3 Probability: Multivariate Models

3.1 Joint distributions for multiple random variables

In this section, we discuss various ways to measure the dependence of one or more variables on each other.

3.1.1 Covariance

The **covariance** between two rv's X and Y measures the degree to which X and Y are (linearly) related. Covariance is defined as

$$\mathrm{Cov}\left[X, Y\right] \triangleq \mathbb{E}\left[(X - \mathbb{E}\left[X\right])(Y - \mathbb{E}\left[Y\right])\right] = \mathbb{E}\left[XY\right] - \mathbb{E}\left[X\right]\mathbb{E}\left[Y\right] \tag{3.1}$$

If $\boldsymbol{x}$ is a D-dimensional random vector, its **covariance matrix** is defined to be the following symmetric, positive semi definite matrix:

$$\mathrm{Cov}\left[\boldsymbol{x}\right] \triangleq \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}\right])(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}\right])^{\mathsf{T}}\right] \triangleq \boldsymbol{\Sigma} \tag{3.2}$$

$$= \begin{pmatrix} \mathbb{V}\left[X_1\right] & \mathrm{Cov}\left[X_1, X_2\right] & \cdots & \mathrm{Cov}\left[X_1, X_D\right] \\ \mathrm{Cov}\left[X_2, X_1\right] & \mathbb{V}\left[X_2\right] & \cdots & \mathrm{Cov}\left[X_2, X_D\right] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}\left[X_D, X_1\right] & \mathrm{Cov}\left[X_D, X_2\right] & \cdots & \mathbb{V}\left[X_D\right] \end{pmatrix} \tag{3.3}$$

from which we get the important result

$$\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\right] = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}} \tag{3.4}$$

Another useful result is that the covariance of a linear transformation is given by

$$\mathrm{Cov}\left[\mathbf{A}\boldsymbol{x} + \boldsymbol{b}\right] = \mathbf{A}\mathrm{Cov}\left[\boldsymbol{x}\right]\mathbf{A}^{\mathsf{T}} \tag{3.5}$$

as shown in Exercise 3.4.

The **cross-covariance** between two random vectors is defined as

$$\mathrm{Cov}\left[\boldsymbol{x}, \boldsymbol{y}\right] = \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}\right])(\boldsymbol{y} - \mathbb{E}\left[\boldsymbol{y}\right])^{\mathsf{T}}\right] \tag{3.6}$$

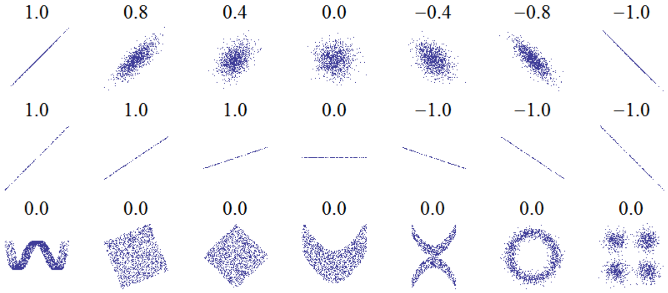


Figure 3.1: Several sets of (x, y) points, with the correlation coefficient of x and y for each set. Note that the correlation reflects the noisiness and direction of a linear relationship (top row), but not the slope of that relationship (middle), nor many aspects of nonlinear relationships (bottom). (Note: the figure in the center has a slope of 0 but in that case the correlation coefficient is undefined because the variance of Y is zero.) From https://en.wikipedia.org/wiki/Pearson_correlation_coefficient. Used with kind permission of Wikipedia author Imagecreator.

3.1.2 Correlation

Covariances can be between negative and positive infinity. Sometimes it is more convenient to work with a normalized measure, with a finite lower and upper bound. The (Pearson) **correlation coefficient** between X and Y is defined as

$$\rho \triangleq \text{corr}\left[X, Y\right] \triangleq \frac{\text{Cov}\left[X, Y\right]}{\sqrt{\mathbb{V}\left[X\right] \mathbb{V}\left[Y\right]}} \tag{3.7}$$

One can show (Exercise 3.2) that $-1 \leq \rho \leq 1$.

One can also show that $\text{corr}\left[X, Y\right] = 1$ if and only if $Y = aX + b$ (and $a > 0$) for some parameters a and b, i.e., if there is a *linear* relationship between X and Y (see Exercise 3.3). Intuitively one might expect the correlation coefficient to be related to the slope of the regression line, i.e., the coefficient a in the expression $Y = aX + b$. However, as we show in Equation (11.27), the regression coefficient is in fact given by $a = \text{Cov}\left[X, Y\right] / \mathbb{V}\left[X\right]$. In Figure 3.1, we show that the correlation coefficient can be 0 for strong, but nonlinear, relationships. (Compare to Figure 6.6.) Thus a better way to think of the correlation coefficient is as *a degree of linearity*. (See correlation2d.ipynb for a demo to illustrate this idea.)

In the case of a vector $\boldsymbol{x}$ of related random variables, the **correlation matrix** is given by

$$\text{corr}(\boldsymbol{x}) = \begin{pmatrix} 1 & \frac{\mathbb{E}[(X_1-\mu_1)(X_2-\mu_2)]}{\sigma_1\sigma_2} & \cdots & \frac{\mathbb{E}[(X_1-\mu_1)(X_D-\mu_D)]}{\sigma_1\sigma_D} \\ \frac{\mathbb{E}[(X_2-\mu_2)(X_1-\mu_1)]}{\sigma_2\sigma_1} & 1 & \cdots & \frac{\mathbb{E}[(X_2-\mu_2)(X_D-\mu_D)]}{\sigma_2\sigma_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\mathbb{E}[(X_D-\mu_D)(X_1-\mu_1)]}{\sigma_D\sigma_1} & \frac{\mathbb{E}[(X_D-\mu_D)(X_2-\mu_2)]}{\sigma_D\sigma_2} & \cdots & 1 \end{pmatrix} \tag{3.8}$$

This can be written more compactly as

$$\text{corr}(\boldsymbol{x}) = (\text{diag}(\mathbf{K}_{xx}))^{-\frac{1}{2}} \mathbf{K}_{xx} (\text{diag}(\mathbf{K}_{xx}))^{-\frac{1}{2}} \tag{3.9}$$

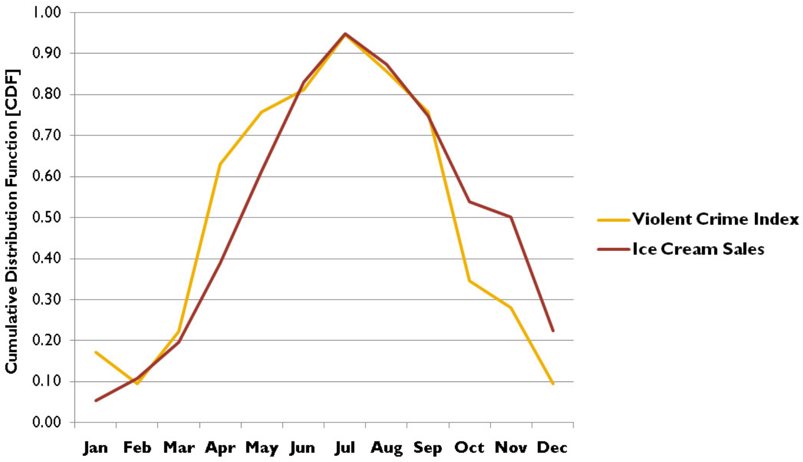


Figure 3.2: Examples of spurious correlation between causally unrelated time series. Consumption of ice cream (red) and violent crime rate (yellow). over time. From http: // icbseverywhere. com/ blog/ 2014/ 10/ the-logic-of-causal-conclusions/ . Used with kind permission of Barbara Drescher.

where $\mathbf{K}_{xx}$ is the **auto-covariance matrix**

$$\mathbf{K}_{xx} = \mathbf{\Sigma} = \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^{\mathsf{T}}\right] = \mathbf{R}_{xx} - \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}} \tag{3.10}$$

and $\mathbf{R}_{xx} = \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\right]$ is the **autocorrelation matrix**.

3.1.3 Uncorrelated does not imply independent

If X and Y are independent, meaning $p(X, Y) = p(X)p(Y)$, then $\text{Cov}[X, Y] = 0$, and hence $\text{corr}[X, Y] = 0$. So independent implies uncorrelated. However, the converse is not true: *uncorrelated does not imply independent*. For example, let $X \sim U(-1, 1)$ and $Y = X^2$. Clearly Y is dependent on X (in fact, Y is uniquely determined by X), yet one can show (Exercise 3.1) that $\text{corr}[X, Y] = 0$. Some striking examples of this fact are shown in Figure 3.1. This shows several data sets where there is clear dependence between X and Y, and yet the correlation coefficient is 0. A more general measure of dependence between random variables is mutual information, discussed in Section 6.3. This is zero only if the variables truly are independent.

3.1.4 Correlation does not imply causation

It is well known that "**correlation does not imply causation**". For example, consider Figure 3.2. In red, we plot $x_{1:T}$, where x_t is the amount of ice cream sold in month t. In yellow, we plot $y_{1:T}$, where y_t is the violent crime rate in month t. (Quantities have been rescaled to make the plots overlap.) We see a strong correlation between these signals. Indeed, it is sometimes claimed that "eating ice cream causes murder" [Pet13]. Of course, this is just a **spurious correlation**, due to a **hidden common cause**, namely the weather. Hot weather increases ice cream sales, for obvious

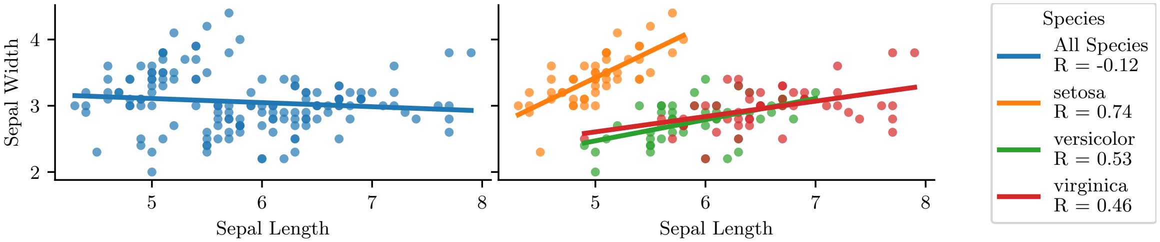


Figure 3.3: Illustration of Simpson's paradox on the Iris dataset. (Left) Overall, y (sepal width) decreases with x (sepal length). (Right) Within each group, y increases with x. Generated by simpsons_paradox.ipynb.

reasons. Hot weather also increases violent crime; the reason for this is hotly (ahem) debated; some claim it is due to an increase in anger [And01], but other claim it is merely due to more people being outside [Ash18], where most murders occur.

Another famous example concerns the positive correlation between birth rates and the presence of storks (a kind of bird). This has given rise to the urban legend that storks deliver babies [Mat00]. Of course, the true reason for the correlation is more likely due to hidden factors, such as increased living standards and hence more food. Many more amusing examples of such spurious correlations can be found in [Vig15].

These examples serve as a "warning sign", that we should not treat the ability for x to predict y as an indicator that x causes y.

3.1.5 Simpson's paradox

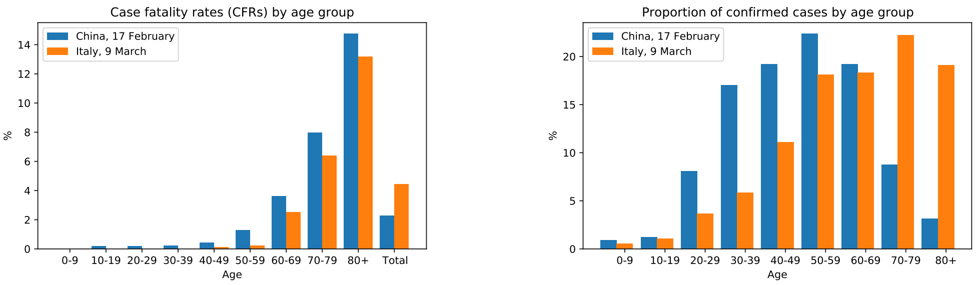
Simpson's paradox says that a statistical trend or relationship that appears in several different groups of data can disappear or reverse sign when these groups are combined. This results in counterintuitive behavior if we misinterpret claims of statistical dependence in a causal way.

A visualization of the paradox is given in Figure 3.3. Overall, we see that y decreases with x, but within each subpopulation, y increases with x.

For a recent real-world example of Simpson's paradox in the context of COVID-19, consider Figure 3.4(a). This shows that the case fatality rate (CFR) of COVID-19 in Italy is less than in China in each age group, but is higher overall. The reason for this is that there are more older people in Italy, as shown in Figure 3.4(b). In other words, Figure 3.4(a) shows $p(F = 1|A, C)$, where A is age, C is country, and $F = 1$ is the event that someone dies from COVID-19, and Figure 3.4(b) shows $p(A|C)$, which is the probability someone is in age bucket A for country C. Combining these, we find $p(F = 1|C = \text{Italy}) > p(F = 1|C = \text{China})$. See [KGS20] for more details.

3.2 The multivariate Gaussian (normal) distribution

The most widely used joint probability distribution for continuous random variables is the **multivariate Gaussian** or **multivariate normal** (**MVN**). This is mostly because it is mathematically convenient, but also because the Gaussian assumption is fairly reasonable in many cases (see the discussion in Section 2.6.4).



(a) *(b)*

Figure 3.4: Illustration of Simpson's paradox using COVID-19, (a) Case fatality rates (CFRs) in Italy and China by age group, and in aggregated form ("Total", last pair of bars), up to the time of reporting (see legend). (b) Proportion of all confirmed cases included in (a) within each age group by country. From Figure 1 of [KGS20]. Used with kind permission of Julius von Kügelgen.

3.2.1 Definition

The MVN density is defined by the following:

$$\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \triangleq \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right] \tag{3.11}$$

where $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{y}] \in \mathbb{R}^D$ is the mean vector, and $\boldsymbol{\Sigma} = \mathrm{Cov}[\boldsymbol{y}]$ is the $D \times D$ **covariance matrix**, defined as follows:

$$\mathrm{Cov}[\boldsymbol{y}] \triangleq \mathbb{E}\left[(\boldsymbol{y}-\mathbb{E}[\boldsymbol{y}])(\boldsymbol{y}-\mathbb{E}[\boldsymbol{y}])^{\mathsf{T}}\right] \tag{3.12}$$

$$= \begin{pmatrix} \mathbb{V}[Y_1] & \mathrm{Cov}[Y_1,Y_2] & \cdots & \mathrm{Cov}[Y_1,Y_D] \\ \mathrm{Cov}[Y_2,Y_1] & \mathbb{V}[Y_2] & \cdots & \mathrm{Cov}[Y_2,Y_D] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[Y_D,Y_1] & \mathrm{Cov}[Y_D,Y_2] & \cdots & \mathbb{V}[Y_D] \end{pmatrix} \tag{3.13}$$

where

$$\mathrm{Cov}[Y_i,Y_j] \triangleq \mathbb{E}[(Y_i-\mathbb{E}[Y_i])(Y_j-\mathbb{E}[Y_j])] = \mathbb{E}[Y_iY_j] - \mathbb{E}[Y_i]\,\mathbb{E}[Y_j] \tag{3.14}$$

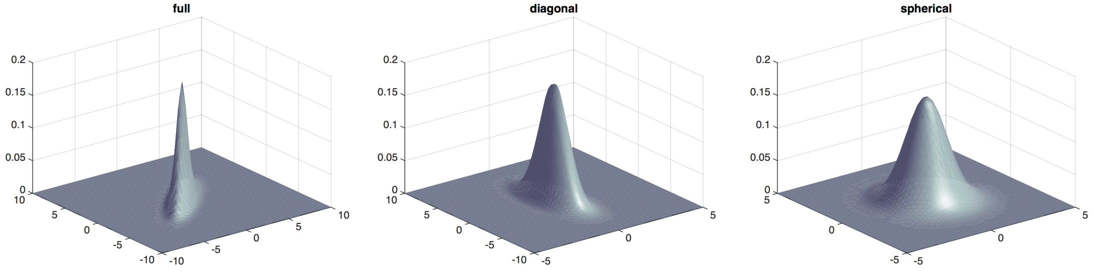
and $\mathbb{V}[Y_i] = \mathrm{Cov}[Y_i,Y_i]$. From Equation (3.12), we get the important result

$$\mathbb{E}\left[\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\right] = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}} \tag{3.15}$$

The normalization constant in Equation (3.11) $Z = (2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}$ just ensures that the pdf integrates to 1 (see Exercise 3.6).

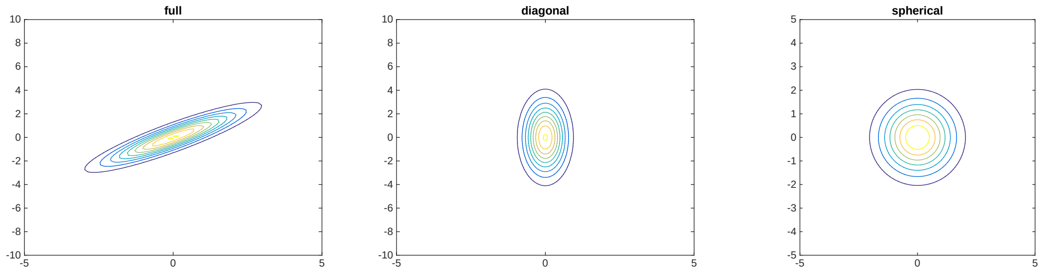
In 2d, the MVN is known as the **bivariate Gaussian** distribution. Its pdf can be represented as $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, where $\boldsymbol{y} \in \mathbb{R}^2$, $\boldsymbol{\mu} \in \mathbb{R}^2$ and

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12}^2 \\ \sigma_{21}^2 & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{3.16}$$



(a) *(b)* *(c)*

Figure 3.5: Visualization of a 2d Gaussian density as a surface plot. (a) Distribution using a full covariance matrix can be oriented at any angle. (b) Distribution using a diagonal covariance matrix must be parallel to the axis. (c) Distribution using a spherical covariance matrix must have a symmetric shape. Generated by gauss_plot_2d.ipynb.



(a) *(b)* *(c)*

Figure 3.6: Visualization of a 2d Gaussian density in terms of level sets of constant probability density. (a) A full covariance matrix has elliptical contours. (b) A diagonal covariance matrix is an **axis aligned** *ellipse. (c) A spherical covariance matrix has a circular shape. Generated by gauss_plot_2d.ipynb.*

where ρ is the **correlation coefficient**, defined by

$$\text{corr}\left[Y_1, Y_2\right] \triangleq \frac{\text{Cov}\left[Y_1, Y_2\right]}{\sqrt{\mathbb{V}\left[Y_1\right]\mathbb{V}\left[Y_2\right]}} = \frac{\sigma_{12}^2}{\sigma_1\sigma_2} \tag{3.17}$$

One can show (Exercise 3.2) that $-1 \leq \text{corr}\left[Y_1, Y_2\right] \leq 1$. Expanding out the pdf in the 2d case gives the following rather intimidating-looking result:

$$\begin{aligned} p(y_1, y_2) = &\frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)} \times \right. \\ &\left.\left[\frac{(y_1-\mu_1)^2}{\sigma_1^2} + \frac{(y_2-\mu_2)^2}{\sigma_2^2} - 2\rho\frac{(y_1-\mu_1)}{\sigma_1}\frac{(y_2-\mu_2)}{\sigma_2}\right]\right) \end{aligned} \tag{3.18}$$

Figure 3.5 and Figure 3.6 plot some MVN densities in 2d for three different kinds of covariance matrices. A **full covariance matrix** has $D(D+1)/2$ parameters, where we divide by 2 since $\mathbf{\Sigma}$ is symmetric. (The reason for the elliptical shape is explained in Section 7.4.4, where we discuss the geometry of quadratic forms.) A **diagonal covariance matrix** has D parameters, and has 0s in the off-diagonal terms. A **spherical covariance matrix**, also called **isotropic covariance matrix**, has the form $\mathbf{\Sigma} = \sigma^2 \mathbf{I}_D$, so it only has one free parameter, namely σ^2.

3.2.2 Mahalanobis distance

In this section, we attempt to gain some insights into the geometric shape of the Gaussian pdf in multiple dimensions. To do this, we will consider the shape of the **level sets** of constant (log) probability.

The log probability at a specific point $\boldsymbol{y}$ is given by

$$\log p(\boldsymbol{y}|\boldsymbol{\mu}, \mathbf{\Sigma}) = -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu})^{\mathsf{T}} \mathbf{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) + \text{const} \tag{3.19}$$

The dependence on $\boldsymbol{y}$ can be expressed in terms of the **Mahalanobis distance** Δ between $\boldsymbol{y}$ and $\boldsymbol{\mu}$, whose square is defined as follows:

$$\Delta^2 \triangleq (\boldsymbol{y} - \boldsymbol{\mu})^{\mathsf{T}} \mathbf{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \tag{3.20}$$

Thus contours of constant (log) probability are equivalent to contours of constant Mahalanobis distance.

To gain insight into the contours of constant Mahalanobis distance, we exploit the fact that $\mathbf{\Sigma}$, and hence $\mathbf{\Lambda} = \mathbf{\Sigma}^{-1}$, are both positive definite matrices (by assumption). Consider the following eigendecomposition (Section 7.4) of $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \sum_{d=1}^{D} \lambda_d \boldsymbol{u}_d \boldsymbol{u}_d^{\mathsf{T}} \tag{3.21}$$

We can similarly write

$$\mathbf{\Sigma}^{-1} = \sum_{d=1}^{D} \frac{1}{\lambda_d} \boldsymbol{u}_d \boldsymbol{u}_d^{\mathsf{T}} \tag{3.22}$$

Let us define $z_d \triangleq \boldsymbol{u}_d^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{\mu})$, so $\boldsymbol{z} = \mathbf{U}(\boldsymbol{y} - \boldsymbol{\mu})$. Then we can rewrite the Mahalanobis distance as follows:

$$(\boldsymbol{y} - \boldsymbol{\mu})^{\mathsf{T}} \mathbf{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) = (\boldsymbol{y} - \boldsymbol{\mu})^{\mathsf{T}} \left(\sum_{d=1}^{D} \frac{1}{\lambda_d} \boldsymbol{u}_d \boldsymbol{u}_d^{\mathsf{T}} \right) (\boldsymbol{y} - \boldsymbol{\mu}) \tag{3.23}$$

$$= \sum_{d=1}^{D} \frac{1}{\lambda_d} (\boldsymbol{y} - \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{u}_d \boldsymbol{u}_d^{\mathsf{T}} (\boldsymbol{y} - \boldsymbol{\mu}) = \sum_{d=1}^{D} \frac{z_d^2}{\lambda_d} \tag{3.24}$$

As we discuss in Section 7.4.4, this means we can interpret the Mahalanobis distance as Euclidean distance in a new coordinate frame $\boldsymbol{z}$ in which we rotate $\boldsymbol{y}$ by $\mathbf{U}$ and scale by $\mathbf{\Lambda}$.

For example, in 2d, let us consider the set of points (z_1, z_2) that satisfy this equation:

$$\frac{z_1^2}{\lambda_1} + \frac{z_2^2}{\lambda_2} = r \tag{3.25}$$

Since these points have the same Mahalanobis distance, they correspond to points of equal probability. Hence we see that the contours of equal probability density of a 2d Gaussian lie along ellipses. This is illustrated in Figure 7.6. The eigenvectors determine the orientation of the ellipse, and the eigenvalues determine how elongated it is.

3.2.3 Marginals and conditionals of an MVN *

Suppose $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2)$ is jointly Gaussian with parameters

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix} \tag{3.26}$$

where $\boldsymbol{\Lambda}$ is the **precision matrix**. Then the marginals are given by

$$\boxed{\begin{aligned} p(\boldsymbol{y}_1) &= \mathcal{N}(\boldsymbol{y}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ p(\boldsymbol{y}_2) &= \mathcal{N}(\boldsymbol{y}_2 | \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \end{aligned}} \tag{3.27}$$

and the posterior conditional is given by

$$\boxed{\begin{aligned} p(\boldsymbol{y}_1 | \boldsymbol{y}_2) &= \mathcal{N}(\boldsymbol{y}_1 | \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} (\boldsymbol{y}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1} \boldsymbol{\Lambda}_{12} (\boldsymbol{y}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\Sigma}_{1|2} \left(\boldsymbol{\Lambda}_{11} \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12} (\boldsymbol{y}_2 - \boldsymbol{\mu}_2) \right) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1} \end{aligned}} \tag{3.28}$$

These equations are of such crucial importance in this book that we have put a box around them, so you can easily find them later. For the derivation of these results (which relies on computing the Schur complement $\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{22} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$), see Section 7.3.5.

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

3.2.4 Example: conditioning a 2d Gaussian

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

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{3.29}$$

The marginal $p(y_1)$ is a 1D Gaussian, obtained by projecting the joint distribution onto the y_1 line:

$$p(y_1) = \mathcal{N}(y_1|\mu_1, \sigma_1^2) \tag{3.30}$$

Suppose we observe $Y_2 = y_2$; the conditional $p(y_1|y_2)$ is obtained by "slicing" the joint distribution through the $Y_2 = y_2$ line:

$$p(y_1|y_2) = \mathcal{N}\left(y_1|\mu_1 + \frac{\rho\sigma_1\sigma_2}{\sigma_2^2}(y_2 - \mu_2),\ \sigma_1^2 - \frac{(\rho\sigma_1\sigma_2)^2}{\sigma_2^2}\right) \tag{3.31}$$

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

$$p(y_1|y_2) = \mathcal{N}\left(y_1|\mu_1 + \rho(y_2 - \mu_2),\ \sigma^2(1 - \rho^2)\right) \tag{3.32}$$

For example, suppose $\rho = 0.8$, $\sigma_1 = \sigma_2 = 1$, $\mu_1 = \mu_2 = 0$, and $y_2 = 1$. We see that $\mathbb{E}[y_1|y_2 = 1] = 0.8$, which makes sense, since $\rho = 0.8$ means that we believe that if y_2 increases by 1 (beyond its mean), then y_1 increases by 0.8. We also see $\mathbb{V}[y_1|y_2 = 1] = 1 - 0.8^2 = 0.36$. This also makes sense: our uncertainty about y_1 has gone down, since we have learned something about y_1 (indirectly) by observing y_2. If $\rho = 0$, we get $p(y_1|y_2) = \mathcal{N}\left(y_1|\mu_1,\ \sigma_1^2\right)$, since y_2 conveys no information about y_1 if they are uncorrelated (and hence independent).

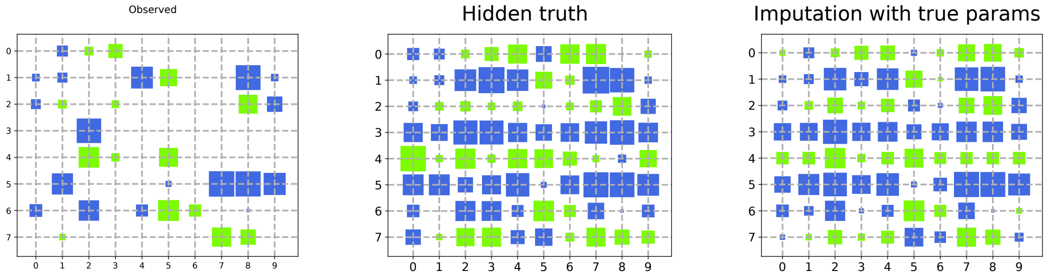
3.2.5 Example: Imputing missing values *

As an example application of the above results, suppose we observe some parts (dimensions) of $\boldsymbol{y}$, with the remaining parts being missing or unobserved. We can exploit the correlation amongst the dimensions (encoded by the covariance matrix) to infer the missing entries; this is called **missing value imputation**.

Figure 3.7 shows a simple example. We sampled N vectors from a $D = 10$-dimensional Gaussian, and then deliberately "hid" 50% of the data in each sample (row). We then inferred the missing entries given the observed entries and the true model parameters.[1] More precisely, for each row n of the data matrix, we compute $p(\boldsymbol{y}_{n,\boldsymbol{h}}|\boldsymbol{y}_{n,\boldsymbol{v}}, \boldsymbol{\theta})$, where $\boldsymbol{v}$ are the indices of the visible entries in that row, $\boldsymbol{h}$ are the remaining indices of the hidden entries, and $\boldsymbol{\theta} = (\boldsymbol{\mu}, \mathbf{\Sigma})$. From this, we compute the marginal distribution of each missing variable $i \in \boldsymbol{h}$, $p(y_{n,i}|\boldsymbol{y}_{n,\boldsymbol{v}}, \boldsymbol{\theta})$. From the marginal, we compute the posterior mean, $\bar{y}_{n,i} = \mathbb{E}[y_{n,i}|\boldsymbol{y}_{n,\boldsymbol{v}}, \boldsymbol{\theta}]$.

The posterior mean represents our "best guess" about the true value of that entry, in the sense that it minimizes our expected squared error, as explained in Chapter 5. We can use $\mathbb{V}[y_{n,i}|\boldsymbol{y}_{n,\boldsymbol{v}}, \boldsymbol{\theta}]$ as a measure of confidence in this guess, although this is not shown. Alternatively, we could draw multiple posterior samples from $p(\boldsymbol{y}_{n,\boldsymbol{h}}|\boldsymbol{y}_{n,\boldsymbol{v}}, \boldsymbol{\theta})$; this is called **multiple imputation**, and provides a more robust estimate to downstream algorithms that consume the "filled in" data.

[1] In practice, we would need to estimate the parameters from the partially observed data. Unfortunately the MLE results in Section 4.2.6 no longer apply, but we can use the EM algorithm to derive an approximate MLE in the presence of missing data. See the sequel to this book for details.



(a) *(b)* *(c)*

Figure 3.7: Illustration of data imputation using an MVN. (a) Visualization of the data matrix. Blank entries are missing (not observed). Blue are positive, green are negative. Area of the square is proportional to the value. (This is known as a **Hinton diagram***, named after Geoff Hinton, a famous ML researcher.) (b) True data matrix (hidden). (c) Mean of the posterior predictive distribution, based on partially observed data in that row, using the true model parameters. Generated by gauss_imputation_known_params_demo.ipynb.*

3.3 Linear Gaussian systems *

In Section 3.2.3, we conditioned on noise-free observations to infer the posterior over the hidden parts of a Gaussian random vector. In this section, we extend this approach to handle noisy observations.

Let $\boldsymbol{z} \in \mathbb{R}^L$ be an unknown vector of values, and $\boldsymbol{y} \in \mathbb{R}^D$ be some noisy measurement of $\boldsymbol{z}$. We assume these variables are related by the following joint distribution:

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \tag{3.33}$$

$$p(\boldsymbol{y}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{y}|\mathbf{W}\boldsymbol{z} + \boldsymbol{b}, \boldsymbol{\Sigma}_y) \tag{3.34}$$

where $\mathbf{W}$ is a matrix of size $D \times L$. This is an example of a **linear Gaussian system**.

The corresponding joint distribution, $p(\boldsymbol{z}, \boldsymbol{y}) = p(\boldsymbol{z})p(\boldsymbol{y}|\boldsymbol{z})$, is a $L + D$ dimensional Gaussian, with mean and covariance given by

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_z \\ \mathbf{W}\boldsymbol{\mu}_z + \boldsymbol{b} \end{pmatrix} \tag{3.35}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_z & \boldsymbol{\Sigma}_z\mathbf{W}^\mathsf{T} \\ \mathbf{W}\boldsymbol{\Sigma}_z & \boldsymbol{\Sigma}_y + \mathbf{W}\boldsymbol{\Sigma}_z\mathbf{W}^\mathsf{T} \end{pmatrix} \tag{3.36}$$

By applying the Gaussian conditioning formula in Equation (3.28) to the joint $p(\boldsymbol{y}, \boldsymbol{z})$ we can compute the posterior $p(\boldsymbol{z}|\boldsymbol{y})$, as we explain below. This can be interpreted as inverting the $\boldsymbol{z} \rightarrow \boldsymbol{y}$ arrow in the generative model from latents to observations.

3.3.1 Bayes rule for Gaussians

The posterior over the latent is given by

$$\boxed{\begin{aligned} p(\boldsymbol{z}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_{z|y}, \boldsymbol{\Sigma}_{z|y}) \\ \boldsymbol{\Sigma}_{z|y}^{-1} &= \boldsymbol{\Sigma}_z^{-1} + \mathbf{W}^{\mathsf{T}}\boldsymbol{\Sigma}_y^{-1}\mathbf{W} \\ \boldsymbol{\mu}_{z|y} &= \boldsymbol{\Sigma}_{z|y}[\mathbf{W}^{\mathsf{T}}\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{y} - \boldsymbol{b}) + \boldsymbol{\Sigma}_z^{-1}\boldsymbol{\mu}_z] \end{aligned}} \tag{3.37}$$

This is known as **Bayes rule for Gaussians**. Furthermore, the normalization constant of the posterior is given by

$$p(\boldsymbol{y}) = \int \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)\mathcal{N}(\boldsymbol{y}|\mathbf{W}\boldsymbol{z} + \boldsymbol{b}, \boldsymbol{\Sigma}_y)d\boldsymbol{z} = \mathcal{N}(\boldsymbol{y}|\mathbf{W}\boldsymbol{\mu}_z + \boldsymbol{b}, \boldsymbol{\Sigma}_y + \mathbf{W}\boldsymbol{\Sigma}_z\mathbf{W}^{\mathsf{T}}) \tag{3.38}$$

We see that the Gaussian prior $p(\boldsymbol{z})$, combined with the Gaussian likelihood $p(\boldsymbol{y}|\boldsymbol{z})$, results in a Gaussian posterior $p(\boldsymbol{z}|\boldsymbol{y})$. Thus Gaussians are closed under Bayesian conditioning. To describe this more generally, we say that the Gaussian prior is a **conjugate prior** for the Gaussian likelihood, since the posterior distribution has the same type as the prior. We discuss the notion of conjugate priors in more detail in Section 4.6.1.

In the sections below, we give various applications of this result. But first, we give the derivation.

3.3.2 Derivation *

We now derive Equation 3.37. The basic idea is to derive the joint distribution, $p(\boldsymbol{z}, \boldsymbol{y}) = p(\boldsymbol{z})p(\boldsymbol{y}|\boldsymbol{z})$, and then to use the results from Section 3.2.3 for computing $p(\boldsymbol{z}|\boldsymbol{y})$.

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

$$\log p(\boldsymbol{z}, \boldsymbol{y}) = -\frac{1}{2}(\boldsymbol{z} - \boldsymbol{\mu}_z)^T\boldsymbol{\Sigma}_z^{-1}(\boldsymbol{z} - \boldsymbol{\mu}_z) - \frac{1}{2}(\boldsymbol{y} - \mathbf{W}\boldsymbol{z} - \boldsymbol{b})^T\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{y} - \mathbf{W}\boldsymbol{z} - \boldsymbol{b}) \tag{3.39}$$

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

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

$$\begin{aligned} Q &= -\frac{1}{2}\boldsymbol{z}^T\boldsymbol{\Sigma}_z^{-1}\boldsymbol{z} - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{\Sigma}_y^{-1}\boldsymbol{y} - \frac{1}{2}(\mathbf{W}\boldsymbol{z})^T\boldsymbol{\Sigma}_y^{-1}(\mathbf{W}\boldsymbol{z}) + \boldsymbol{y}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{W}\boldsymbol{z} & (3.40) \\ &= -\frac{1}{2}\begin{pmatrix}\boldsymbol{z}\\ \boldsymbol{y}\end{pmatrix}^T\begin{pmatrix}\boldsymbol{\Sigma}_z^{-1} + \mathbf{W}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{W} & -\mathbf{W}^T\boldsymbol{\Sigma}_y^{-1} \\ -\boldsymbol{\Sigma}_y^{-1}\mathbf{W} & \boldsymbol{\Sigma}_y^{-1}\end{pmatrix}\begin{pmatrix}\boldsymbol{z}\\ \boldsymbol{y}\end{pmatrix} & (3.41) \\ &= -\frac{1}{2}\begin{pmatrix}\boldsymbol{z}\\ \boldsymbol{y}\end{pmatrix}^T\boldsymbol{\Sigma}^{-1}\begin{pmatrix}\boldsymbol{z}\\ \boldsymbol{y}\end{pmatrix} & (3.42) \end{aligned}$$

where the precision matrix of the joint is defined as

$$\boldsymbol{\Sigma}^{-1} = \begin{pmatrix}\boldsymbol{\Sigma}_z^{-1} + \mathbf{W}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{W} & -\mathbf{W}^T\boldsymbol{\Sigma}_y^{-1} \\ -\boldsymbol{\Sigma}_y^{-1}\mathbf{W} & \boldsymbol{\Sigma}_y^{-1}\end{pmatrix} \triangleq \boldsymbol{\Lambda} = \begin{pmatrix}\boldsymbol{\Lambda}_{zz} & \boldsymbol{\Lambda}_{zy} \\ \boldsymbol{\Lambda}_{yz} & \boldsymbol{\Lambda}_{yy}\end{pmatrix} \tag{3.43}$$

From Equation 3.28, and using the fact that $\boldsymbol{\mu}_y = \mathbf{W}\boldsymbol{\mu}_z + \boldsymbol{b}$, we have

$$p(\boldsymbol{z}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_{z|y}, \boldsymbol{\Sigma}_{z|y}) \tag{3.44}$$

$$\boldsymbol{\Sigma}_{z|y} = \boldsymbol{\Lambda}_{zz}^{-1} = (\boldsymbol{\Sigma}_z^{-1} + \mathbf{W}^T \boldsymbol{\Sigma}_y^{-1} \mathbf{W})^{-1} \tag{3.45}$$

$$\boldsymbol{\mu}_{z|y} = \boldsymbol{\Sigma}_{z|y} \left(\boldsymbol{\Lambda}_{zz} \boldsymbol{\mu}_z - \boldsymbol{\Lambda}_{zy} (\boldsymbol{y} - \boldsymbol{\mu}_y) \right) \tag{3.46}$$

$$= \boldsymbol{\Sigma}_{z|y} \left(\boldsymbol{\Sigma}_z^{-1} \boldsymbol{\mu}_z + \mathbf{W}^\mathsf{T} \boldsymbol{\Sigma}_y^{-1} \mathbf{W} \boldsymbol{\mu}_z + \mathbf{W}^\mathsf{T} \boldsymbol{\Sigma}_y^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_y) \right) \tag{3.47}$$

$$= \boldsymbol{\Sigma}_{z|y} \left(\boldsymbol{\Sigma}_z^{-1} \boldsymbol{\mu}_z + \mathbf{W}^\mathsf{T} \boldsymbol{\Sigma}_y^{-1} (\mathbf{W} \boldsymbol{\mu}_z + \boldsymbol{y} - \boldsymbol{\mu}_y) \right) \tag{3.48}$$

$$= \boldsymbol{\Sigma}_{z|y} \left(\boldsymbol{\Sigma}_z^{-1} \boldsymbol{\mu}_z + \mathbf{W}^T \boldsymbol{\Sigma}_y^{-1} (\boldsymbol{y} - \boldsymbol{b}) \right) \tag{3.49}$$

3.3.3 Example: Inferring an unknown scalar

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

$$p(y_i|z) = \mathcal{N}(y_i|z, \lambda_y^{-1}) \tag{3.50}$$

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

$$p(z) = \mathcal{N}(z|\mu_0, \lambda_0^{-1}) \tag{3.51}$$

We want to compute $p(z|y_1, \ldots, y_N, \sigma^2)$. We can convert this to a form that lets us apply Bayes rule for Gaussians by defining $\boldsymbol{y} = (y_1, \ldots, y_N)$, $\mathbf{W} = \mathbf{1}_N$ (an $N \times 1$ column vector of 1's), and $\boldsymbol{\Sigma}_y^{-1} = \text{diag}(\lambda_y \mathbf{I})$. Then we get

$$p(z|\boldsymbol{y}) = \mathcal{N}(z|\mu_N, \lambda_N^{-1}) \tag{3.52}$$

$$\lambda_N = \lambda_0 + N\lambda_y \tag{3.53}$$

$$\mu_N = \frac{N\lambda_y \overline{y} + \lambda_0 \mu_0}{\lambda_N} = \frac{N\lambda_y}{N\lambda_y + \lambda_0} \overline{y} + \frac{\lambda_0}{N\lambda_y + \lambda_0} \mu_0 \tag{3.54}$$

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 3.8.

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, as follows:

$$p(z|\mathcal{D}, \sigma^2) = \mathcal{N}(z|\mu_N, \tau_N^2) \tag{3.55}$$

$$\tau_N^2 = \frac{1}{\frac{N}{\sigma^2} + \frac{1}{\tau_0^2}} = \frac{\sigma^2 \tau_0^2}{N\tau_0^2 + \sigma^2} \tag{3.56}$$

$$\mu_N = \tau_N^2 \left(\frac{\mu_0}{\tau_0^2} + \frac{N\overline{y}}{\sigma^2} \right) = \frac{\sigma^2}{N\tau_0^2 + \sigma^2} \mu_0 + \frac{N\tau_0^2}{N\tau_0^2 + \sigma^2} \overline{y} \tag{3.57}$$

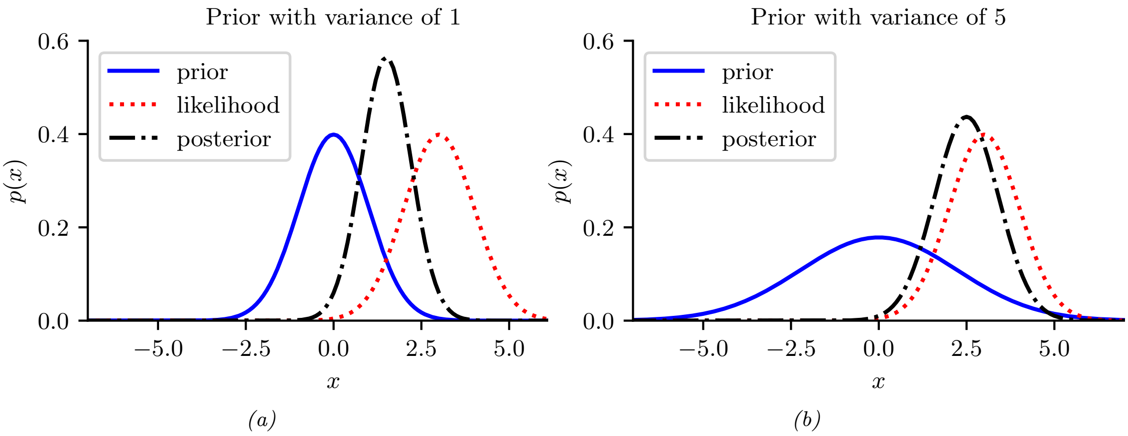


Figure 3.8: Inference about z given a noisy observation $y = 3$. (a) Strong prior $\mathcal{N}(0, 1)$. The posterior mean is "shrunk" towards the prior mean, which is 0. (b) Weak prior $\mathcal{N}(0, 5)$. The posterior mean is similar to the MLE. Generated by gauss_infer_1d.ipynb.

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):

$$p(z|y) = \mathcal{N}(z|\mu_1, \Sigma_1) \tag{3.58}$$

$$\Sigma_1 = \left(\frac{1}{\Sigma_0} + \frac{1}{\Sigma_y}\right)^{-1} = \frac{\Sigma_y \Sigma_0}{\Sigma_0 + \Sigma_y} \tag{3.59}$$

$$\mu_1 = \Sigma_1 \left(\frac{\mu_0}{\Sigma_0} + \frac{y}{\Sigma_y}\right) \tag{3.60}$$

We can rewrite the posterior mean in 3 different ways:

$$\mu_1 = \frac{\Sigma_y}{\Sigma_y + \Sigma_0}\mu_0 + \frac{\Sigma_0}{\Sigma_y + \Sigma_0}y \tag{3.61}$$

$$= \mu_0 + (y - \mu_0)\frac{\Sigma_0}{\Sigma_y + \Sigma_0} \tag{3.62}$$

$$= y - (y - \mu_0)\frac{\Sigma_y}{\Sigma_y + \Sigma_0} \tag{3.63}$$

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 3.8(a)), whereas if Σ_0 is large relative to Σ_y, corresponding to a weak prior, the amount of shrinkage is small (see Figure 3.8(b)).

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

$$\text{SNR} \triangleq \frac{\mathbb{E}\left[Z^2\right]}{\mathbb{E}\left[\epsilon^2\right]} = \frac{\Sigma_0 + \mu_0^2}{\Sigma_y} \tag{3.64}$$

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

3.3.4 Example: inferring an unknown vector

Suppose we have an unknown quantity of interest, $\boldsymbol{z} \in \mathbb{R}^D$, which we endow with a Gaussian prior, $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$. If we "know nothing" about $\boldsymbol{z}$ a priori, we can set $\boldsymbol{\Sigma}_z = \infty\mathbf{I}$, which means we are completely uncertain about what the value of $\boldsymbol{z}$ should be. (In practice, we can use a large but finite value for the covariance.) By symmetry, it seems reasonable to set $\boldsymbol{\mu}_z = \mathbf{0}$.

Now suppose we make N noisy but independent measurements of $\boldsymbol{z}$, $\boldsymbol{y}_n \sim \mathcal{N}(\boldsymbol{z}, \boldsymbol{\Sigma}_y)$, each of size D. We can represent the likelihood as follows:

$$p(\mathcal{D}|\boldsymbol{z}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{z}, \boldsymbol{\Sigma}_y) = \mathcal{N}(\overline{\boldsymbol{y}}|\boldsymbol{z}, \frac{1}{N}\boldsymbol{\Sigma}_y) \tag{3.65}$$

Note that we can replace the N observations with their average, $\overline{\boldsymbol{y}}$, provided we scale down the covariance by $1/N$ to compensate. Setting $\mathbf{W} = \mathbf{I}$, $\boldsymbol{b} = \mathbf{0}$, we can then use Bayes rule for Gaussian to compute the posterior over $\boldsymbol{z}$:

$$p(\boldsymbol{z}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N) = \mathcal{N}(\boldsymbol{z}|\,\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}) \tag{3.66}$$

$$\widehat{\boldsymbol{\Sigma}}^{-1} = \boldsymbol{\Sigma}_z^{-1} + N_{\mathcal{D}}\boldsymbol{\Sigma}_y^{-1} \tag{3.67}$$

$$\widehat{\boldsymbol{\mu}} = \widehat{\boldsymbol{\Sigma}}\,(\boldsymbol{\Sigma}_y^{-1}(N_{\mathcal{D}}\overline{\boldsymbol{y}}) + \boldsymbol{\Sigma}_z^{-1}\boldsymbol{\mu}_z) \tag{3.68}$$

where $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ are the parameters of the posterior.

Figure 3.9 gives a 2d example. We can think of $\boldsymbol{z}$ as representing the true, but unknown, location of an object in 2d space, such as a missile or airplane, and the $\boldsymbol{y}_n$ as being noisy observations, such as radar "blips". As we receive more blips, we are better able to localize the source. (In the sequel to this book, [Mur23], we discuss the **Kalman filter** algorithm, which extends this idea to a temporal sequence of observations.)

The posterior uncertainty about each component of $\boldsymbol{z}$ location vector depends on how reliable the sensor is in each of these dimensions. In the above example, the measurement noise in dimension 1 is higher than in dimension 2, so we have more posterior uncertainty about z_1 (horizontal axis) than about z_2 (vertical axis).

3.3.5 Example: sensor fusion

In this section, we extend Section 3.3.4, to the case where we have multiple measurements, coming from different sensors, each with different reliabilities. That is, the model has the form

$$p(\boldsymbol{z}, \boldsymbol{y}) = p(\boldsymbol{z}) \prod_{m=1}^{M} \prod_{n=1}^{N_m} \mathcal{N}(\boldsymbol{y}_{n,m}|\boldsymbol{z}, \boldsymbol{\Sigma}_m) \tag{3.69}$$

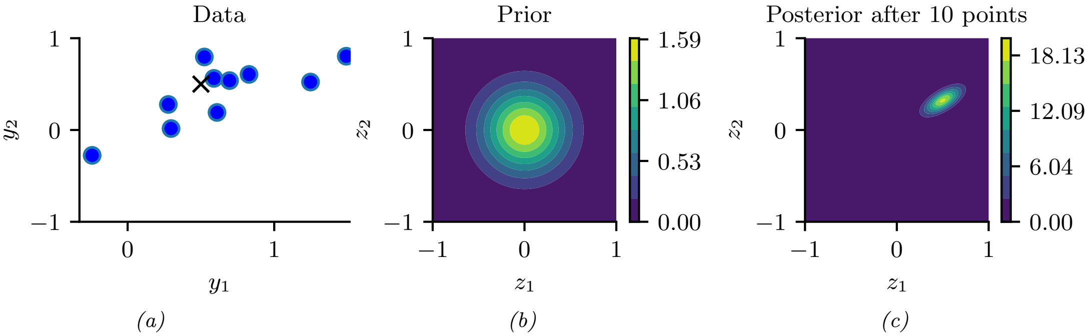


Figure 3.9: Illustration of Bayesian inference for a 2d Gaussian random vector $\boldsymbol{z}$. (a) The data is generated from $\boldsymbol{y}_n \sim \mathcal{N}(\boldsymbol{z}, \boldsymbol{\Sigma}_y)$, where $\boldsymbol{z} = [0.5, 0.5]^\mathsf{T}$ and $\boldsymbol{\Sigma}_y = 0.1[2, 1; 1, 1])$. We assume the sensor noise covariance $\boldsymbol{\Sigma}_y$ is known but $\boldsymbol{z}$ is unknown. The black cross represents $\boldsymbol{z}$. (b) The prior is $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, 0.1\mathbf{I}_2)$. (c) We show the posterior after 10 data points have been observed. Generated by gauss_infer_2d.ipynb.

where M is the number of sensors (measurement devices), and N_m is the number of observations from sensor m, and $\boldsymbol{y} = \boldsymbol{y}_{1:N,1:M} \in \mathbb{R}^K$. Our goal is to combine the evidence together, to compute $p(\boldsymbol{z}|\boldsymbol{y})$. This is known as **sensor fusion**.

We now give a simple example, where there are just two sensors, so $\boldsymbol{y}_1 \sim \mathcal{N}(\boldsymbol{z}, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{y}_2 \sim \mathcal{N}(\boldsymbol{z}, \boldsymbol{\Sigma}_2)$. Pictorially, we can represent this example as $\boldsymbol{y}_1 \leftarrow \boldsymbol{z} \rightarrow \boldsymbol{y}_2$. We can combine $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ into a single vector $\boldsymbol{y}$, so the model can be represented as $\boldsymbol{z} \rightarrow [\boldsymbol{y}_1, \boldsymbol{y}_2]$, where $p(\boldsymbol{y}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{y}|\mathbf{W}\boldsymbol{z}, \boldsymbol{\Sigma}_y)$, where $\mathbf{W} = [\mathbf{I}; \mathbf{I}]$ and $\boldsymbol{\Sigma}_y = [\boldsymbol{\Sigma}_1, \mathbf{0}; \mathbf{0}, \boldsymbol{\Sigma}_2]$ are block-structured matrices. We can then apply Bayes' rule for Gaussians to compute $p(\boldsymbol{z}|\boldsymbol{y})$.

Figure 3.10(a) gives a 2d example, where we set $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = 0.01\mathbf{I}_2$, so both sensors are equally reliable. In this case, the posterior mean is halfway between the two observations, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. In Figure 3.10(b), we set $\boldsymbol{\Sigma}_1 = 0.05\mathbf{I}_2$ and $\boldsymbol{\Sigma}_2 = 0.01\mathbf{I}_2$, so sensor 2 is more reliable than sensor 1. In this case, the posterior mean is closer to $\boldsymbol{y}_2$. In Figure 3.10(c), we set

$$\boldsymbol{\Sigma}_1 = 0.01 \begin{pmatrix} 10 & 1 \\ 1 & 1 \end{pmatrix}, \quad \boldsymbol{\Sigma}_2 = 0.01 \begin{pmatrix} 1 & 1 \\ 1 & 10 \end{pmatrix} \tag{3.70}$$

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

3.4 The exponential family *

In this section, we define the **exponential family**, which includes many common probability distributions. The exponential family plays a crucial role in statistics and machine learning. In this book, we mainly use it in the context of generalized linear models, which we discuss in Chapter 12. We will see more applications of the exponential family in the sequel to this book, [Mur23].

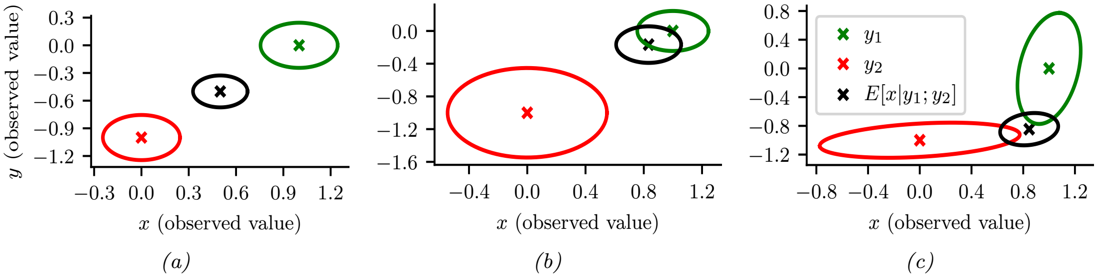


Figure 3.10: We observe $\boldsymbol{y}_1 = (0, -1)$ (red cross) and $\boldsymbol{y}_2 = (1, 0)$ (green cross) and estimate $\mathbb{E}[\boldsymbol{z}|\boldsymbol{y}_1, \boldsymbol{y}_2]$ (black cross). (a) Equally reliable sensors, so the posterior mean estimate is in between the two circles. (b) Sensor 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. Generated by sensor_fusion_2d.ipynb.

3.4.1 Definition

Consider a family of probability distributions parameterized by $\boldsymbol{\eta} \in \mathbb{R}^K$ with fixed support over $\mathcal{Y}^D \subseteq \mathbb{R}^D$. We say that the distribution $p(\boldsymbol{y}|\boldsymbol{\eta})$ is in the **exponential family** if its density can be written in the following way:

$$p(\boldsymbol{y}|\boldsymbol{\eta}) \triangleq \frac{1}{Z(\boldsymbol{\eta})} h(\boldsymbol{y}) \exp[\boldsymbol{\eta}^{\mathsf{T}} \mathcal{T}(\boldsymbol{y})] = h(\boldsymbol{y}) \exp[\boldsymbol{\eta}^{\mathsf{T}} \mathcal{T}(\boldsymbol{y}) - A(\boldsymbol{\eta})] \tag{3.71}$$

where $h(\boldsymbol{y})$ is a scaling constant (also known as the **base measure**, often 1), $\mathcal{T}(\boldsymbol{y}) \in \mathbb{R}^K$ are the **sufficient statistics**, $\boldsymbol{\eta}$ are the **natural parameters** or **canonical parameters**, $Z(\boldsymbol{\eta})$ is a normalization constant known as the **partition function**, and $A(\boldsymbol{\eta}) = \log Z(\boldsymbol{\eta})$ is the **log partition function**. One can show that A is a convex function over the concave set $\Omega \triangleq \{\boldsymbol{\eta} \in \mathbb{R}^K : A(\boldsymbol{\eta}) < \infty\}$.

It is convenient if the natural parameters are independent of each other. Formally, we say that an exponential family is **minimal** if there is no $\boldsymbol{\eta} \in \mathbb{R}^K \setminus \{0\}$ such that $\boldsymbol{\eta}^{\mathsf{T}} \mathcal{T}(\boldsymbol{y}) = 0$. This last condition can be violated in the case of multinomial distributions, because of the sum to one constraint on the parameters; however, it is easy to reparameterize the distribution using $K - 1$ independent parameters, as we show below.

Equation (3.71) can be generalized by defining $\boldsymbol{\eta} = f(\boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is some other, possibly smaller, set of parameters. In this case, the distribution has the form

$$p(\boldsymbol{y}|\boldsymbol{\phi}) = h(\boldsymbol{y}) \exp[f(\boldsymbol{\phi})^{\mathsf{T}} \mathcal{T}(\boldsymbol{y}) - A(f(\boldsymbol{\phi}))] \tag{3.72}$$

If the mapping from $\boldsymbol{\phi}$ to $\boldsymbol{\eta}$ is nonlinear, we call this a **curved exponential family**. If $\boldsymbol{\eta} = f(\boldsymbol{\phi}) = \boldsymbol{\phi}$, the model is said to be in **canonical form**. If, in addition, $\mathcal{T}(\boldsymbol{y}) = \boldsymbol{y}$, we say this is a **natural exponential family** or **NEF**. In this case, it can be written as

$$p(\boldsymbol{y}|\boldsymbol{\eta}) = h(\boldsymbol{y}) \exp[\boldsymbol{\eta}^{\mathsf{T}} \boldsymbol{y} - A(\boldsymbol{\eta})] \tag{3.73}$$

3.4.2 Example

As a simple example, let us consider the Bernoulli distribution. We can write this in exponential family form as follows:

$$\begin{aligned}
\mathrm{Ber}(y|\mu) &= \mu^y(1-\mu)^{1-y} && (3.74)\\
&= \exp[y\log(\mu) + (1-y)\log(1-\mu)] && (3.75)\\
&= \exp[\mathcal{T}(y)^\mathsf{T}\boldsymbol{\eta}] && (3.76)
\end{aligned}$$

where $\mathcal{T}(y) = [\mathbb{I}(y=1), \mathbb{I}(y=0)]$, $\boldsymbol{\eta} = [\log(\mu), \log(1-\mu)]$, and μ is the mean parameter. However, this is an **over-complete representation** since there is a linear dependence between the features. We can see this as follows:

$$\mathbf{1}^\mathsf{T}\mathcal{T}(y) = \mathbb{I}(y=0) + \mathbb{I}(y=1) = 1 \tag{3.77}$$

If the representation is overcomplete, $\boldsymbol{\eta}$ is not uniquely identifiable. It is common to use a **minimal representation**, which means there is a unique $\boldsymbol{\eta}$ associated with the distribution. In this case, we can just define

$$\mathrm{Ber}(y|\mu) = \exp\left[y\log\left(\frac{\mu}{1-\mu}\right) + \log(1-\mu)\right] \tag{3.78}$$

We can put this into exponential family form by defining

$$\begin{aligned}
\eta &= \log\left(\frac{\mu}{1-\mu}\right) && (3.79)\\
\mathcal{T}(y) &= y && (3.80)\\
A(\eta) &= -\log(1-\mu) = \log(1+e^\eta) && (3.81)\\
h(y) &= 1 && (3.82)
\end{aligned}$$

We can recover the mean parameter μ from the canonical parameter η using

$$\mu = \sigma(\eta) = \frac{1}{1+e^{-\eta}} \tag{3.83}$$

which we recognize as the logistic (sigmoid) function.

See the sequel to this book, [Mur23], for more examples.

3.4.3 Log partition function is cumulant generating function

The first and second **cumulants** of a distribution are its mean $\mathbb{E}[Y]$ and variance $\mathbb{V}[Y]$, whereas the first and second moments are $\mathbb{E}[Y]$ and $\mathbb{E}\left[Y^2\right]$. We can also compute higher order cumulants (and moments). An important property of the exponential family is that derivatives of the log partition function can be used to generate all the **cumulants** of the sufficient statistics. In particular, the first and second cumulants are given by

$$\begin{aligned}
\nabla A(\boldsymbol{\eta}) &= \mathbb{E}[\mathcal{T}(\boldsymbol{y})] && (3.84)\\
\nabla^2 A(\boldsymbol{\eta}) &= \mathrm{Cov}[\mathcal{T}(\boldsymbol{y})] && (3.85)
\end{aligned}$$

From the above result, we see that the Hessian is positive definite, and hence $A(\boldsymbol{\eta})$ is convex in $\boldsymbol{\eta}$. Since the log likelihood has the form $\log p(\boldsymbol{y}|\boldsymbol{\eta}) = \boldsymbol{\eta}^\mathsf{T}\mathcal{T}(\boldsymbol{y}) - A(\boldsymbol{\eta}) + \text{const}$, we see that this is concave, and hence the MLE has a unique global maximum.

3.4.4 Maximum entropy derivation of the exponential family

Suppose we want to find a distribution $p(\boldsymbol{x})$ to describe some data, where all we know are the expected values (F_k) of certain features or functions $f_k(\boldsymbol{x})$:

$$\int d\boldsymbol{x}\, p(\boldsymbol{x}) f_k(\boldsymbol{x}) = F_k \tag{3.86}$$

For example, f_1 might compute x, f_2 might compute x^2, making F_1 the empirical mean and F_2 the empirical second moment. Our prior belief in the distribution is $q(x)$.

To formalize what we mean by "least number of assumptions", we will search for the distribution that is as close as possible to our prior $q(\boldsymbol{x})$, in the sense of KL divergence (Section 6.2), while satisfying our constraints:

$$p = \operatorname*{argmin}_p D_{\mathbb{KL}}\left(p \parallel q\right) \text{ subject to constraints} \tag{3.87}$$

If we use a uniform prior, $q(\boldsymbol{x}) \propto 1$, minimizing the KL divergence is equivalent to maximizing the entropy (Section 6.1):

$$p = \operatorname*{argmax}_p \mathbb{H}(p) \text{ subject to constraints} \tag{3.88}$$

The result is called a **maximum entropy model**.

To minimize the KL subject to the constraints in Equation (3.86), and the constraint that $p(\boldsymbol{x}) \geq 0$ and $\sum_{\boldsymbol{x}} p(\boldsymbol{x}) = 1$, we will use Lagrange multipliers (see Section 8.5.1). The Lagrangian is given by

$$J(p, \boldsymbol{\lambda}) = -\sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} + \lambda_0 \left(1 - \sum_{\boldsymbol{x}} p(\boldsymbol{x})\right) + \sum_k \lambda_k \left(F_k - \sum_{\boldsymbol{x}} p(\boldsymbol{x}) f_k(\boldsymbol{x})\right) \tag{3.89}$$

We can use the calculus of variations to take derivatives wrt the function p, but we will adopt a simpler approach and treat $\boldsymbol{p}$ as a fixed length vector (since we are assuming that $\boldsymbol{x}$ is discrete). Then we have

$$\frac{\partial J}{\partial p_c} = -1 - \log \frac{p(x=c)}{q(x=c)} - \lambda_0 - \sum_k \lambda_k f_k(x=c) \tag{3.90}$$

Setting $\frac{\partial J}{\partial p_c} = 0$ for each c yields

$$p(\boldsymbol{x}) = \frac{q(\boldsymbol{x})}{Z} \exp\left(-\sum_k \lambda_k f_k(\boldsymbol{x})\right) \tag{3.91}$$

where we have defined $Z \triangleq e^{1+\lambda_0}$. Using the sum-to-one constraint, we have

$$1 = \sum_{\boldsymbol{x}} p(\boldsymbol{x}) = \frac{1}{Z} \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \exp\left(-\sum_k \lambda_k f_k(\boldsymbol{x})\right) \tag{3.92}$$

Hence the normalization constant is given by

$$Z = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \exp\left(-\sum_k \lambda_k f_k(\boldsymbol{x})\right) \tag{3.93}$$

This has exactly the form of the exponential family, where $\boldsymbol{f}(\boldsymbol{x})$ is the vector of sufficient statistics, $-\boldsymbol{\lambda}$ are the natural parameters, and $q(\boldsymbol{x})$ is our base measure.

For example, if the features are $f_1(x) = x$ and $f_2(x) = x^2$, and we want to match the first and second moments, we get the Gaussian disribution.

3.5 Mixture models

One way to create more complex probability models is to take a convex combination of simple distributions. This is called a **mixture model**. This has the form

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{y}) \tag{3.94}$$

where p_k is the k'th mixture component, and π_k are the mixture weights which satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

We can re-express this model as a hierarchical model, in which we introduce the discrete **latent variable** $z \in \{1, \ldots, K\}$, which specifies which distribution to use for generating the output $\boldsymbol{y}$. The prior on this latent variable is $p(z = k|\boldsymbol{\theta}) = \pi_k$, and the conditional is $p(\boldsymbol{y}|z = k, \boldsymbol{\theta}) = p_k(\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{\theta}_k)$. That is, we define the following joint model:

$$p(z|\boldsymbol{\theta}) = \mathrm{Cat}(z|\boldsymbol{\pi}) \tag{3.95}$$

$$p(\boldsymbol{y}|z = k, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{\theta}_k) \tag{3.96}$$

where $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_K, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ are all the model parameters. The "generative story" for the data is that we first sample a specific component z, and then we generate the observations $\boldsymbol{y}$ using the parameters chosen according to the value of z. By marginalizing out z, we recover Equation (3.94):

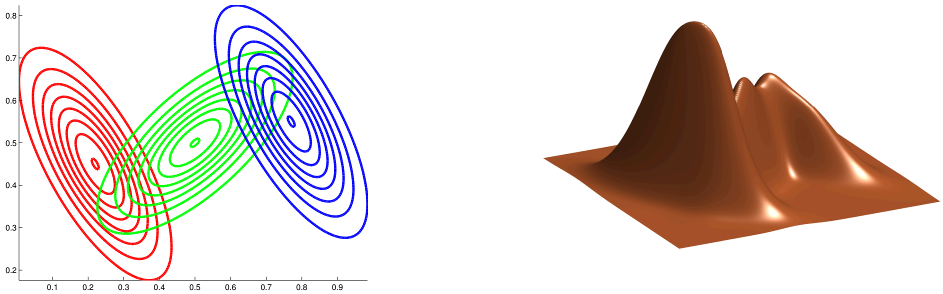
$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K} p(z = k|\boldsymbol{\theta}) p(\boldsymbol{y}|z = k, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{y}|\boldsymbol{\theta}_k) \tag{3.97}$$

We can create different kinds of mixture model by varying the base distribution p_k, as we illustrate below.

3.5.1 Gaussian mixture models

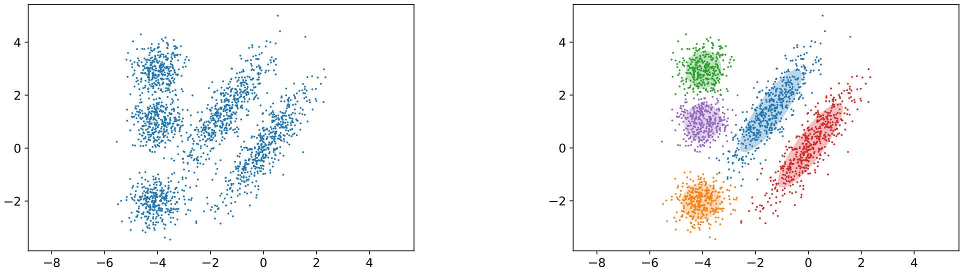
A **Gaussian mixture model** or **GMM**, also called a **mixture of Gaussians** (**MoG**), is defined as follows:

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.98}$$



(a) (b)

Figure 3.11: A mixture of 3 Gaussians in 2d. (a) We show the contours of constant probability for each component in the mixture. (b) A surface plot of the overall density. Adapted from Figure 2.23 of [Bis06]. Generated by gmm_plot_demo.ipynb



(a) (b)

Figure 3.12: (a) Some data in 2d. (b) A possible clustering using $K = 5$ clusters computed using a GMM. Generated by gmm_2d.ipynb.

In Figure 3.11 we show the density defined by a mixture of 3 Gaussians in 2d. Each mixture component is represented by a different set of elliptical contours. If we let the number of mixture components grow sufficiently large, a GMM can approximate any smooth distribution over $\mathbb{R}^D$.

GMMs are often used for unsupervised **clustering** of real-valued data samples $\boldsymbol{y}_n \in \mathbb{R}^D$. This works in two stages. First we fit the model e.g., by computing the MLE $\hat{\boldsymbol{\theta}} = \operatorname{argmax} \log p(\mathcal{D}|\boldsymbol{\theta})$, where $\mathcal{D} = \{\boldsymbol{y}_n : n = 1 : N\}$. (We discuss how to compute this MLE in Section 8.7.3.) Then we associate each data point $\boldsymbol{y}_n$ with a discrete latent or hidden variable $z_n \in \{1, \ldots, K\}$ which specifies the identity of the mixture component or cluster which was used to generate $\boldsymbol{y}_n$. These latent identities are unknown, but we can compute a posterior over them using Bayes rule:

$$r_{nk} \triangleq p(z_n = k|\boldsymbol{y}_n, \boldsymbol{\theta}) = \frac{p(z_n = k|\boldsymbol{\theta})p(\boldsymbol{y}_n|z_n = k, \boldsymbol{\theta})}{\sum_{k'=1}^{K} p(z_n = k'|\boldsymbol{\theta})p(\boldsymbol{y}_n|z_n = k', \boldsymbol{\theta})} \tag{3.99}$$

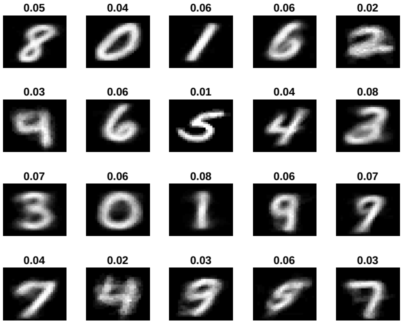


Figure 3.13: We fit a mixture of 20 Bernoullis to the binarized MNIST digit data. We visualize the estimated cluster means $\hat{\boldsymbol{\mu}}_k$. The numbers on top of each image represent the estimated mixing weights $\hat{\pi}_k$. No labels were used when training the model. Generated by mix_bernoulli_em_mnist.ipynb.

The quantity r_{nk} is called the **responsibility** of cluster k for data point n. Given the responsibilities, we can compute the most probable cluster assignment as follows:

$$\hat{z}_n = \arg\max_k r_{nk} = \arg\max_k \left[\log p(\boldsymbol{y}_n|z_n = k, \boldsymbol{\theta}) + \log p(z_n = k|\boldsymbol{\theta})\right] \tag{3.100}$$

This is known as **hard clustering**. (If we use the responsibilities to fractionally assign each data point to different clusters, it is called **soft clustering**.) See Figure 3.12 for an example.

If we have a uniform prior over z_n, and we use spherical Gaussians with $\boldsymbol{\Sigma}_k = \mathbf{I}$, the hard clustering problem reduces to

$$z_n = \operatorname*{argmin}_k ||\boldsymbol{y}_n - \hat{\boldsymbol{\mu}}_k||_2^2 \tag{3.101}$$

In other words, we assign each data point to its closest centroid, as measured by Euclidean distance. This is the basis of the **K-means clustering** algorithm, which we discuss in Section 21.3.

3.5.2 Bernoulli mixture models

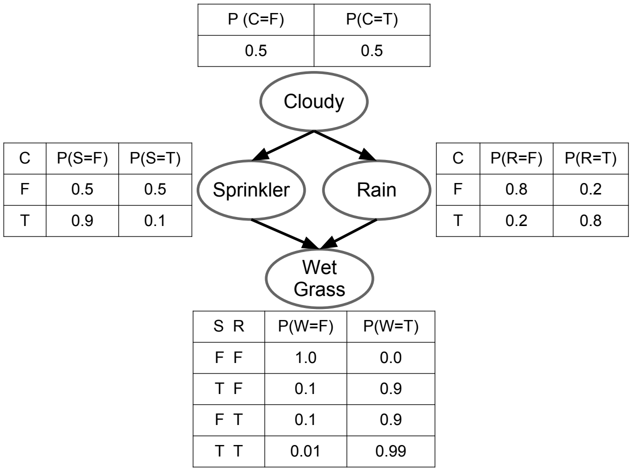
If the data is binary valued, we can use a **Bernoulli mixture model** or **BMM** (also called a **mixture of Bernoullis**), where each mixture component has the following form:

$$p(\boldsymbol{y}|z = k, \boldsymbol{\theta}) = \prod_{d=1}^{D} \text{Ber}(y_d|\mu_{dk}) = \prod_{d=1}^{D} \mu_{dk}^{y_d}(1 - \mu_{dk})^{1-y_d} \tag{3.102}$$

Here μ_{dk} is the probability that bit d turns on in cluster k.

As an example, we fit a BMM using $K = 20$ components to the MNIST dataset (Section 3.5.2). (We use the EM algorithm to do this fitting, which is similar to EM for GMMs discussed in Section 8.7.3; however we can also use SGD to fit the model, which is more efficient for large datasets.[2]) The

[2] For the SGD code, see mix_bernoulli_sgd_mnist.ipynb.



P (C=F)	P(C=T)
0.5	0.5

C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

S R	P(W=F)	P(W=T)
F F	1.0	0.0
T F	0.1	0.9
F T	0.1	0.9
T T	0.01	0.99

Figure 3.14: Water sprinkler PGM with corresponding binary CPTs. T and F stand for true and false.

resulting parameters for each mixture component (i.e., $\boldsymbol{\mu}_k$ and π_k) are shown in Figure 3.13. We see that the model has "discovered" a representation of each type of digit. (Some digits are represented multiple times, since the model does not know the "true" number of classes. See Section 21.3.7 for more information on how to choose the number K of mixture components.)

3.6 Probabilistic graphical models *

> I basically know of two principles for treating complicated systems in simple ways: the first is the principle of modularity and the second is the principle of abstraction. I am an apologist for computational probability in machine learning because I believe that probability theory implements these two principles in deep and intriguing ways — namely through factorization and through averaging. Exploiting these two mechanisms as fully as possible seems to me to be the way forward in machine learning. — Michael Jordan, 1997 (quoted in [Fre98]).

We have now introduced a few simple probabilistic building blocks. In Section 3.3, we showed one way to combine some Gaussian building blocks to build a high dimensional distribution $p(\boldsymbol{y})$ from simpler parts, namely the marginal $p(\boldsymbol{y}_1)$ and the conditional $p(\boldsymbol{y}_2|\boldsymbol{y}_1)$. This idea can be extended to define joint distributions over sets of many random variables. The key assumption we will make is that some variables are **conditionally independent** of others. We will represent our CI assumptions using graphs, as we briefly explain below. (See the sequel to this book, [Mur23], for more information.)

3.6.1 Representation

A **probabilistic graphical model** or **PGM** is a joint probability distribution that uses a graph structure to encode conditional independence assumptions. When the graph is a **directed acyclic**

graph or **DAG**, the model is sometimes called a **Bayesian network**, although there is nothing inherently Bayesian about such models.

The basic idea in PGMs is that each node in the graph represents a random variable, and each edge represents a direct dependency. More precisely, each lack of edge represents a conditional independency. In the DAG case, we can number the nodes in **topological order** (parents before children), and then we connect them such that each node is conditionally independent of all its predecessors given its parents:

$$Y_i \perp \mathbf{Y}_{\text{pred}(i)\setminus\text{pa}(i)}|\mathbf{Y}_{\text{pa}(i)} \tag{3.103}$$

where pa(i) are the parents of node i, and pred(i) are the predecessors of node i in the ordering. (This is called the **ordered Markov property**.) Consequently, we can represent the joint distribution as follows:

$$p(\mathbf{Y}_{1:N_G}) = \prod_{i=1}^{N_G} p(Y_i|\mathbf{Y}_{\text{pa}(i)}) \tag{3.104}$$

where N_G is the number of nodes in the graph.

3.6.1.1 Example: water sprinkler network

Suppose we want to model the dependencies between 4 random variables: C (whether it is cloudy season or not), R (whether it is raining or not), S (whether the water sprinkler is on or not), and W (whether the grass is wet or not). We know that the cloudy season makes rain more likely, so we add a $C \rightarrow R$ arc. We know that the cloudy season makes turning on a water sprinkler less likely, so we add a $C \rightarrow S$ arc. Finally, we know that either rain or sprinklers can cause the grass to get wet, so we add $S \rightarrow W$ and $R \rightarrow W$ edges.

Formally, this defines the following joint distribution:

$$p(C, S, R, W) = p(C)p(S|C)p(R|C, \cancel{S})p(W|S, R, \cancel{C}) \tag{3.105}$$

where we strike through terms that are not needed due to the conditional independence properties of the model.

Each term $p(Y_i|\mathbf{Y}_{\text{pa}(i)})$ is a called the **conditional probability distribution** or **CPD** for node i. This can be any kind of distribution we like. In Figure 3.14, we assume each CPD is a conditional categorical distribution, which can be represented as a **conditional probability table** or **CPT**. We can represent the i'th CPT as follows:

$$\theta_{ijk} \triangleq p(Y_i = k|\mathbf{Y}_{\text{pa}(i)} = j) \tag{3.106}$$

This satisfies the properties $0 \leq \theta_{ijk} \leq 1$ and $\sum_{k=1}^{K_i} \theta_{ijk} = 1$ for each row j. Here i indexes nodes, $i \in [N_G]$; k indexes node states, $k \in [K_i]$, where K_i is the number of states for node i; and j indexes joint parent states, $j \in [J_i]$, where $J_i = \prod_{p \in \text{pa}(i)} K_p$. For example, the wet grass node has 2 binary parents, so there are 4 parent states.

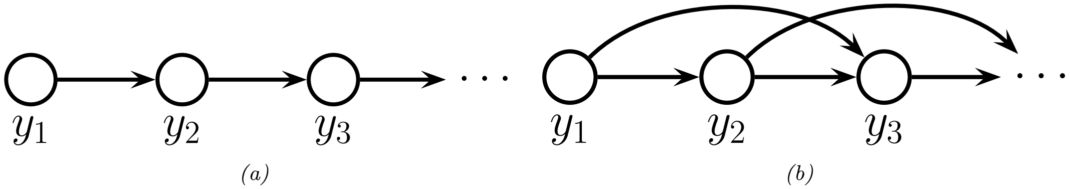


Figure 3.15: Illustration of first and second order autoregressive (Markov) models.

3.6.1.2 Example: Markov chain

Suppose we want to create a joint probability distribution over variable-length sequences, $p(y_{1:T})$. If each variable y_t represents a word from a vocabulary with K possible values, so $y_t \in \{1, \ldots, K\}$, the resulting model represents a distribution over possible sentences of length T; this is often called a **language model**.

By the chain rule of probability, we can represent any joint distribution over T variables as follows:

$$p(\boldsymbol{y}_{1:T}) = p(y_1)p(y_2|y_1)p(y_3|y_2, y_1)p(y_4|y_3, y_2, y_1)\ldots = \prod_{t=1}^{T} p(y_t|\boldsymbol{y}_{1:t-1}) \tag{3.107}$$

Unfortunately, the number of parameters needed to represent each conditional distribution $p(y_t|\boldsymbol{y}_{1:t-1})$ grows exponentially with t. However, suppose we make the conditional independence assumption that the future, $\boldsymbol{y}_{t+1:T}$, is independent of the past, $\boldsymbol{y}_{1:t-1}$, given the present, y_t. This is called the **first order Markov condition**, and is repesented by the PGM in Figure 3.15(a). With this assumption, we can write the joint distribution as follows:

$$p(\boldsymbol{y}_{1:T}) = p(y_1)p(y_2|y_1)p(y_3|y_2)p(y_4|y_3)\ldots = p(y_1)\prod_{t=2}^{T} p(y_t|y_{t-1}) \tag{3.108}$$

This is called a **Markov chain**, **Markov model** or **autoregressive model** of order 1.

The function $p(y_t|y_{t-1})$ is called the **transition function**, **transition kernel** or **Markov kernel**. This is just a conditional distribution over the states at time t given the state at time $t-1$, and hence it satisfies the conditions $p(y_t|y_{t-1}) \geq 0$ and $\sum_{k=1}^{K} p(y_t = k|y_{t-1} = j) = 1$. We can represent this CPT as a **stochastic matrix**, $A_{jk} = p(y_t = k|y_{t-1} = j)$, where each row sums to 1. This is known as the **state transition matrix**. We assume this matrix is the same for all time steps, so the model is said to be **homogeneous**, **stationary**, or **time-invariant**. This is an example of **parameter tying**, since the same parameter is shared by multiple variables. This assumption allows us to model an arbitrary number of variables using a fixed number of parameters.

The first-order Markov assumption is rather strong. Fortunately, we can easily generalize first-order models to depend on the last M observations, thus creating a model of order (memory length) M:

$$p(\boldsymbol{y}_{1:T}) = p(\boldsymbol{y}_{1:M}) \prod_{t=M+1}^{T} p(y_t|\boldsymbol{y}_{t-M:t-1}) \tag{3.109}$$

This is called an M**'th order Markov model**. For example, if $M = 2$, y_t depends on y_{t-1} and y_{t-2}, as shown in Figure 3.15(b). This is called a **trigram model**, since it models the distribution over word triples. If we use $M = 1$, we get a **bigram model**, which models the distribution over word pairs.

For large vocabulary sizes, the number of parameters needed to estimate the conditional distributions for M-gram models for large M can become prohibitive. In this case, we need to make additional assumptions beyond conditional independence. For example, we can assume that $p(y_t|\boldsymbol{y}_{t-M:t-1})$ can be represented as a low-rank matrix, or in terms of some kind of neural network. This is called a **neural language model**. See Chapter 15 for details.

3.6.2 Inference

A PGM defines a joint probability distribution. We can therefore use the rules of marginalization and conditioning to compute $p(\mathbf{Y}_i|\mathbf{Y}_j = \boldsymbol{y}_j)$ for any sets of variables i and j. Efficient algorithms to perform this computation are discussed in the sequel to this book, [Mur23].

For example, consider the water sprinkler example in Figure 3.14. Our prior belief that it has rained is given by $p(R = 1) = 0.5$. If we see that the grass is wet, then our posterior belief that it has rained changes to $p(R = 1|W = 1) = 0.7079$. Now suppose we also notice the water sprinkler was turned on: our belief that it rained goes down to $p(R = 1|W = 1, S = 1) = 0.3204$. This negative mutual interaction between multiple causes of some observations is called the **explaining away** effect, also known as **Berkson's paradox**. (See sprinkler_pgm.ipynb for some code that reproduces these calculations.)

3.6.3 Learning

If the parameters of the CPDs are unknown, we can view them as additional random variables, add them as nodes to the graph, and then treat them as **hidden variables** to be inferred. Figure 3.16(a) shows a simple example, in which we have N iid random variables, $\boldsymbol{y}_n$, all drawn from the same distribution with common parameter $\boldsymbol{\theta}$. (The **shaded nodes** represent observed values, whereas the unshaded (hollow) nodes represent latent variables or parameters.)

More precisely, the model encodes the following "generative story" about the data:

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}) \tag{3.110}$$

$$\boldsymbol{y}_n \sim p(\boldsymbol{y}|\boldsymbol{\theta}) \tag{3.111}$$

where $p(\boldsymbol{\theta})$ is some (unspecified) prior over the parameters, and $p(\boldsymbol{y}|\boldsymbol{\theta})$ is some specified likelihood function. The corresponding joint distribution has the form

$$p(\mathcal{D}, \boldsymbol{\theta}) = p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta}) \tag{3.112}$$

where $\mathcal{D} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$. By virtue of the iid assumption, the likelihood can be rewritten as follows:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{\theta}) \tag{3.113}$$

Notice that the order of the data vectors is not important for defining this model, i.e., we can permute the numbering of the leaf nodes in the PGM. When this property holds, we say that the data is **exchangeable**.

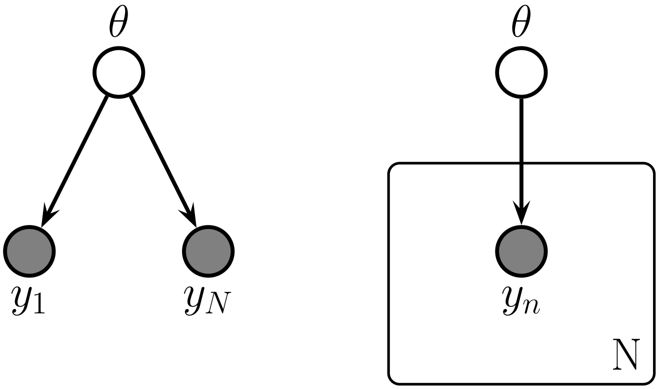


Figure 3.16: Left: data points $\boldsymbol{y}_n$ are conditionally independent given $\boldsymbol{\theta}$. Right: Same model, using plate notation. This represents the same model as the one on the left, except the repeated $\boldsymbol{y}_n$ nodes are inside a box, known as a plate; the number in the lower right hand corner, N, specifies the number of repetitions of the $\boldsymbol{y}_n$ node.

3.6.3.1 Plate notation

In Figure 3.16(a), we see that the $\boldsymbol{y}$ nodes are repeated N times. To avoid visual clutter, it is common to use a form of **syntactic sugar** called **plates**. This is a notational convention in which we draw a little box around the repeated variables, with the understanding that nodes within the box will get repeated when the model is **unrolled**. We often write the number of copies or repetitions in the bottom right corner of the box. This is illustrated in Figure 3.16(b). This notation is widely used to represent certain kinds of Bayesian model.

Figure 3.17 shows a more interesting example, in which we represent a GMM (Section 3.5.1) as a graphical model. We see that this encodes the joint distribution

$$p(\boldsymbol{y}_{1:N}, \boldsymbol{z}_{1:N}, \boldsymbol{\theta}) = p(\boldsymbol{\pi}) \left[\prod_{k=1}^{K} p(\boldsymbol{\mu}_k) p(\boldsymbol{\Sigma}_k) \right] \left[\prod_{n=1}^{N} p(z_n|\boldsymbol{\pi}) p(\boldsymbol{y}_n|z_n, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K}) \right] \tag{3.114}$$

We see that the latent variables z_n as well as the unknown paramters, $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K})$, are all shown as unshaded nodes.

3.7 Exercises

Exercise 3.1 [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 $\mathbb{V}[X] = (b - a)^2/12$.

Exercise 3.2 [Correlation coefficient is between -1 and +1]

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

Exercise 3.3 [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 $\rho(X, Y) = -1$.

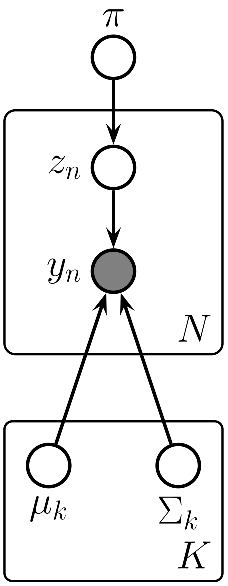


Figure 3.17: A Gaussian mixture model represented as a graphical model.

Exercise 3.4 [Linear combinations of random variables]

Let $\boldsymbol{x}$ be a random vector with mean $\boldsymbol{m}$ and covariance matrix Σ. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices.

a. Derive the covariance matrix of $\mathbf{A}\boldsymbol{x}$.

b. Show that $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$.

c. Derive an expression for $\mathbb{E}\left[\boldsymbol{x}^T\mathbf{A}\boldsymbol{x}\right]$.

Exercise 3.5 [Gaussian vs *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 $Y \sim \mathcal{N}(0,1)$.

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

$$\text{Cov}\left[X,Y\right] = \mathbb{E}\left[XY\right] - \mathbb{E}\left[X\right]\mathbb{E}\left[Y\right] \tag{3.115}$$

and the **rule of iterated expectation**

$$\mathbb{E}\left[XY\right] = \mathbb{E}\left[\mathbb{E}\left[XY|W\right]\right] \tag{3.116}$$

Exercise 3.6 [Normalization constant for a multidimensional Gaussian]

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

$$(2\pi)^{d/2}|\mathbf{\Sigma}|^{\frac{1}{2}} = \int \exp(-\frac{1}{2}(\boldsymbol{x}-\mu)^T\mathbf{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}))d\boldsymbol{x} \tag{3.117}$$

Hint: diagonalize $\mathbf{\Sigma}$ and use the fact that $|\mathbf{\Sigma}| = \prod_i \lambda_i$ 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 3.7 [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 3.8 [Show that the Student distribution can be written as a Gaussian scale mixture]

Show that a Student distribution can be written as a **Gaussian scale mixture**, where we use a Gamma mixing distribution on the precision α, i.e.

$$p(x|\mu, a, b) = \int_0^\infty \mathcal{N}(x|\mu, \alpha^{-1})\text{Ga}(\alpha|a, b)d\alpha \tag{3.118}$$

This can be viewed as an infinite mixture of Gaussians, with different precisions.