{1}{(\overline{x}-\mu)^2 + s_x^2} \quad \frac{1}{(\overline{y}-\mu)^2 + s_y^2} \tag{4.262}$$

The exact posterior is plotted in Figure 4.20(b). We see that it has two modes, one near $\overline{x} = 1.5$ and one near $\overline{y} = 3.5$. These correspond to the beliefs that the x sensor is more reliable than the y one, and vice versa. The weight of the first mode is larger, since the data from the x sensor agree more with each other, so it seems slightly more likely that the x sensor is the reliable one. (They obviously cannot both be reliable, since they disagree on the values that they are reporting.) However, the Bayesian solution keeps open the possibility that the y sensor is the more reliable one; from two measurements, we cannot tell, and choosing just the x sensor, as the plug-in approximation does, results in over confidence (a posterior that is too narrow).

Exercises

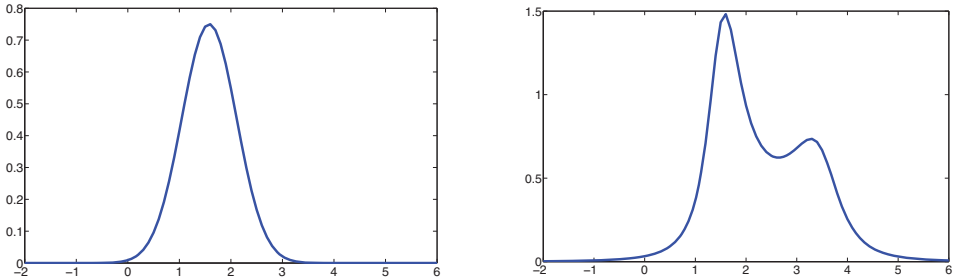
Exercise 4.1 Uncorrelated does not imply independent

Let $X \sim U(-1, 1)$ and $Y = X^2$. Clearly Y is dependent on X (in fact, Y is uniquely determined by X). However, show that $\rho(X, Y) = 0$. Hint: if $X \sim U(a, b)$ then $E[X] = (a + b)/2$ and $\mathrm{var}[X] = (b - a)^2/12$.

Exercise 4.2 Uncorrelated and Gaussian does not imply independent unless *jointly* Gaussian

Let $X \sim \mathcal{N}(0, 1)$ and $Y = WX$, where $p(W = -1) = p(W = 1) = 0.5$. It is clear that X and Y are not independent, since Y is a function of X.

a. Show $Y \sim \mathcal{N}(0, 1)$.



(a) (b)

Figure 4.20 Posterior for μ. (a) Plug-in approximation. (b) Exact posterior. Figure generated by `sensorFusionUnknownPrec`.

b. Show $\text{cov}\,[X, Y] = 0$. Thus X and Y are uncorrelated but dependent, even though they are Gaussian. Hint: use the definition of covariance

$$\text{cov}\,[X, Y] = \mathbb{E}\,[XY] - \mathbb{E}\,[X]\,\mathbb{E}\,[Y] \tag{4.263}$$

and the **rule of iterated expectation**

$$\mathbb{E}\,[XY] = \mathbb{E}\,[\mathbb{E}\,[XY|W]] \tag{4.264}$$

Exercise 4.3 Correlation coefficient is between -1 and +1

Prove that $-1 \leq \rho(X, Y) \leq 1$

Exercise 4.4 Correlation coefficient for linearly related variables is ± 1

Show that, if $Y = aX + b$ for some parameters $a > 0$ and b, then $\rho(X, Y) = 1$. Similarly show that if $a < 0$, then $\rho(X, Y) = -1$.

Exercise 4.5 Normalization constant for a multidimensional Gaussian

Prove that the normalization constant for a d-dimensional Gaussian is given by

$$(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{\frac{1}{2}} = \int \exp(-\frac{1}{2}(\mathbf{x} - \mu)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}))d\mathbf{x} \tag{4.265}$$

Hint: diagonalize $\boldsymbol{\Sigma}$ and use the fact that $|\boldsymbol{\Sigma}| = \prod_i \lambda_i$ to write the joint pdf as a product of d one-dimensional Gaussians in a transformed coordinate system. (You will need the change of variables formula.) Finally, use the normalization constant for univariate Gaussians.

Exercise 4.6 Bivariate Gaussian

Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\mathbf{x} \in \mathbb{R}^2$ and

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{4.266}$$

where ρ is the correlation coefficient. Show that the pdf is given by

$$p(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \tag{4.267}$$

$$\exp\left(-\frac{1}{2(1-\rho^2)}\left(\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} - 2\rho\frac{(x_1-\mu_1)}{\sigma_1}\frac{(x_2-\mu_2)}{\sigma_2}\right)\right) \tag{4.268}$$

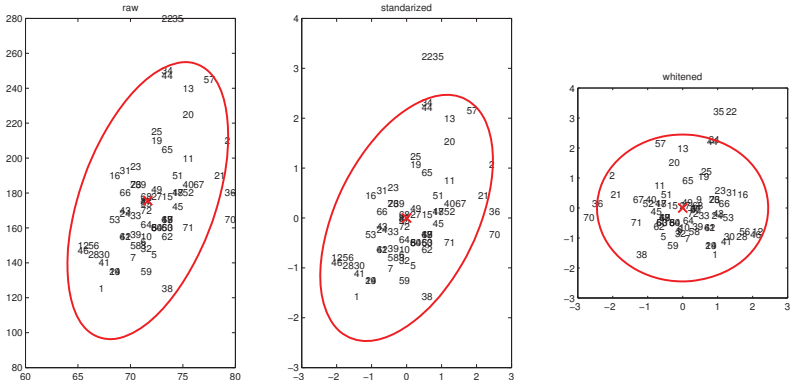


Figure 4.21 (a) Height/weight data for the men. (b) Standardized. (c) Whitened.

Exercise 4.7 Conditioning a bivariate Gaussian

Consider a bivariate Gaussian distribution $p(x_1, x_2) = \mathcal{N}(x|\mu, \Sigma)$ where

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix} = \sigma_1\sigma_2 \begin{pmatrix} \frac{\sigma_1}{\sigma_2} & \rho \\ \rho & \frac{\sigma_2}{\sigma_1} \end{pmatrix} \tag{4.269}$$

where the correlation coefficient is given by

$$\rho \triangleq \frac{\sigma_{12}}{\sigma_1\sigma_2} \tag{4.270}$$

a. What is $P(X_2|x_1)$? Simplify your answer by expressing it in terms of ρ, σ_2, σ_1, μ_1,μ_2 and x_1.

b. Assume $\sigma_1 = \sigma_2 = 1$. What is $P(X_2|x_1)$ now?

Exercise 4.8 Whitening vs standardizing

a. Load the height/weight data using `rawdata = dlmread('heightWeightData.txt')`. The first column is the class label (1=male, 2=female), the second column is height, the third weight. Extract the height/weight data corresponding to the males. Fit a 2d Gaussian to the male data, using the empirical mean and covariance. Plot your Gaussian as an ellipse (use `gaussPlot2d`), superimposing on your scatter plot. It should look like Figure 4.21(a), where have labeled each datapoint by its index. Turn in your figure and code.

b. **Standardizing** the data means ensuring the empirical variance along each dimension is 1. This can be done by computing $\frac{x_{ij}-\overline{x}_j}{\sigma_j}$, where σ_j is the empirical std of dimension j. Standardize the data and replot. It should look like Figure 4.21(b). (Use `axis('equal')`.) Turn in your figure and code.

c. **Whitening** or **sphereing** the data means ensuring its empirical covariance matrix is proportional to $\mathbf{I}$, so the data is uncorrelated and of equal variance along each dimension. This can be done by computing $\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{x}$ for each data vector $\mathbf{x}$, where $\mathbf{U}$ are the eigenvectors and $\mathbf{\Lambda}$ the eigenvalues of $\mathbf{X}$. Whiten the data and replot. It should look like Figure 4.21(c). Note that whitening rotates the data, so people move to counter-intuitive locations in the new coordinate system (see e.g., person 2, who moves from the right hand side to the left).

Exercise 4.9 Sensor fusion with known variances in 1d

Suppose we have two sensors with known (and different) variances v_1 and v_2, but unknown (and the same) mean μ. Suppose we observe n_1 observations $y_i^{(1)} \sim \mathcal{N}(\mu, v_1)$ from the first sensor and n_2 observations

$y_i^{(2)} \sim \mathcal{N}(\mu, v_2)$ from the second sensor. (For example, suppose μ is the true temperature outside, and sensor 1 is a precise (low variance) digital thermosensing device, and sensor 2 is an imprecise (high variance) mercury thermometer.) Let $\mathcal{D}$ represent all the data from both sensors. What is the posterior $p(\mu|\mathcal{D})$, assuming a non-informative prior for μ (which we can simulate using a Gaussian with a precision of 0)? Give an explicit expression for the posterior mean and variance.

Exercise 4.10 Derivation of information form formulae for marginalizing and conditioning

Derive the information form results of Section 4.3.1.

Exercise 4.11 Derivation of the NIW posterior

Derive Equation 4.209. Hint: one can show that

$$N(\overline{\mathbf{x}} - \boldsymbol{\mu})(\overline{\mathbf{x}} - \boldsymbol{\mu})^T + \kappa_0(\boldsymbol{\mu} - \mathbf{m}_0)(\boldsymbol{\mu} - \mathbf{m}_0)^T \tag{4.271}$$

$$= \kappa_N(\boldsymbol{\mu} - \mathbf{m}_N)(\boldsymbol{\mu} - \mathbf{m}_N)^T + \frac{\kappa_0 N}{\kappa_N}(\overline{\mathbf{x}} - \mathbf{m}_0)(\overline{\mathbf{x}} - \mathbf{m}_0)^T \tag{4.272}$$

This is a matrix generalization of an operation called **completing the square**.[5]

Derive the corresponding result for the normal-Wishart model.

Exercise 4.12 BIC for Gaussians

(Source: Jaakkola.)

The Bayesian information criterion (BIC) is a penalized log-likelihood function that can be used for model selection (see Section 5.3.2.4). It is defined as

$$BIC = \log p(\mathcal{D}|\hat{\boldsymbol{\theta}}_{ML}) - \frac{d}{2}\log(N) \tag{4.273}$$

where d is the number of free parameters in the model and N is the number of samples. In this question, we will see how to use this to choose between a full covariance Gaussian and a Gaussian with a diagonal covariance. Obviously a full covariance Gaussian has higher likelihood, but it may not be "worth" the extra parameters if the improvement over a diagonal covariance matrix is too small. So we use the BIC score to choose the model.

Following Section 4.1.3, we can write

$$\log p(\mathcal{D}|\hat{\boldsymbol{\Sigma}}, \hat{\boldsymbol{\mu}}) = -\frac{N}{2}\mathrm{tr}\left(\hat{\boldsymbol{\Sigma}}^{-1}\hat{\mathbf{S}}\right) - \frac{N}{2}\log(|\hat{\boldsymbol{\Sigma}}|) \tag{4.274}$$

$$\hat{\mathbf{S}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T \tag{4.275}$$

where $\hat{\mathbf{S}}$ is the scatter matrix (empirical covariance), the trace of a matrix is the sum of its diagonals, and we have used the trace trick.

a. Derive the BIC score for a Gaussian in D dimensions with full covariance matrix. Simplify your answer as much as possible, exploiting the form of the MLE. Be sure to specify the number of free parameters d.

b. Derive the BIC score for a Gaussian in D dimensions with a *diagonal* covariance matrix. Be sure to specify the number of free parameters d. Hint: for the digaonal case, the ML estimate of $\boldsymbol{\Sigma}$ is the same as $\hat{\boldsymbol{\Sigma}}_{ML}$ except the off-diagonal terms are zero:

$$\hat{\boldsymbol{\Sigma}}_{diag} = \mathrm{diag}(\hat{\boldsymbol{\Sigma}}_{ML}(1,1), \ldots, \hat{\boldsymbol{\Sigma}}_{ML}(D,D)) \tag{4.276}$$

5. In the scalar case, completing the square means rewriting $c_2x^2 + c_1x + c_0$ as $-a(x-b)^2 + w$ where $a = -c_2$, $b = \frac{c_1}{2c_2}$ and $w = \frac{c_1^2}{4c_2} + c_0$.

Exercise 4.13 Gaussian posterior credible interval

(Source: DeGroot.)

Let $X \sim \mathcal{N}(\mu, \sigma^2 = 4)$ where μ is unknown but has prior $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2 = 9)$. The posterior after seeing n samples is $\mu \sim \mathcal{N}(\mu_n, \sigma_n^2)$. (This is called a credible interval, and is the Bayesian analog of a confidence interval.) How big does n have to be to ensure

$$p(\ell \leq \mu_n \leq u|D) \geq 0.95 \tag{4.277}$$

where (ℓ, u) is an interval (centered on μ_n) of width 1 and D is the data. Hint: recall that 95% of the probability mass of a Gaussian is within $\pm 1.96\sigma$ of the mean.

Exercise 4.14 MAP estimation for 1D Gaussians

(Source: Jaakkola.)

Consider samples $x_1, \ldots, x_n$ from a Gaussian random variable with known variance σ^2 and unknown mean μ. We further assume a prior distribution (also Gaussian) over the mean, $\mu \sim \mathcal{N}(m, s^2)$, with fixed mean m and fixed variance s^2. Thus the only unknown is μ.

a. Calculate the MAP estimate $\hat{\mu}_{MAP}$. You can state the result without proof. Alternatively, with a lot more work, you can compute derivatives of the log posterior, set to zero and solve.
b. Show that as the number of samples n increase, the MAP estimate converges to the maximum likelihood estimate.
c. Suppose n is small and fixed. What does the MAP estimator converge to if we increase the prior variance s^2?
d. Suppose n is small and fixed. What does the MAP estimator converge to if we decrease the prior variance s^2?

Exercise 4.15 Sequential (recursive) updating of $\hat{\mathbf{\Sigma}}$

(Source: (Duda et al. 2001, Q3.35,3.36).)

The unbiased estimates for the covariance of a d-dimensional Gaussian based on n samples is given by

$$\hat{\mathbf{\Sigma}} = \mathbf{C}_n \quad = \quad \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{m}_n)(\mathbf{x}_i - \mathbf{m}_n)^T \tag{4.278}$$

It is clear that it takes $O(nd^2)$ time to compute $\mathbf{C}_n$. If the data points arrive one at a time, it is more efficient to incrementally update these estimates than to recompute from scratch.

a. Show that the covariance can be sequentially udpated as follows

$$\mathbf{C}_{n+1} = \frac{n-1}{n} \mathbf{C}_n + \frac{1}{n+1} (\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T \tag{4.279}$$

b. How much time does it take per sequential update? (Use big-O notation.)
c. Show that we can sequentially update the precision matrix using

$$\mathbf{C}_{n+1}^{-1} = \frac{n}{n-1} \left[\mathbf{C}_n^{-1} - \frac{\mathbf{C}_n^{-1} (\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T \mathbf{C}_n^{-1}}{\frac{n^2-1}{n} + (\mathbf{x}_{n+1} - \mathbf{m}_n)^T \mathbf{C}_n^{-1} (\mathbf{x}_{n+1} - \mathbf{m}_n)} \right] \tag{4.280}$$

Hint: notice that the update to $\mathbf{C}_{n+1}$ consists of adding a rank-one matrix, namely $\mathbf{u}\mathbf{u}^T$, where $\mathbf{u} = \mathbf{x}_{n+1} - \mathbf{m}_n$. Use the matrix inversion lemma for rank-one updates (Equation 4.111), which we repeat here for convenience:

$$(\mathbf{E} + \mathbf{u}\mathbf{v}^T)^{-1} \quad = \quad \mathbf{E}^{-1} - \frac{\mathbf{E}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{E}^{-1}}{1 + \mathbf{v}^T\mathbf{E}^{-1}\mathbf{u}} \tag{4.281}$$

d. What is the time complexity per update?

Exercise 4.16 Likelihood ratio for Gaussians

Source: Source: Alpaydin p103 ex 4. Consider a binary classifier where the K class conditional densities are MVN $p(x|y=j) = \mathcal{N}(x|\mu_j, \Sigma_j)$. By Bayes rule, we have

$$\log \frac{p(y=1|x)}{p(y=0|x)} = \log \frac{p(x|y=1)}{p(x|y=0)} + \log \frac{p(y=1)}{p(y=0)} \tag{4.282}$$

In other words, the log posterior ratio is the log likelihood ratio plus the log prior ratio. For each of the 4 cases in the table below, derive an expression for the log likelihood ratio $\log \frac{p(x|y=1)}{p(x|y=0)}$, simplifying as much as possible.

Form of Σ_j	Cov	Num parameters
Arbitrary	Σ_j	$Kd(d+1)/2$
Shared	$\Sigma_j = \Sigma$	$d(d+1)/2$
Shared, axis-aligned	$\Sigma_j = \Sigma$ with $\Sigma_{ij} = 0$ for $i \neq j$	d
Shared, spherical	$\Sigma_j = \sigma^2 I$	1

Exercise 4.17 LDA/QDA on height/weight data

The function `discrimAnalysisHeightWeightDemo` fits an LDA and QDA model to the height/weight data. Compute the misclassification rate of both of these models on the training set. Turn in your numbers and code.

Exercise 4.18 Naive Bayes with mixed features

Consider a 3 class naive Bayes classifier with one binary feature and one Gaussian feature:

$$y \sim \text{Mu}(y|\boldsymbol{\pi}, 1),\ x_1|y=c \sim \text{Ber}(x_1|\theta_c),\ x_2|y=c \sim \mathcal{N}(x_2|\mu_c, \sigma_c^2) \tag{4.283}$$

Let the parameter vectors be as follows:

$$\boldsymbol{\pi} = (0.5, 0.25, 0.25),\ \boldsymbol{\theta} = (0.5, 0.5, 0.5),\ \boldsymbol{\mu} = (-1, 0, 1),\ \boldsymbol{\sigma}^2 = (1, 1, 1) \tag{4.284}$$

a. Compute $p(y|x_1 = 0, x_2 = 0)$ (the result should be a vector of 3 numbers that sums to 1).

b. Compute $p(y|x_1 = 0)$.

c. Compute $p(y|x_2 = 0)$.

d. Explain any interesting patterns you see in your results. Hint: look at the parameter vector θ.

Exercise 4.19 Decision boundary for LDA with semi tied covariances

Consider a generative classifier with class conditional densities of the form $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$. In LDA, we assume $\boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}$, and in QDA, each $\boldsymbol{\Sigma}_c$ is arbitrary. Here we consider the 2 class case in which $\boldsymbol{\Sigma}_1 = k\boldsymbol{\Sigma}_0$, for $k > 1$. That is, the Gaussian ellipsoids have the same "shape", but the one for class 1 is "wider". Derive an expression for $p(y = 1|\mathbf{x}, \boldsymbol{\theta})$, simplifying as much as possible. Give a geometric interpretation of your result, if possible.

Exercise 4.20 Logistic regression vs LDA/QDA

(Source: Jaakkola.) Suppose we train the following binary classifiers via maximum likelihood.

a. GaussI: A generative classifier, where the class conditional densities are Gaussian, with both covariance matrices set to $\mathbf{I}$ (identity matrix), i.e., $p(\mathbf{x}|y=c) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \mathbf{I})$. We assume $p(y)$ is uniform.

b. GaussX: as for GaussI, but the covariance matrices are unconstrained, i.e., $p(\mathbf{x}|y=c) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$.

c. LinLog: A logistic regression model with linear features.

d. QuadLog: A logistic regression model, using linear and quadratic features (i.e., polynomial basis function expansion of degree 2).

After training we compute the performance of each model M on the training set as follows:

$$L(M) = \frac{1}{n}\sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i, \hat{\boldsymbol{\theta}}, M) \tag{4.285}$$

(Note that this is the *conditional* log-likelihood $p(y|\mathbf{x}, \hat{\boldsymbol{\theta}})$ and not the joint log-likelihood $p(y, \mathbf{x}|\hat{\boldsymbol{\theta}})$.) We now want to compare the performance of each model. We will write $L(M) \leq L(M')$ if model M *must* have lower (or equal) log likelihood (on the training set) than M', for any training set (in other words, M is worse than M', at least as far as training set logprob is concerned). For each of the following model pairs, state whether $L(M) \leq L(M')$, $L(M) \geq L(M')$, or whether no such statement can be made (i.e., M might sometimes be better than M' and sometimes worse); also, for each question, briefly (1-2 sentences) explain why.

a. GaussI, LinLog.

b. GaussX, QuadLog.

c. LinLog, QuadLog.

d. GaussI, QuadLog.

e. Now suppose we measure performance in terms of the average misclassification rate on the training set:

$$R(M) = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}(\mathbf{x}_i)) \tag{4.286}$$

Is it true in general that $L(M) > L(M')$ implies that $R(M) < R(M')$? Explain why or why not.

Exercise 4.21 Gaussian decision boundaries

(Source: (Duda et al. 2001, Q3.7).) Let $p(x|y = j) = \mathcal{N}(x|\mu_j, \sigma_j)$ where $j = 1, 2$ and $\mu_1 = 0, \sigma_1^2 = 1, \mu_2 = 1, \sigma_2^2 = 10^6$. Let the class priors be equal, $p(y = 1) = p(y = 2) = 0.5$.

a. Find the decision region

$$R_1 = \{x : p(x|\mu_1, \sigma_1) \geq p(x|\mu_2, \sigma_2)\} \tag{4.287}$$

Sketch the result. Hint: draw the curves and find where they intersect. Find *both* solutions of the equation

$$p(x|\mu_1, \sigma_1) = p(x|\mu_2, \sigma_2) \tag{4.288}$$

Hint: recall that to solve a quadratic equation $ax^2 + bx + c = 0$, we use

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{4.289}$$

b. Now suppose $\sigma_2 = 1$ (and all other parameters remain the same). What is R_1 in this case?

Exercise 4.22 QDA with 3 classes

Consider a three category classification problem. Let the prior probabilites:

$$P(Y=1) = P(Y=2) = P(Y=3) = 1/3 \tag{4.290}$$

The class-conditional densities are multivariate normal densities with parameters:

$$\mu_1 = [0,0]^T, \mu_2 = [1,1]^T, \mu_3 = [-1,1]^T \tag{4.291}$$

$$\Sigma_1 = \begin{bmatrix} 0.7 & 0 \\ 0 & 0.7 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix} \tag{4.292}$$

Classify the following points:

a. $\mathbf{x} = [-0.5, 0.5]$
b. $\mathbf{x} = [0.5, 0.5]$

Exercise 4.23 Scalar QDA

[Note: you can solve this exercise by hand or using a computer (matlab, R, whatever). In either case, show your work.] Consider the following training set of heights x (in inches) and gender y (male/female) of some US college students: $\mathbf{x} = (67, 79, 71, 68, 67, 60)$, $\mathbf{y} = (m, m, m, f, f, f)$.

a. Fit a Bayes classifier to this data, using maximum likelihood estimation, i.e., estimate the parameters of the class conditional likelihoods

$$p(x|y=c) = \mathcal{N}(x; \mu_c, \sigma_c) \tag{4.293}$$

and the class prior

$$p(y=c) = \pi_c \tag{4.294}$$

What are your values of μ_c, σ_c, π_c for $c = m, f$? Show your work (so you can get partial credit if you make an arithmetic error).

b. Compute $p(y = m|x, \hat{\theta})$, where $x = 72$, and $\hat{\theta}$ are the MLE parameters. (This is called a plug-in prediction.)

c. What would be a simple way to extend this technique if you had multiple attributes per person, such as height and weight? Write down your proposed model as an equation.