Chapter 4
Gaussian Models

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.

4.1 Basics

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

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

The expression inside the exponent is the Mahalanobis distance between a data vector $\boldsymbol{x}$ 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 $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{U}$ is an orthonormal matrix of eigenvectors satsifying $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues. Using the eigendecomposition, we have that

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

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

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

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

$$= \sum_{i=1}^{D}\frac{y_i^2}{\lambda_i} \tag{4.5}$$

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

$$\frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} = 1 \tag{4.6}$$

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.

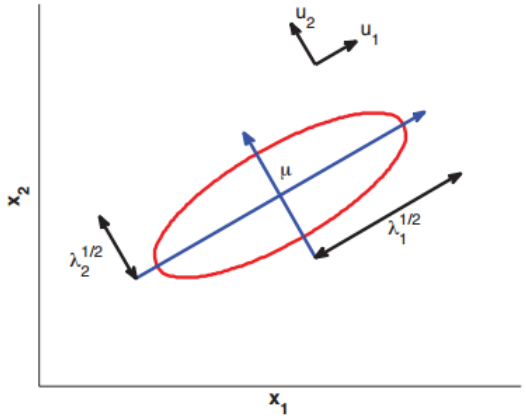


Fig. 4.1: Visualization of a 2 dimensional Gaussian density. The major and minor axes of the ellipse are defined by the first two eigenvectors of the covariance matrix, namely $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. Based on Figure 2.7 of (Bishop 2006a)

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 $\boldsymbol{U}$.

4.1.1 MLE for a MVN

Theorem 4.1. *(**MLE for a MVN**) If we have N iid samples $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, then the MLE for the parameters is given by*

$$\bar{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \triangleq \bar{\boldsymbol{x}} \tag{4.7}$$

$$\bar{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \tag{4.8}$$

$$= \frac{1}{N}\left(\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T\right) - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^T \tag{4.9}$$

4.1.2 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 TODO). 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

$$f(\boldsymbol{x}) = \frac{1}{Z}\exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x}\right) \tag{4.10}$$

4.2 Gaussian discriminant analysis

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

$$p(\boldsymbol{x}|y=c,\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \tag{4.11}$$

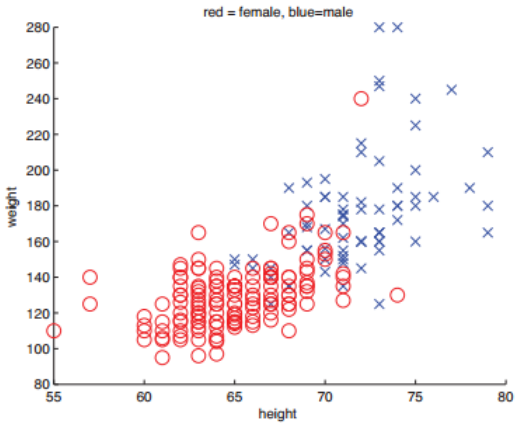
The resulting technique is called (Gaussian) **discriminant analysis** or **GDA** (even though it is a generative, not discriminative, classifier see Section TODO for more on this distinction). If $\boldsymbol{\Sigma}_c$ is diagonal, this is equivalent to naive Bayes.

We can classify a feature vector using the following decision rule, derived from Equation 3.1:

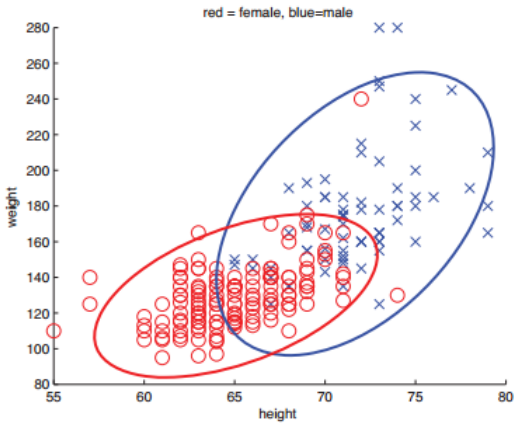
$$y = \arg\max_c \left[\log p(y=c|\boldsymbol{\pi}) + \log p(\boldsymbol{x}|\boldsymbol{\theta})\right] \tag{4.12}$$

When we compute the probability of $\boldsymbol{x}$ under each class conditional density, we are measuring the distance from $\boldsymbol{x}$ 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



(a)



(b)

Fig. 4.2: (a) Height/weight data. (b) Visualization of 2d Gaussians fit to each class. 95% of the probability mass is inside the ellipse.

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:

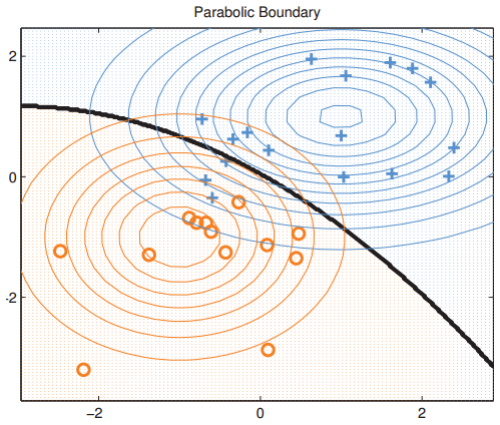
$$y = \arg\max_c (\boldsymbol{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_c) \tag{4.13}$$

4.2.1 Quadratic discriminant analysis (QDA)

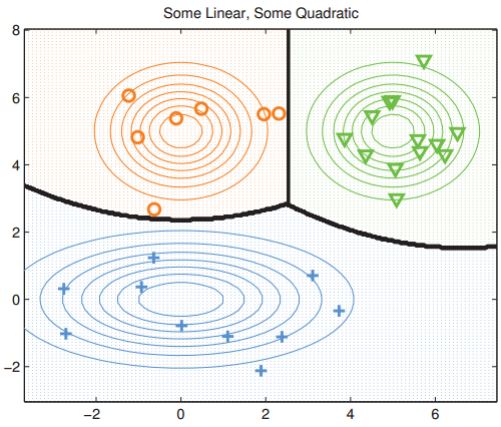
By plugging in the definition of the Gaussian density to Equation 3.1, we can get

$$p(y|\boldsymbol{x},\boldsymbol{\theta}) = \frac{\pi_c |2\pi\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]}{\sum_{c'} \pi_{c'} |2\pi\boldsymbol{\Sigma}_{c'}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]} \tag{4.14}$$

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



(a)



(b)

Fig. 4.3: Quadratic decision boundaries in 2D for the 2 and 3 class case.

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 4.14 as follows:

$$\begin{aligned} p(y|\boldsymbol{x},\boldsymbol{\theta}) &\propto \pi_c \exp\left(\boldsymbol{\mu}_c \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\right) \\ &= \exp\left(\boldsymbol{\mu}_c \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c\right) \\ &\quad \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}\right) \\ &\propto \exp\left(\boldsymbol{\mu}_c \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c\right) \end{aligned} \tag{4.15}$$

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

$$\gamma_c \triangleq -\frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c \tag{4.16}$$

$$\boldsymbol{\beta}_c \triangleq \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c \tag{4.17}$$

then we can write

$$p(y|\boldsymbol{x},\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_c^T\boldsymbol{x}+\gamma_c}}{\sum_{c'} e^{\boldsymbol{\beta}_{c'}^T\boldsymbol{x}+\gamma_{c'}}} \triangleq \sigma(\boldsymbol{\eta}, c) \tag{4.18}$$

where $\boldsymbol{\eta} \triangleq (e^{\boldsymbol{\beta}_1^T\boldsymbol{x}} + \gamma_1, \cdots, e^{\boldsymbol{\beta}_C^T\boldsymbol{x}} + \gamma_C)$, $\sigma()$ is the **softmax activation function**[16], defined as follows:

$$\sigma(\boldsymbol{q}, i) \triangleq \frac{\exp(q_i)}{\sum_{j=1}^n \exp(q_j)} \tag{4.19}$$

When parameterized by some constant, $\alpha > 0$, the following formulation becomes a smooth, differentiable approximation of the maximum function:

$$\mathcal{S}_\alpha(\boldsymbol{x}) = \frac{\sum_{j=1}^D x_j e^{\alpha x_j}}{\sum_{j=1}^D e^{\alpha x_j}} \tag{4.20}$$

$\mathcal{S}_\alpha$ has the following properties:

1. $\mathcal{S}_\alpha \to \max$ as $\alpha \to \infty$
2. $\mathcal{S}_0$ is the average of its inputs
3. $\mathcal{S}_\alpha \to \min$ as $\alpha \to -\infty$

Note that the softmax activation function comes from the area of statistical physics, where it is common to use the **Boltzmann distribution**, which has the same form as the softmax activation function.

An interesting property of Equation 4.18 is that, if we take logs, we end up with a linear function of $\boldsymbol{x}$. (The reason it is linear is because the $\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$ cancels from the numerator and denominator.) Thus the decision boundary between any two classes, says c and c', will be a straight line. Hence this technique is called **linear discriminant analysis** or **LDA**.

[16] http://en.wikipedia.org/wiki/Softmax_activation_function

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

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

$$p(y=1|\boldsymbol{x},\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_1^T\boldsymbol{x}+\gamma_1}}{e^{\boldsymbol{\beta}_0^T\boldsymbol{x}+\gamma_0}+e^{\boldsymbol{\beta}_1^T\boldsymbol{x}+\gamma_1}}) \tag{4.21}$$

$$= \frac{1}{1+e^{(\boldsymbol{\beta}_0-\boldsymbol{\beta}_1)^T\boldsymbol{x}+(\gamma_0-\gamma_1)}} \tag{4.22}$$

$$= \text{sigm}((\boldsymbol{\beta}_1-\boldsymbol{\beta}_0)^T\boldsymbol{x}+(\gamma_0-\gamma_1)) \tag{4.23}$$

where $\text{sigm}(x)$ refers to the sigmoid function[17].

Now

$$\gamma_1-\gamma_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1+\frac{1}{2}\boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0+\log(\pi_1/\pi_0) \tag{4.24}$$

$$= -\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_0)+\log(\pi_1/\pi_0) \tag{4.25}$$

So if we define

$$\boldsymbol{w} = \boldsymbol{\beta}_1-\boldsymbol{\beta}_0 = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) \tag{4.26}$$

$$\boldsymbol{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_0)-(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)\frac{\log(\pi_1/\pi_0)}{(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)} \tag{4.27}$$

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

$$p(y=1|\boldsymbol{x},\boldsymbol{\theta}) = \text{sigm}(\boldsymbol{w}^T(\boldsymbol{x}-\boldsymbol{x}_0)) \tag{4.28}$$

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

If $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}$, then $\boldsymbol{w}$ 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 4.4. Furthemore, if $\pi_1 = \pi_0$, then $\boldsymbol{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_0)$, which is half way between the means. If we make $\pi_1 > \pi_0$, then $\boldsymbol{x}_0$ 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, c, just changes the decision threshold, and not the overall geometry, as we claimed above. (A similar argument applies in the multi-class case.)

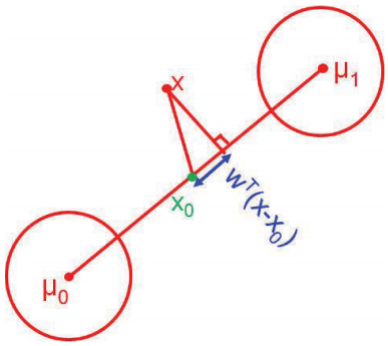


Fig. 4.4: Geometry of LDA in the 2 class case where $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}$.

The magnitude of $\boldsymbol{w}$ 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**:

$$d' \triangleq \frac{\mu_1-\mu_0}{\sigma} \tag{4.29}$$

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

The log-likelihood function is as follows:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{c=1}^{C}\sum_{i:y_i=c}\log\pi_c+\sum_{c=1}^{C}\sum_{i:y_i=c}\log\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c) \tag{4.30}$$

The MLE for each parameter is as follows:

$$\hat{\boldsymbol{\pi}}_c = \frac{N_c}{N} \tag{4.31}$$

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{N_c}\sum_{i:y_i=c}\boldsymbol{x}_i \tag{4.32}$$

$$\hat{\boldsymbol{\Sigma}}_c = \frac{1}{N_c}\sum_{i:y_i=c}(\boldsymbol{x}_i-\hat{\boldsymbol{\mu}}_c)(\boldsymbol{x}_i-\hat{\boldsymbol{\mu}}_c)^T \tag{4.33}$$

[17] http://en.wikipedia.org/wiki/Sigmoid_function

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 TODO.
- 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 TODO; 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 kindsof prior below.
- Project the data into a low dimensional subspace and fit the Gaussians there. See Section TODO for a way to find the best (most discriminative) linear projection.

We discuss some of these options below.

4.2.6 Regularized LDA *

4.2.7 Diagonal LDA

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).

4.3 Inference in jointly Gaussian distributions

Given a joint distribution, $p(\boldsymbol{x}_1, \boldsymbol{x}_2)$, it is useful to be able to compute marginals $p(\boldsymbol{x}_1)$ and conditionals $p(\boldsymbol{x}_1|\boldsymbol{x}_2)$. 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 TODO for faster methods.

4.3.1 Statement of the result

Theorem 4.2. ***(Marginals and conditionals of an MVN).*** *Suppose $X = (\boldsymbol{x}_1, \boldsymbol{x}_2)$is jointly Gaussian with parameters*

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix}, \tag{4.34}$$

Then the marginals are given by

$$\begin{aligned} p(\boldsymbol{x}_1) &= \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ p(\boldsymbol{x}_2) &= \mathcal{N}(\boldsymbol{x}_2|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \end{aligned} \tag{4.35}$$

and the posterior conditional is given by

$$\boxed{\begin{aligned} p(\boldsymbol{x}_1|\boldsymbol{x}_2) &= \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\Sigma}_{1|2}(\boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1} \end{aligned}} \tag{4.36}$$

Equation 4.36 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 TODO.

We see that both the marginal and conditional distributions are themselves Gaussian. For the marginals, we just extract the rows and columns corresponding to $\boldsymbol{x}_1$ or $\boldsymbol{x}_2$. 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 $\boldsymbol{x}_2$, and the conditional covariance is just a constant matrix that is independent of $\boldsymbol{x}_2$. 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

4.4 Linear Gaussian systems

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

$$\boxed{\begin{aligned} p(\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \\ p(\boldsymbol{y}|\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{W}\boldsymbol{x} + \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) \end{aligned}} \tag{4.37}$$

where $\boldsymbol{W}$ 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 $\boldsymbol{x} \rightarrow \boldsymbol{y}$, meaning $\boldsymbol{x}$ generates $\boldsymbol{y}$. In this section, we show how to invert the arrow, that is, how to infer $\boldsymbol{x}$ from $\boldsymbol{y}$. 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 4.3. ***(Bayes rule for linear Gaussian systems).*** *Given a linear Gaussian system, as in Equation 4.37, the posterior $p(\boldsymbol{x}|\boldsymbol{y})$ is given by the following:*

$$\boxed{\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y}) \\ \boldsymbol{\Sigma}_{x|y} &= \boldsymbol{\Sigma}_x^{-1} + \boldsymbol{W}^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{W} \\ \boldsymbol{\mu}_{x|y} &= \boldsymbol{\Sigma}_{x|y} \left[\boldsymbol{W}^T \boldsymbol{\Sigma}_y^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_y) + \boldsymbol{\Sigma}_x^{-1} \boldsymbol{\mu}_x \right] \end{aligned}} \tag{4.38}$$

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

$$\boxed{p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{W}\boldsymbol{\mu}_x + \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y + \boldsymbol{W}\boldsymbol{\Sigma}_x\boldsymbol{W}^T)} \tag{4.39}$$

For the proof, see Section 4.4.3 TODO.

4.5 Digression: The Wishart distribution *

4.6 Inferring the parameters of an MVN

4.6.1 Posterior distribution of μ

4.6.2 Posterior distribution of Σ *

4.6.3 Posterior distribution of μ and Σ *

4.6.4 Sensor fusion with unknown precisions *