An Introduction to Probabilistic Graphical Models

Michael I. Jordan
University of California, Berkeley

June 30, 2003

Chapter 7

Linear classification

In this chapter we continue our discussion of elementary building blocks for graphical models, treating the case of a discrete node taking on a finite number of values. As in Chapter 6, our interest is in the conditional relationship between the node Y and a vector of explanatory variables X. We explore a number of possible representations for the conditional probability $p(y \mid x)$.

What form should our model of $p(y \mid x)$ take in the case of discrete Y? If X is also discrete, then we might consider models in which all possible combinations of X and Y are represented in a table. We will indeed consider such a model in this chapter; however, it is important to keep in mind that the size of such a table is exponential in the number of components of X, and we would like to develop models to handle the (commonplace) situation in which this number is large. Moreover, we wish to develop tools that allow for continuous-valued X. In either case a natural first step is to try and mimic what we did with regression, exploiting the simplicity and mathematical convenience of linearity assumptions. It is unclear, however, how to represent the conditional expectation of Y—a number between zero and one for Bernoulli and multinomial variables—within the framework of a linear model. Some sort of nonlinearity seems to be needed, but which nonlinearity? Does introducing such a nonlinearity leave us with any role for linearity?

One way to help organize our thinking on these issues is to recall that we have already seen problems involving discrete Y in Chapter 5. In particular, in our discussion of classification models in that chapter, we found it useful to explore the relationship between two kinds of models: *discriminative models*—in which Y is the child of X—and *generative models*—in which Y is the parent of X. While the former approach represents $p(y \mid x)$ *explicitly*, the latter approach makes use of Bayes rule to represent the posterior probability $p(y \mid x)$ *implicitly*, in terms of the class-conditional probability $p(x \mid y)$ and the prior $p(y)$. Thus we can begin to get ideas for representations of $p(y \mid x)$ by studying generative models in which Y is a parent of X, and using Bayes rule to invert the model and thereby calculate the corresponding posterior probability $p(y \mid x)$. This approach will allow us to achieve some of the goals that we alluded to above—it will suggest a certain basic mathematical structure in which linearity plays a role, and it will cope with both discrete-valued and continuous-valued X. Moreover, it will suggest a natural "upgrade path" to more complex models.

In this chapter we retain our assumption from the previous chapter that both X and Y are observed in our data set. We cast our presentation within the context of classification, where as before we refer to Y and X as the "class label" and the "feature vector," respectively. We will fill in some of the details that were glossed over in Chapter 5 regarding the parameterization and estimation of generative and discriminative approaches to the classification problem. We present maximum likelihood methods for parameter estimation in both frameworks.

While we place our activity in this chapter within the framework of classification, it is worth noting that there are aspects of classification problems that fall beyond the scope of our discussion. In particular, our goal in this chapter is that of obtaining a model of the conditional probability $p(y \mid x)$. While $p(y \mid x)$ is a desirable quantity to model in a classification setting, it is also true that classification involves something more than evaluating a probability—in particular, classification involves making a *decision*. We can threshold the probability distribution $p(y \mid x)$ to obtain a decision, but this is only one possible way to use this probability; perhaps there are others. Indeed, perhaps there are some decisions which are in some sense more costly than others; our thresholding scheme should be sensitive to such costs. Moreover, we can imagine classification algorithms that do not make use of posterior probability $p(y \mid x)$ at all; rather they go directly from a data set to a decision rule. Evaluating these alternatives appropriately requires the mathematical framework of *decision theory*. In particular a decision-theoretic approach to classification allows us to specify costs associated with decisions and to evaluate alternative approaches to forming decision rules. We will return to these issues in Chapter 27, where we present a full treatment of decision theory in the graphical model setting. In that discussion we will in fact show that a reasonable first step in classification problems is to obtain a model of the conditional probability $p(y \mid x)$.

It is also worth noting that there is a flip side to this coin—there are problems other than classification problems for which the methods of this chapter are useful. In particular in Chapter 10 we discuss models that are structurally identical to the models in this chapter, but for which Y is no longer assumed to be observed; that is, for which Y is a latent variable. The results that we obtain here will play an important role in that chapter.

7.1 Linear regression and linear classification

A discrete-valued node can be viewed as a special case of a real-valued node, and this leads one to wonder why we need a separate treatment of discrete nodes. In particular, why not use the regression methods that we developed in Chapter 6 to solve classification problems?

To see some of the problems that arise if we pursue this approach, consider the simple case of a binary classification problem with a scalar-valued feature variable X. Let us represent the class label with a real-valued variable Y, with $Y = 0$ and $Y = 1$ representing the two classes. Figure 7.1 presents an example of such a problem, with the data pairs (x_n, y_n) represented as points in the plane. The linear regression fit to these data is also shown in the figure. Note that even though the data $\{y_n\}$ are restricted to the values zero and one, the fitted line is not restricted to these values. How are we to interpret this line? In Chapter 6 we showed that the linear regression fit is a conditional mean—the expected value of Y conditioned on the observed value of X. For an indicator random variable Y the expected value is the same as the probability that the variable takes on the value 1. The fact that the fitted line in Figure 7.1 strays outside of the range $(0, 1)$

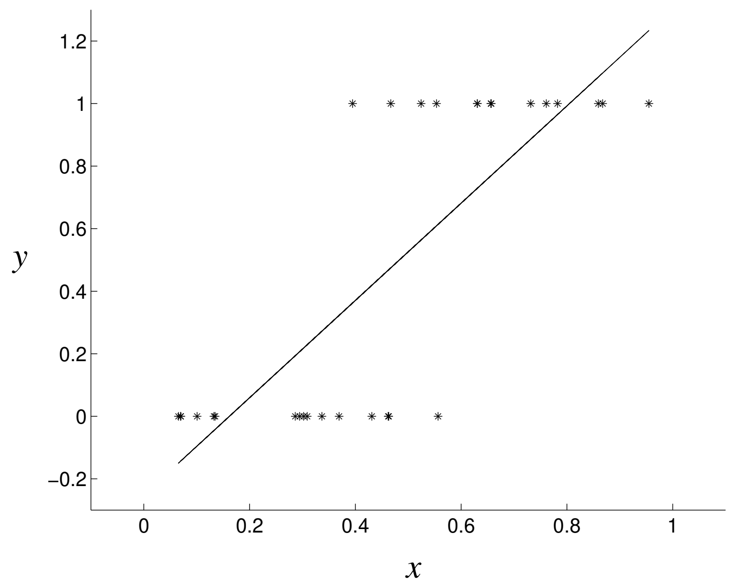

Figure 7.1: Data for a binary classification problem. The abscissa represents the one-dimensional feature vector x, and the ordinate represents the binary class label y, with 0 and 1 representing the two classes. Also shown is the least squares linear regression fit.

makes it difficult to sustain such an interpretation, however, in the setting of binary output data.

Even more serious problems arise when we consider in more detail how the regression fit depends on the data. Suppose in particular that we add the point $(1.5, 1)$ to the data set (see Figure 7.2). The earlier fit (Figure 7.1) yields a fitted value of 2.01 at $x = 1.5$, suggesting, under any reasonable interpretation of this value (e.g., thresholding), that the predicted class label at $x = 1.5$ should be 1. This correctly predicts the class of the new data point, suggesting that the parameters can already accommodate the new data point and need not be changed. Refitting the linear regression, however, changes the slope parameter from 1.55 to 1.23 and the intercept parameter from -0.32 to -0.17 (see Figure 7.2). Moreover, taking the value at which the fit equals 0.5 as the boundary between the two classes, this boundary changes significantly after the introduction of the new data point, leading to changes in the classification of some of the points near the boundary. If we add four additional data points at $x = 1.5$ the boundary moves even further, as shown in Figure 7.2. Given that these new data points are predicted correctly by the original fit, and are far from the boundary, this behavior is disconcerting.

The assumptions underlying linear regression are clearly not met in the classification setting; in particular, the assumption that the variable Y is Gaussian is clearly false. This mismatch between the assumptions and the data is responsible for the problems that we have identified. Once we have made probabilistic assumptions that are appropriate for the classification setting—in particular once we have discarded the Gaussian assumption—we will obtain classification models in which the fitted values behave in an intuitively reasonable manner.

As in Chapter 6 we focus on linear models throughout the current chapter. The notion of

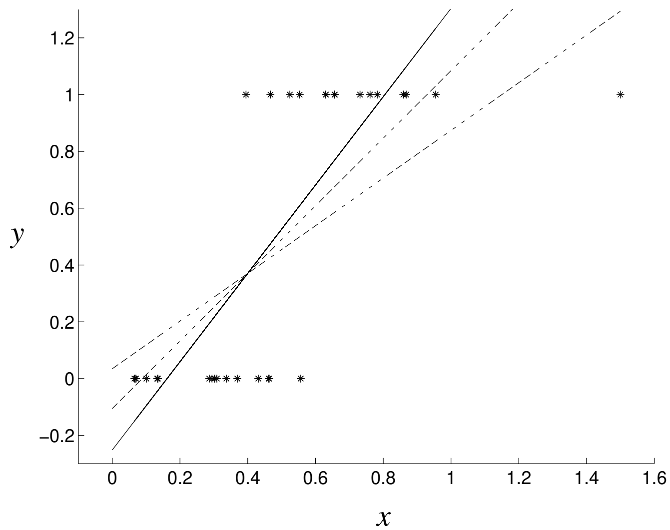

Figure 7.2: Three least squares regression fits. The solid line is the same fit as shown in Figure 7.1, the dash-dot line is the fit to the data with one additional point at (1.5,1), and the dashed line is the fit to the data with five additional points at (1.5,1).

"linearity" in the current chapter is, however, different from that in Chapter 6. We postpone the mathematical details until later sections, where in fact we will find that different classification models invoke the linearity assumption in somewhat different ways. All of the classification models that we study, however, can be viewed as providing a partitioning of the feature space into regions corresponding to the class labels. For linear models the boundaries between these regions are hyperplanes (see Figure 7.3).

7.2 Generative models

Figure 7.4 presents three graphical representations of generative classification models. In all three cases the class label node Y is the parent of the feature vector $X = (X_1, X_2, \ldots, X_m)$. In Figure 7.4(a), the component features are treated as separate nodes; in this case, the children X_j are assumed to be conditionally independent given Y, as confirmed by the d-separation properties of the graph. This is a simplifying assumption that provides a starting point for our presentation and will be our focus through most of this section. In Figure 7.4(b), we have an alternative model in which the components of the feature vector are interdependent, with specific conditional independencies assumed to hold among specific sets of features. In this model general graphical model machinery must be invoked both to parameterize the class-conditional densities and to learn the values of the parameters. Accordingly we will not treat this model explicitly in this section but will return to it in later chapters once the appropriate machinery is in place. Finally, in Figure 7.4(c), we have a model in which no specific conditional independencies are assumed among the components of the

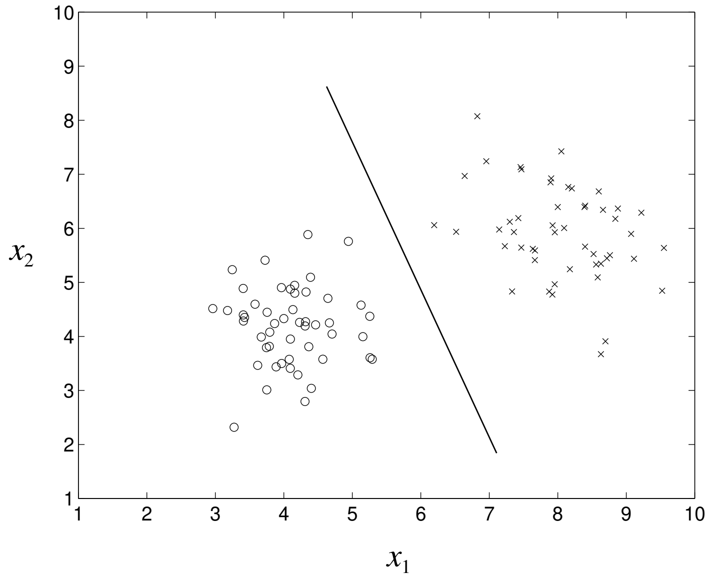

Figure 7.3: A binary classification problem in a two-dimensional feature space. The feature vectors in the training set are plotted as x's and o's for the two classes. Based on the training set, a classifier partitions the feature space into decision regions, one region for each class. In the case of a *linear classifier*, the boundaries between these regions are hyperplanes.

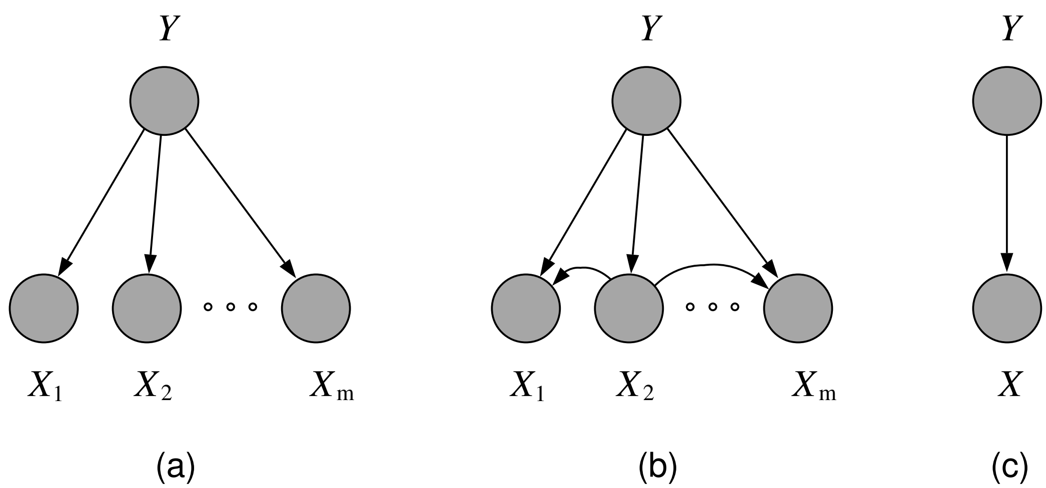

Figure 7.4: Three examples of generative classification models: (a) the case of conditionally independent features, (b) the case of dependent features with some conditional independence assumptions, and (c) the case of no conditional independence assumptions.

feature vector. In this case we represent the feature vector as a single node. This model, despite its simple graphical appearance, is the most general model of the three. We will discuss an example of this model in this section, where a Gaussian assumption for the class-conditional densities will allow us to obtain a simple model despite the absence of conditional independencies.

In all of the examples that we discuss, our goal is twofold: to describe the parametric representation of the posterior probability $p(y \mid x)$ for particular models, and to present maximum likelihood methods for estimating the parameters of the model from data.

7.2.1 Gaussian class-conditional densities

We begin by discussing the model in Figure 7.4(a) in the setting in which the features are continuous and endowed with Gaussian distributions. We initially treat the case of *binary classification*, in which the class label Y can take on one of two values. The extension to multiple classes is discussed in Section 7.2.1.

The model in Figure 7.4(a) requires a marginal probability for Y and a conditional probability for X given Y. Let $Y \in \{0, 1\}$ be a Bernoulli random variable with parameter π:

$$p(y \mid \pi) = \pi^y (1 - \pi)^{1-y}. \tag{7.1}$$

Given the conditional independence assumption expressed by the graph, the probability $p(x \mid y)$ factors into a product over conditional probabilities $p(x_j \mid y)$. For $Y = 0$, let each X_j have a Gaussian distribution:

$$p(x_j \mid Y = 0, \theta_j) = \frac{1}{(2\pi\sigma_j^2)^{1/2}} \exp\left\{ -\frac{1}{2\sigma_j^2}(x_j - \mu_{0j})^2 \right\}, \tag{7.2}$$

where μ_{0j} is the jth component of the mean vector for class $Y = 0$. For $Y = 1$ we have:

$$p(x_j \mid Y = 1, \theta_j) = \frac{1}{(2\pi\sigma_j^2)^{1/2}} \exp\left\{ -\frac{1}{2\sigma_j^2}(x_j - \mu_{1j})^2 \right\}. \tag{7.3}$$

Note that we use θ_j to denote all of the parameters for feature component x_j, including the means μ_{0j} and μ_{1j}, and the variance σ_j^2. Note also that the variances σ_j^2 are allowed to vary across feature components x_j, but are assumed to be constant between the two classes.

Figure 7.5(a) presents an example of a contour plot of two Gaussians in a two-dimensional feature space for the case in which $\sigma_0^2 = \sigma_1^2$. An example in which the variances are unequal is shown in Figure 7.5(b).

The joint probability associated with the graph in Figure 7.4(a) is as follows:

$$p(x, y \mid \theta) = p(y \mid \pi) \prod_{j=1}^{m} p(x_j \mid y, \theta_j), \tag{7.4}$$

where $\theta = (\pi, \theta_1, \theta_2, \ldots, \theta_m)$.

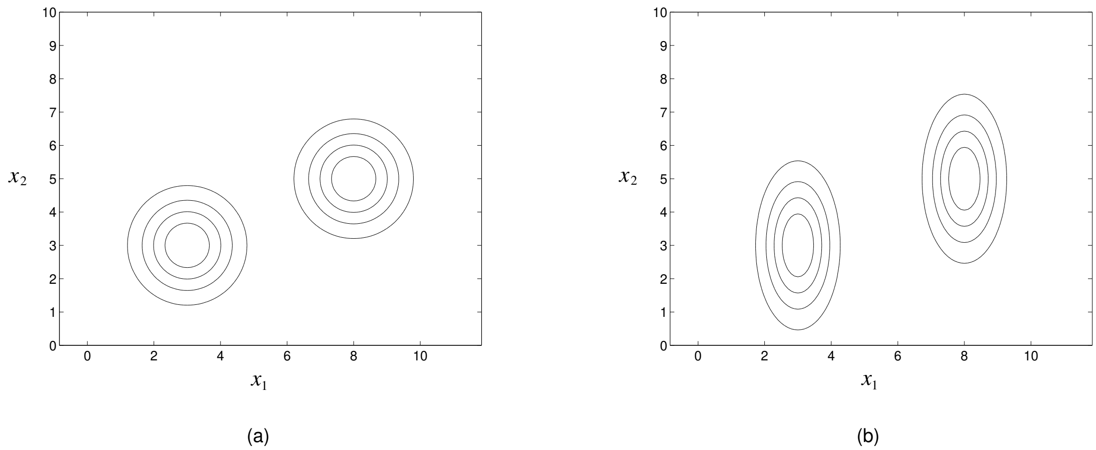

Figure 7.5: (a) A contour plot of Gaussian class-conditional densities for $\sigma_1 = 1$ and $\sigma_2 = 1$. (b) A contour plot for Gaussian class-conditional densities when $\sigma_1 = 0.5$ and $\sigma_2 = 2.0$.

Posterior probability

Let us calculate the posterior probability $p(Y = 1 \mid x, \theta)$. The algebra is somewhat simplified if we work with matrix notation. Thus let:

$$p(x \mid y = k, \theta) = \frac{1}{(2\pi)^{1/2}|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x - \mu_k)^T \Sigma^{-1} (x - \mu_k) \right\}, \tag{7.5}$$

for each of the two classes $k \in \{0, 1\}$, where $\mu_k \triangleq (\mu_{k1}, \mu_{k2}, \ldots, \mu_{km})^T$ is the vector of means for the kth Gaussian, and where $\Sigma \triangleq \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2)$ is a diagonal covariance matrix. We have:

$$\begin{aligned}
p(Y = 1 \mid x, \theta) &= \frac{p(x \mid Y = 1, \theta) p(Y = 1 \mid \pi)}{p(x \mid Y = 1, \theta) p(Y = 1 \mid \pi) + p(x \mid Y = 0, \theta) p(Y = 0 \mid \pi)} \\
&= \frac{\pi \exp\{-\frac{1}{2}(x - \mu_1)^T \Sigma^{-1} (x - \mu_1)\}}{\pi \exp\{-\frac{1}{2}(x - \mu_1)^T \Sigma^{-1} (x - \mu_1)\} + (1 - \pi) \exp\{-\frac{1}{2}(x - \mu_0)^T \Sigma^{-1} (x - \mu_0)\}} \\
&= \frac{1}{1 + \exp\{-\log \frac{\pi}{1-\pi} + \frac{1}{2}(x - \mu_1)^T \Sigma^{-1} (x - \mu_1) - \frac{1}{2}(x - \mu_0)^T \Sigma^{-1} (x - \mu_0)\}} \\
&= \frac{1}{1 + \exp\{-(\mu_1 - \mu_0)^T \Sigma^{-1} x + \frac{1}{2}(\mu_1 - \mu_0)^T \Sigma^{-1} (\mu_1 + \mu_0) - \log \frac{\pi}{1-\pi}\}} \qquad (7.6) \\
&= \frac{1}{1 + \exp\{-\beta^T x - \gamma\}} \qquad (7.7)
\end{aligned}$$

where the final equation defines parameters β and γ:

$$\beta \triangleq \Sigma^{-1}(\mu_1 - \mu_0) \qquad \gamma \triangleq -\frac{1}{2}(\mu_1 - \mu_0)^T \Sigma^{-1} (\mu_1 + \mu_0) + \log \frac{\pi}{1 - \pi}. \tag{7.8}$$

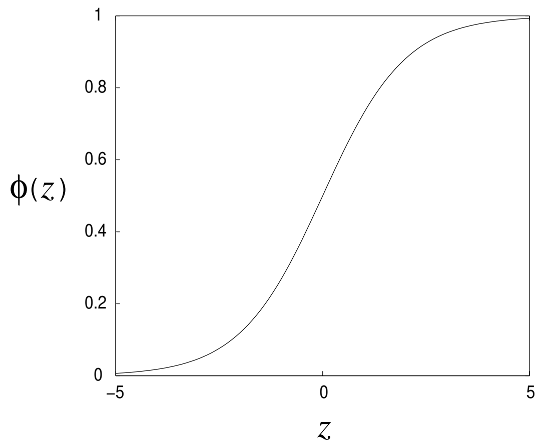

Figure 7.6: A plot of the logistic function.

We see that the posterior probability that $Y = 1$ takes the form:

$$\phi(z) \triangleq \frac{1}{1+e^{-z}}, \tag{7.9}$$

where $z = \beta^T x + \gamma$ is an affine function of x. The function $\phi(z)$ is a smooth, sigmoid-shaped function known as the *logistic function* (see Figure 7.6).

The fact that the feature vector x enters into the posterior probability via an affine function has an important geometric interpretation; in particular, this implies that the contours of equal posterior probability are lines in the feature space. That is, the term $\beta^T x$ is proportional to the projection of x on β, and this projection is equal for all vectors x that lie along a line orthogonal to β. Consider in particular the case in which the variances σ_j^2 are equal to one; thus let $\Sigma = I$. In this case β is equal to $\mu_1 - \mu_0$, and the contours of equal posterior probability are lines that are orthogonal to the difference vector between the means of the two classes (see Figure 7.7(a)).

We obtain equal values of posterior probability for the two classes when $z = 0$ (because the logistic function in Eq. (7.9) evaluates to 0.5 when $z = 0$). To interpret this result geometrically, consider first the case in which the prior probabilities π and $1 - \pi$ are equal. In this case the term $\log(\pi/(1-\pi))$ vanishes and we can rewrite z as follows:

$$z = (\mu_1 - \mu_0)^T \left(x - \frac{(\mu_1 + \mu_0)}{2} \right). \tag{7.10}$$

This is equal to zero for vectors x whose projection on $(\mu_1 - \mu_0)$ is equal to the arithmetic average of the two class means. Thus the posterior probabilities for the two classes are equal when x is equidistant from the two means. This corresponds to the solid line in Figure 7.7(a).

The prior probability π enters via the *log odds ratio* $\log(\pi/(1-\pi))$. This effect of this term can be interpreted as a shift along the abscissa in Figure 7.6. For values of π larger than 0.5 we obtain a shift to the left, which, for a given point in the feature space, corresponds to a larger value of the

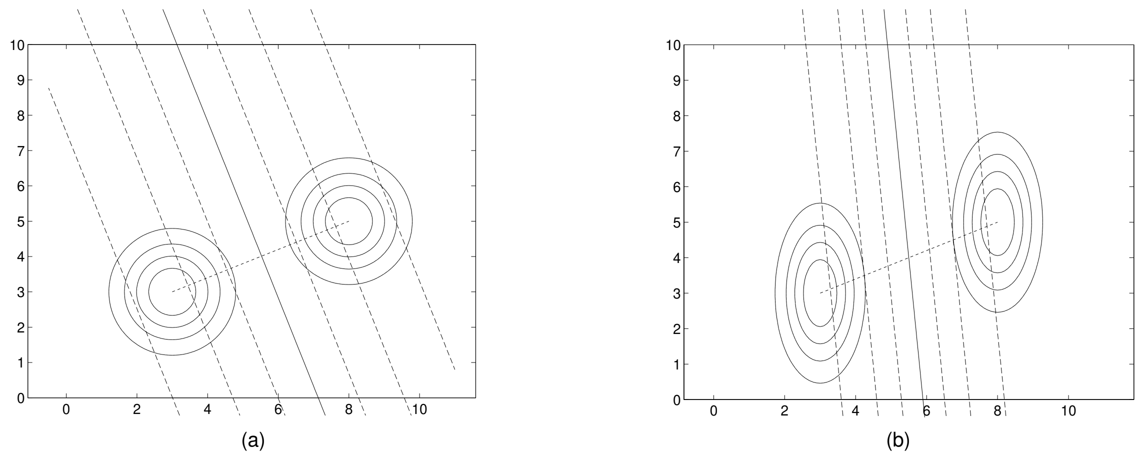

Figure 7.7: (a) The dashed lines and the solid line are contours of equal posterior probability. Note that they are orthogonal to the dotted line connecting the two mean vectors. (b) When $\sigma_1 \neq \sigma_2$, the contours of equal posterior probability are still lines, but they are no longer orthogonal to the difference between the mean vectors.

posterior for class $Y = 1$ (see Figure 7.8(a)). We obtain a shift to the right for π smaller than 0.5 (see Figure 7.8(b)).

Finally, let us consider the case of a general matrix Σ. The contours of equal posterior probability are still lines in the feature space, but in general these lines are no longer orthogonal to the difference vector between the means. If we define new features w via the equation $w \triangleq \Sigma^{-1}x$, however, we obtain the orthogonal geometry of Figure 7.7(a) in the w feature space, which implies an affine geometry in the original feature space. Figure 7.7(b) is an example of this case. Note that the set of vectors that have equal posterior probability for the two classes—the solid line in the figure—are no longer equidistant from the two class means.[1]

As in Chapter 6 it is common to suppress the difference between linear and affine functions to simplify our notation. Thus we augment the vector x to include a first component that is equal to 1, and define the augmented parameter vector $\theta \triangleq (\gamma - \log(\pi/(1-\pi)), \beta^T)^T$. Using this notation, we can summarize the results of this section as follows: for Gaussian class-conditional densities, the posterior probability takes the form:

$$p(Y = 1 \mid x, \theta) = \frac{1}{1 + e^{-\theta^T x}} \tag{7.11}$$

where the parameter vector θ is a function of the means μ_k, the covariance matrix Σ, and the prior probability π.

In summary, we have found that the posterior probability for Gaussian class-conditional densities is the logistic function of a linear function of a feature vector x. We thus have obtained a

[1]We can redefine the distance metric, however, basing it on the matrix Σ^{-1}. In this case the points on the solid line *are* equidistant from the class means. This metric is known as *Mahalanobis distance*; see Exercise **??** for more details.

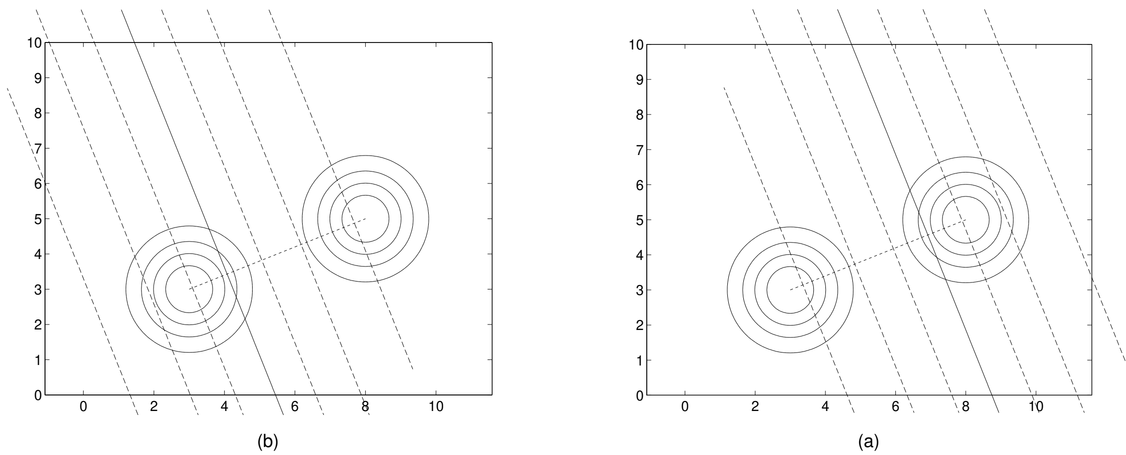

Figure 7.8: The class $Y = 1$ is the upper rightmost of the two Gaussians. (a) When the prior π is greater than 0.5, the contours are shifted to the left, corresponding to a greater posterior probability of $Y = 1$ for a given point in the feature space. (b) When the prior π is less than 0.5, the contours are shifted to the right.

linear classifier—contours of equal posterior probability are lines in the feature space. Inspecting the derivation that yielded this result, we see that the key assumption is that the covariance matrix is the same in the two classes; this leads to a cancellation of the quadratic $x^T\Sigma^{-1}x$ term in the numerator and denominator of the posterior probability. If we retract this assumption and allow different covariance matrices for the two classes, we still obtain a logistic form for the posterior probability, but the argument to the logistic function is now quadratic in x. The corresponding classifier, which has quadratic contours of equal posterior probability, is referred to as a *quadratic classifier*.

Maximum likelihood estimates

In this section we show how to obtain maximum likelihood parameter estimates based on a training set $\mathcal{D}$ composed of N observations: $\mathcal{D} = \{(x_n, y_n); n = 1, \ldots, N\}$. This problem has a straightforward solution that makes use of our work on density estimation in Chapter 5. Reasoning intuitively, suppose that we split the training data into two subsets, one in which $y_n = 0$ and the other in which $y_n = 1$. To estimate π we calculate the proportion of the data in the subset corresponding to $y_n = 1$; this is the maximum likelihood estimate of π. Moreover, we obtain separate maximum likelihood estimates of the Gaussian parameters for each of the two classes, pooling the estimates of the variances to take account of the fact that σ_j is the same in the two classes. This intuitively-defined solution is in fact the overall maximum likelihood solution, as we now verify.

We first form the log likelihood:

$$l(\theta \,|\, \mathcal{D}) \quad = \quad \log \left\{ \prod_{n=1}^{N} p(y_n \,|\, \pi) \prod_{j=1}^{m} p(x_{j,n} \,|\, y_n, \theta_j) \right\} \tag{7.12}$$

$$= \sum_{n=1}^{N} \log p(y_n \,|\, \pi) + \sum_{n=1}^{N} \sum_{j=1}^{m} \log p(x_{j,n} \,|\, y_n, \theta_j), \tag{7.13}$$

where we see that we obtain two separate terms, one for the marginal distribution of Y and the other for the conditional distribution of X_j given Y. Maximizing with respect to π involves only the former term, and for π we therefore obtain:

$$\begin{aligned} \hat{\pi}_{ML} &= \arg\max_{\pi} \sum_{n=1}^{N} \log p(y_n \,|\, \pi) && (7.14) \\ &= \arg\max_{\pi} \sum_{n=1}^{N} \left\{ y_n \log \pi + (1 - y_n) \log(1 - \pi) \right\}, && (7.15) \end{aligned}$$

where the latter equation uses Eq. (7.1). As we have seen in Chapter 5 (cf. Eq. (5.37)), the solution to this constrained optimization problem is the sample proportion:

$$\hat{\pi}_{ML} = \frac{\sum_{n=1}^{N} y_n}{N}, \tag{7.16}$$

where the numerator $\sum_{n=1}^{N} y_n$ is the count of the number of times that the class $Y = 1$ is observed.

Maximization with respect to the parameters θ_j involves only the second term in Eq. (7.13), which we expand further as:

$$\begin{aligned} & \sum_{n=1}^{N} \sum_{j=1}^{m} \log p(x_{j,n} \,|\, y_n, \theta_j) \\ &= \sum_{n=1}^{N} \sum_{j=1}^{m} \log \left\{ p(x_{j,n} \,|\, y_n = 1, \mu_{j1}, \sigma_j)^{y_n} p(x_{j,n} \,|\, y_n = 0, \mu_{j0}, \sigma_j)^{1-y_n} \right\} && (7.17) \\ &= \sum_{j=1}^{m} \left\{ \sum_{n=1}^{N} y_n \log p(x_{j,n} \,|\, y_n = 1, \mu_{j1}, \sigma_j) + \sum_{n=1}^{N} (1 - y_n) \log p(x_{j,n} \,|\, y_n = 0, \mu_{j0}, \sigma_j) \right\}. && (7.18) \end{aligned}$$

Each term in the brackets depends on only one of the parameter vectors $\theta_j = (\mu_{j0}, \mu_{j1}, \sigma_j)$. Thus the problem decomposes into m separate optimization problems, one for each j.

Let us first consider the estimation of μ_{j1}. Plugging in from Eq. (7.3) for $p(x_{j,n} \,|\, y_n = 1, \mu_{j1}, \sigma_j)$, and dropping constants, we have:

$$\hat{\mu}_{j1,ML} = \arg\max_{\mu_{j1}} \left\{ -\frac{1}{2} \sum_{n=1}^{N} y_n (x_{j,n} - \mu_{1j})^2 \right\}. \tag{7.19}$$

This is a weighted least-squares problem, where the "weights" are the binary values y_n. Taking the derivative and setting to zero, we obtain:

$$\hat{\mu}_{j1,ML} = \frac{\sum_{n=1}^{N} y_n x_{j,n}}{\sum_{n=1}^{N} y_n}. \tag{7.20}$$

Thus the maximum likelihood estimate is the sample average of the values $x_{j,n}$ for those data points in class $Y = 1$. Similarly, for $\hat{\mu}_{j0}$ we obtain:

$$\hat{\mu}_{j0,ML} = \frac{\sum_{n=1}^{N}(1-y_n)x_{j,n}}{\sum_{n=1}^{N}(1-y_n)}, \tag{7.21}$$

which is the average of the $x_{j,n}$ for those data points in class $Y = 0$.

Finally, as we ask the reader to verify in Exercise **??**, maximization with respect to the variance σ_j^2 yields:

$$\hat{\sigma}^2_{j,ML} = \frac{\sum_{n=1}^{N} y_n(x_{j,n}-\hat{\mu}_{j1,ML})^2 + \sum_{n=1}^{N}(1-y_n)(x_{j,n}-\hat{\mu}_{j0,ML})^2}{N}; \tag{7.22}$$

a pooled estimate of the variance.

Multiway classification

In this section we consider the generalization to multiway classification, in which the class label Y can take on one of K values.

Let Y be a multinomial random variable with components Y^k and parameter vector π. By definition we have:

$$\pi_k = p(Y^k = 1 \,|\, \pi). \tag{7.23}$$

For each of the K values of Y, define a Gaussian class-conditional density:

$$p(x \,|\, Y^k = 1, \theta) = \frac{1}{(2\pi)^{m/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\right\}, \tag{7.24}$$

where μ_k is the mean associated with the kth class and Σ is a covariance matrix, assumed constant across the K classes. If Σ is diagonal, then the components of X are conditionally independent given the class label Y and the appropriate graphical model is given by Figure 7.4(a). For general Σ, we represent our model as Figure 7.4(c).

The posterior probability of class k is obtained via Bayes rule:

$$\begin{aligned}
p(Y^k = 1 \,|\, x, \theta) &= \frac{p(x \,|\, Y^k = 1, \theta)p(Y^k = 1 \,|\, \pi)}{\sum_l p(x \,|\, Y^l = 1, \theta p(Y^l = 1 \,|\, \pi))} && (7.25)\\
&= \frac{\pi_k \exp\{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\}}{\sum_l \pi_l \exp\{-\frac{1}{2}(x-\mu_l)^T\Sigma^{-1}(x-\mu_l)\}} && (7.26)\\
&= \frac{\exp\{\mu_k^T\Sigma^{-1}x - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log\pi_k\}}{\sum_l \exp\{\mu_l^T\Sigma^{-1}x - \frac{1}{2}\mu_l^T\Sigma^{-1}\mu_l + \log\pi_l\}}, && (7.27)
\end{aligned}$$

where the cancellation of the quadratic $x^T\Sigma^{-1}x$ terms again leaves us with exponents that are linear in x. Defining parameter vectors β_k:

$$\beta_k \triangleq \begin{bmatrix} -\mu_k^T\Sigma^{-1}\mu_k + \log\pi_k \\ \Sigma^{-1}\mu_k \end{bmatrix} \tag{7.28}$$

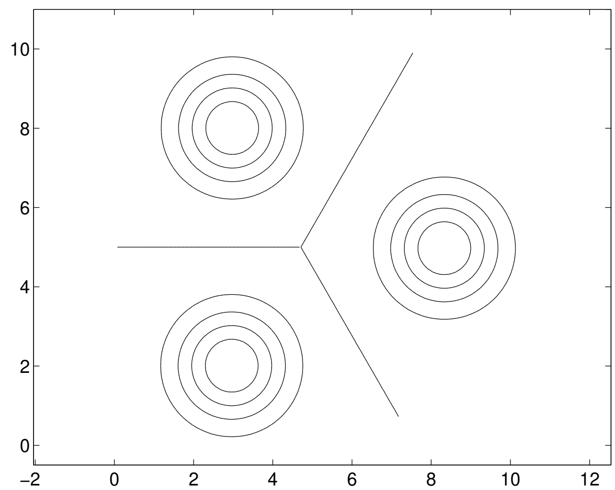

Figure 7.9: Contours of the softmax function. Each line is obtained by setting $\phi_k(z) = \phi_l(z)$ for $k \neq l$. Such a line is a contour of equal posterior probability for classes k and l.

and again simplifying our result by augmenting the vector x to include a first component equal to one, we have:

$$p(Y^k = 1 \,|\, x, \theta) = \frac{e^{\beta_k^T x}}{\sum_l e^{\beta_l^T x}}. \tag{7.29}$$

The function $\phi_k(z) \triangleq e^{z_k} / \sum_l e^{z_l}$ is a smooth function known as the *softmax function.*

The softmax function is a generalization of the logistic function and it has a similar geometric interpretation. Indeed we can transfer much of our earlier work to the multiway setting by considering the ratios of posterior probabilities between pairs of classes. In taking the ratio of $p(Y^k = 1 \,|\, x, \theta)$ and $p(Y^l = 1 \,|\, x, \theta)$, for $k \neq l$, the denominator in the softmax function cancels and we obtain an exponential with exponent $(\beta_k - \beta_l)^T x$. This again involves a projection and thus contours of equal pairwise probability are again lines in the feature space (see Figure 7.9). Moreover, the prior probabilities π again take the form of log odds and act as additive constants in the exponential.

When $\Sigma = \sigma I$, we see from Eq. (7.28) that β_k is proportional to μ_k, and thus the contours of equal probability are again orthogonal to the differences between the class means. For general Σ we obtain the same orthogonal geometry for the transformed coordinates $w \triangleq \Sigma^{-1} x$, which implies an affine geometry for the features x.

The calculation of maximum likelihood estimates for the multiway Gaussian classifier is straightforward and we we ask the reader to carry out the calculation in Exercise **??**. The results can be summarized as follows: We again divide the data into subsets corresponding to the different values of Y. Separate maximum likelihood estimates of the Gaussian parameters are obtained for each class, and the covariance estimates are pooled. Moreover, the maximum likelihood estimates of π are the proportions of data falling into the K classes.

The classifier that we have presented in this section is again a *linear classifier.* The linearity again arises from the Gaussian assumption for the class-conditional densities, together with the assumption of a constant covariance matrix.

7.2.2 The naive Bayes classifier

We now turn to the setting of discrete features, in which each feature X_j can take on one of K values. In this setting the graphical model shown in Figure 7.4(a) is often referred to as the "naive Bayes classifier." We discuss the naive Bayes classifier in this section, calculating the posterior probability and maximum likelihood parameter estimates.

Much of the work in the previous section carries over to the discrete setting. In particular, the joint probability remains the same as before:

$$p(x, y \mid \theta) = p(y \mid \pi) \prod_{j=1}^{m} p(x_j \mid y, \theta_j). \tag{7.30}$$

We again let Y be a multinomial random variable with components Y^k, defining the probability vector π, where:

$$\pi_k \triangleq p(Y^k = 1 \mid \pi). \tag{7.31}$$

Finally, treating the variables X_j as multinomial random variables with components X_j^k, where $X_j^k = 1$ for one and only one value of k, we write the class-conditional densities as follows:

$$p(x_1, x_2, \ldots, x_m \mid Y^i = 1, \eta) = \prod_j \prod_k \eta_{ijk}^{x_j^k}, \tag{7.32}$$

where $\eta_{ijk} \triangleq p(x_j^k = 1 \mid Y^i = 1, \eta)$ is the probability that the jth feature X_j takes on its kth value, for the ith value of the class label Y. Note that the product over k in Eq. (7.32) arises from the definition of multinomial probabilities, and the product over j reflects the assumption that the features are conditionally independent.

Posterior probability

Let us calculate the posterior probability for the naive Bayes classifier. We have:

$$\begin{aligned} p(Y^i = 1 \mid x, \eta) &= \frac{\pi_i \prod_j \prod_k \eta_{ijk}^{x_j^k}}{\sum_l \pi_l \prod_j \prod_k \eta_{ljk}^{x_j^k}} && (7.33) \\ &= \frac{\exp\{\log \pi_i + \sum_j \sum_k x_j^k \log \eta_{ijk}\}}{\sum_l \exp\{\log \pi_l + \sum_j \sum_k x_j^k \log \eta_{ljk}\}}. && (7.34) \end{aligned}$$

As in the Gaussian case, this is again a softmax function of a linear combination of the features. We can express this result in the standardized form:

$$p(Y^i = 1 \mid x, \eta) = \frac{e^{\beta_i^T x}}{\sum_l e^{\beta_l^T x}}, \tag{7.35}$$

with a bit of creativity in the definitions of x and β. In particular, we redefine the vector x by stacking the multinomial vectors x_j vertically. Thus, the components of x are the values x_j^k, where the superscript k varies more rapidly than the subscript j. We also augment the resulting vector to have a first component of one. Similarly, we define β_i as a vector in which the doubly-indexed components $\log \eta_{ijk}$ are arranged, with i fixed and k varying faster than j. We let the the first component of β_i be equal to $\log \pi_i$. Given these definitions we obtain Eq. (7.35) as the posterior probability for the naive Bayes model.

Although the feature space is a discrete hypercube in the naive Bayes setting, it is interesting that the classifier is formally the same as the linear discriminant classifier, with log odds playing the role that difference vectors played in the Gaussian case.

In the case of binary classification, we can divide numerator and denominator by the numerator in Eq. (7.34) and obtain the logistic function of a linear function of the features:

$$p(Y = 1 \mid x, \theta) = \frac{1}{1 + \exp\{-\theta^T x\}} \tag{7.36}$$

for appropriate definitions of θ and x.

Maximum likelihood estimates

Finally, let us calculate the maximum likelihood estimates of the parameters for the naive Bayes classifier. We again assume that we have a training set $\mathcal{D}$ composed of N observations: $\mathcal{D} = \{(x_n, y_n); n = 1, \ldots, N\}$.

From Eq. (7.30) we obtain the log likelihood:

$$l(\theta \mid \mathcal{D}) = \sum_{n=1}^{N} \log p(y_n \mid \pi) + \sum_{n=1}^{N} \sum_{j=1}^{m} \log p(x_{j,n} \mid y_n, \eta), \tag{7.37}$$

where for the purposes of this section we define x and y to be the vectors of all observations $x_{j,n}$ and y_n, respectively. The first term again decouples to yield separate maximum likelihood estimates of π. Focusing on the second term, and recalling that the sum over k of the parameters η_{ijk} must equal one, we introduce Lagrange multipliers λ_{ij} and maximize:

$$\tilde{l}(\eta \mid \mathcal{D}) \triangleq \sum_{n=1}^{N} \sum_{i} \sum_{j} \sum_{k} x_{j,n}^k y_n^i \log \eta_{ijk} + \sum_{i} \sum_{j} \lambda_{ij} (1 - \sum_{k} \eta_{ijk}). \tag{7.38}$$

This yields:

$$\frac{\partial \tilde{l}}{\partial \eta_{ijk}} = \frac{\sum_n x_{j,n}^k y_n^i}{\eta_{ijk}} - \lambda_{ij}. \tag{7.39}$$

Setting to zero and summing both sides with respect to k, we have:

$$\lambda_{ij} = \sum_{k} \sum_{n} x_{j,n}^k y_n^i \tag{7.40}$$

$$= \sum_n \sum_k x_{j,n}^k y_n^i \tag{7.41}$$

$$= \sum_n y_n^i. \tag{7.42}$$

Finally, substituting back into Eq. (7.39), we obtain:

$$\hat{\eta}_{ijk,ML} = \frac{\sum_n x_{j,n}^k y_n^i}{\sum_n y_n^i}, \tag{7.43}$$

in which the numerator is the number of observations in the ith class for which the the jth feature takes on its kth value. The denominator normalizes this count by dividing by the number of observations in the ith class.

7.2.3 The exponential family

For all of the generative classification models studied thus far, the posterior probability takes a simple functional form—a logistic function for the binary problem and a softmax function in the multiway problem. Moreover, for multinomial and Gaussian class-conditional densities (in the latter case with equal, but otherwise arbitrary, class covariance matrices), the contours of equal posterior probability are hyperplanes in the feature space. In fact, as we see in this section, these results are not restricted to multinomial and Gaussian probabilities; but hold for a wide range of class-conditional densities.

The exponential family of probability distributions is a large family that includes the multinomial and Gaussian distributions, as well as a number of other classical distributions such as the binomial, the Poisson, the gamma and the Dirichlet. In Chapter 8 we provide a detailed discussion of the exponential family; here we simply present the functional form of this family, and consider using exponential family distributions as class-conditional densities for classification.

The exponential family is defined as follows:

$$p(x \mid \eta) = \exp\{\eta^T x - a(\eta)\}h(x), \tag{7.44}$$

where η is a parameter vector. It is a useful exercise to verify that the distributions listed above can all be put in this standard form, for appropriate definitions of the functions $a(\eta)$ and $h(x)$. (We will carry out this exercise in Chapter 8).

Let us now consider a binary classification problem for a generic class-conditional density from the exponential family. We assume that the densities for the two classes are the same, up to the parameter vector η. That is, we let the density for class $Y = 1$ be parameterized by η_1 and let the density for class $Y = 0$ be parameterized by η_0. Let the prior probabilities be equal for simplicity. We obtain the posterior probability from Bayes rule:

$$p(Y = 1 \mid x, \eta) = \frac{p(x \mid Y = 1, \eta)p(Y = 1 \mid \pi)}{p(x \mid Y = 1, \eta)p(Y = 1 \mid \pi) + p(x \mid Y = 0, \eta)p(Y = 0 \mid \pi)} \tag{7.45}$$

$$= \frac{\exp\{\eta_1^T x - a(\eta_1)\}h(x)}{\exp\{\eta_1^T x - a(\eta_1)\}h(x) + \exp\{\eta_0^T x - a(\eta_0)\}h(x)} \tag{7.46}$$

$$= \frac{1}{1 + \exp\{-(\eta_0 - \eta_1)^T x - a(\eta_0) + a(\eta_1)\}}. \tag{7.47}$$

Thus we find that the posterior probability is the logistic function of a linear function of x.

Similarly, for the multiway classification problem we have:

$$p(Y^k = 1 \,|\, x, \eta) = \frac{p(x \,|\, Y^k = 1, \eta_k) p(Y^k = 1 \,|\, \pi)}{\sum_l p(x \,|\, Y^l = 1, \eta_l) p(Y^l = 1 \,|\, \pi)} \tag{7.48}$$

$$= \frac{\exp\{\eta_k^T x - a(\eta_k)\} h(x)}{\sum_l \exp\{\eta_l^T x - a(\eta_l)\} h(x)} \tag{7.49}$$

$$= \frac{\exp\{\eta_k^T x - a(\eta_k)\}}{\sum_l \exp\{\eta_l^T x - a(\eta_l)\}}, \tag{7.50}$$

where again we have assumed equal class priors for simplicity. The result is the softmax function of a linear function of x.

7.3 Discriminative models

In Section 7.2.3 we have seen that a wide range of class-conditional densities all yield the same logistic-linear or softmax-linear form for the posterior probability. This invariance of the functional form of the posterior probability to the specific choice of class-conditional density is good news, because in practice it can be difficult to choose the class-conditional density. This problem is particularly difficult in the case of a high-dimensional feature vector. Consider the Gaussian case. The assumption of a diagonal covariance matrix—corresponding to conditional independence of the features—is often unrealistic. We can allow arbitrary covariance matrices, but this requires us to estimate $O(m^2)$ parameters, which may be prohibitive for large m. Often we would like instead to consider families of covariance matrices that depend on more than m but fewer than m^2 parameters. In some cases there is a natural ordering or grouping of the features (e.g., in the case of time series data or spatial data) that yield natural definitions of such structured covariance matrices. In many other cases, however, there is no obvious way to justify a particular form of structured covariance matrix, and we are left with a choice between the (highly-biased) case of a diagonal covariance matrix and the (highly-variable) case of a full covariance matrix. The fact, however, that all of these choices yield the same linear form for the posterior probability suggests that it may not be necessary to make such a choice. Moreover, the fact that densities other than the Gaussian density yield the same linear classifier suggests that we may not even need to specify the density.

In this section we discuss discriminative models. In discriminative modeling the posterior probability is modeled directly, quite apart from any considerations regarding class-conditional probabilities. Instead of assuming Gaussian or multinomial class-conditional densities and deriving the linearity of the classifier as a consequence, we instead assume linearity at the outset, by assuming that x enters into the model via a linear combination $\theta^T x$. To complete the model, we make an additional assumption regarding the (nonlinear) function that maps from $\theta^T x$ to the posterior

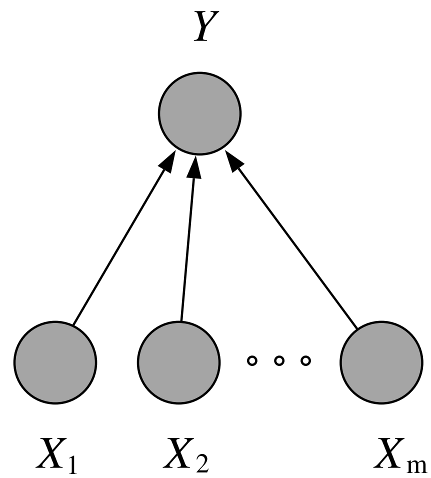

Figure 7.10: The graphical representation of a discriminative classification model.

probability. Taking a hint from the generative setting, we assume a logistic or softmax function at the outset, but we will also explore other possibilities.

The main problem will be that of estimating the parameters of the resulting classifier. Given that we no longer have underlying class-conditional densities, we cannot define the parameters of the classifier in terms of underlying means, covariances, log probabilities or the like. Instead we will have to find a way to estimate the parameters "directly."

The graphical model that we study in this section is shown in Figure 7.10. Note that in this figure we have treated the components of the feature vector X as separate nodes: $X_1, X_2, \ldots, X_m$. We have done this to emphasize the relationship—as well as the contrast—with the discussion of the generative approach in the previous section. Note in particular that we are not assuming nor implying conditional independence of the features. Indeed, in this section we make no assumptions regarding the marginal probability $p(x)$; our goal is only to model the conditional probability $p(y \,|\, x)$. This is of course the same setting as that of regression, and indeed the methods that we discuss in this section are closely related to regression.

7.3.1 Logistic regression

We begin by considering the case of binary classification. The first model that we consider is *logistic regression*, in which the conditional probability $p(y \,|\, x)$ is modeled as a function $\phi(\theta^T x)$, where ϕ is the logistic function. This functional form is of course suggested by the generative models in Section 7.2.

The class label Y is a Bernoulli random variable, and the modeling problem is that of determining the probability that Y takes the value one for each input X. Note that this probability, $p(Y = 1 \,|\, x)$, is the same as the conditional expectation:

$$E(y \,|\, x) \quad = \quad 1 \cdot p(Y = 1 \,|\, x) + 0 \cdot p(Y = 0 \,|\, x) \tag{7.51}$$

$$= \quad p(Y = 1 \mid x). \tag{7.52}$$

Thus, as in the case of regression, the goal is that of modeling the conditional expectation of Y given X. In the regression case, we added a Gaussian error term ϵ to the conditional expectation. This approach, however, is clearly inappropriate here given that Y can only take on the discrete values zero and one. Instead, we define $\mu(x) \triangleq p(Y = 1 \mid x)$ and write the Bernoulli distribution in the following way:

$$p(y \mid x) = \mu(x)^y (1 - \mu(x))^{1-y}. \tag{7.53}$$

This is the usual definition of the Bernoulli distribution; however, we still need to specify the dependence of the Bernoulli parameter $\mu(x)$ on x.

To complete the specification of the model, we assume that (1) the conditional expectation depends on x via the inner product $\eta(x) \triangleq \theta^T x$, where θ is a parameter vector, and (2) the inner product $\eta(x)$ is converted to a probability scale via the logistic function. Thus we have:

$$\mu(x) = \frac{1}{1 + e^{-\eta(x)}}. \tag{7.54}$$

as the probability model for the conditional expectation $\mu(x) \triangleq p(Y = 1 \mid x, \theta)$.

Recall that in the current section we simply treat these assumptions as axiomatic—as an attempt to model posterior probabilities in a simple parametric way independently of assumptions regarding class-conditional densities. Figure 7.11 shows an example that helps to suggest the reasonableness of this approach. In this figure it appears to be difficult to choose a model for the class-conditional densities; in particular, a Gaussian assumption does not seem reasonable. It seems significantly less problematic to choose a discriminative model in this case, and indeed the linear boundary implied by the logistic regression model appears to be reasonable. Such examples are by no means uncommon.

Some properties of the logistic function

In this section we collect together several results regarding the logistic function that will be of use in the following section and in several later chapters.

Let us write the logistic function as a map from a variable η to a variable μ:

$$\mu = \frac{1}{1 + e^{-\eta}} \tag{7.55}$$

The logistic function is invertible; thus we can also obtain a map from μ to η:

$$\eta = \log\left(\frac{\mu}{1 - \mu}\right), \tag{7.56}$$

which has the form of a log odds.

This inverse form simplifies the calculation of derivatives. In particular, we have:

$$\frac{d\eta}{d\mu} = \frac{d}{d\mu} \log\left(\frac{\mu}{1 - \mu}\right) \tag{7.57}$$

$$= \frac{1}{\mu(1 - \mu)}, \tag{7.58}$$

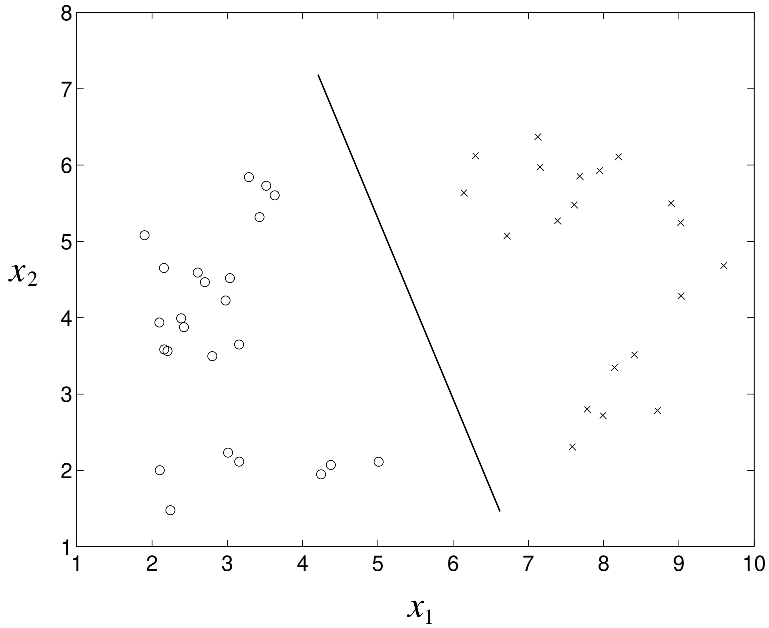

Figure 7.11: An example in which it is difficult to specify the class-conditional densities required for a generative model, but where a linear discriminative boundary between the classes seems reasonable.

from which we obtain:

$$\frac{d\mu}{d\eta} = \mu(1-\mu). \tag{7.59}$$

This expresses the derivative of the logistic function as a function of μ. We can also use Eq. (7.55) to obtain the derivative as a function of η, but the form in Eq. (7.59) will prove to be more useful.

The likelihood

In this section we begin our discussion of maximum likelihood estimation of the parameters θ based on a training set $\mathcal{D} = \{(x_n, y_n); n = 1, \ldots, N\}$. As in our discussion of regression in Chapter 6, we consider batch and on-line methods for parameter estimation.

Let $\eta_n = \theta^T x_n$ and let $\mu_n = 1/(1+e^{-\eta_n})$ denote the corresponding value of the logistic function, in accordance with our definitions in the previous section. Note that we have omitted the explicit dependence of μ_n on x_n to simplify our notation. Moreover, let η and μ denote the vectors of these values as we range across n; thus: $\eta = (\eta_1, \eta_2, \ldots, \eta_N)$ and $\mu = (\mu_1, \mu_2, \ldots, \mu_N)$.

To obtain the likelihood we take the product of N Bernoulli probabilities using Eq. (7.53):

$$p(y_1, \ldots, y_N \,|\, x_1, \ldots, x_N, \theta) = \prod_n \mu_n^{y_n}(1-\mu_n)^{1-y_n}. \tag{7.60}$$

Taking logarithms yields:

$$l(\theta \,|\, \mathcal{D}) = \sum_n \left\{ y_n \log \mu_n + (1 - y_n) \log(1 - \mu_n) \right\}, \tag{7.61}$$

and it is this expression that we must maximize with respect to θ.[2] Recall that μ_n is a function of θ whereas y_n is not.

We calculate the gradient of the log likelihood:

$$\begin{aligned} \nabla_\theta l &= \sum_n \left(\frac{y_n}{\mu_n} - \frac{1 - y_n}{1 - \mu_n} \right) \frac{d\mu_n}{d\eta_n} x_n & (7.62) \\ &= \sum_n \frac{y_n - \mu_n}{\mu_n(1 - \mu_n)} \mu_n (1 - \mu_n) x_n & (7.63) \\ &= \sum_n (y_n - \mu_n) x_n. & (7.64) \end{aligned}$$

It is interesting to note that this gradient has the same form as the gradient of the log likelihood for linear regression (cf. Eq. (6.12)). In both cases we obtain a difference between y_n and the conditional expectation μ_n, multiplied by the input x_n.

An on-line estimation algorithm

An on-line estimation algorithm can be obtained by dropping the summation sign and following the stochastic gradient of the log likelihood. Let $\theta^{(t)}$ denote the value of the parameter vector at the tth step of the algorithm. If (x_n, y_n) denotes the data point presented to the algorithm at the tth step, we write:

$$\theta^{(t+1)} = \theta^{(t)} + \rho(y_n - \mu_n^{(t)}) x_n, \tag{7.65}$$

where $\mu_n^{(t)} \triangleq \phi(\theta^{(t)T} x_n)$ and where ρ is a step size.

Note that this on-line algorithm is identical in form to the LMS algorithm differing only in the definition of the conditional expectation. To understand the (important) implications of the difference, let us return to an issue that motivated our development of classification methods. Recall in particular Figure **??**, where we considered the effect on linear regression of adding the point $(1.5, 1)$ to the training set. The linear fit is altered significantly by the addition of this point. One way to see this is to note that the error, $(y_n - \mu_n^{(t)})$, in the LMS algorithm is large; thus the algorithm will make a large adjustment to the parameter vector $\theta^{(t)}$. For the on-line logistic regression algorithm in Eq. (7.65), however, $\mu_n^{(t)}$ is near one, given that the logistic function is evaluated in its rightmost tail. As suggested in Figure 7.12, the error, $(y_n - \mu_n^{(t)})$, is therefore essentially zero. Thus, as we see from Eq. (7.65), there is little change in the parameters. In general, points that are already classified correctly do not affect the fit.

[2] The function in Eq. (7.61) is the *cross-entropy* function. See Appendix XXX for further discussion of the cross-entropy in the context of information theory.

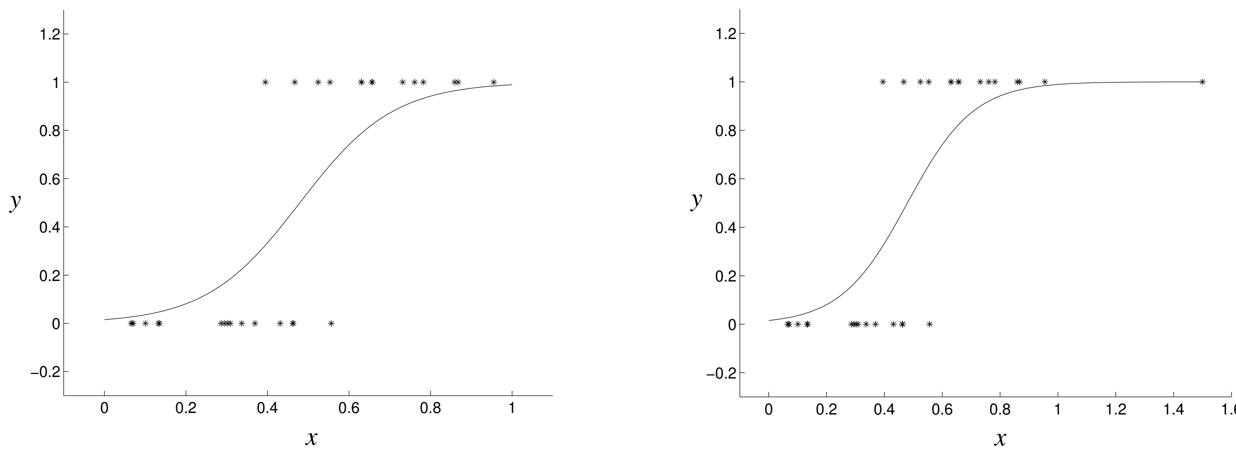

Figure 7.12: (a) The fit of a logistic regression model to the data in Figure 7.1. (b) Adding the point (1.5,1) to the data set does not change the fit (cf. Figure 7.2).

The iteratively reweighted least squares (IRLS) algorithm

To obtain a batch algorithm we could restore the summation sign in Eq. (7.64) and follow the steepest descent direction, but as in the linear regression case this algorithm has little to recommend it. We instead describe an algorithm, known as the *iteratively reweighted least squares (IRLS)* algorithm, that is closer in spirit to the direct solution of the normal equations.

The IRLS algorithm is a Newton-Raphson algorithm.[3] In preparation for deriving the algorithm, let us note that the normal equations can also be viewed, somewhat perversely, from the point of view of the Newton-Raphson algorithm.

Consider a function $J(\theta)$ which is to be minimized with respect to θ. Recall (see Appendix XXX) that the Newton-Raphson algorithm is an iterative algorithm that takes the following general form:

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}\nabla_\theta J, \tag{7.66}$$

where $\nabla_\theta J$ and H are the gradient vector and Hessian matrix of $J(\theta)$ respectively (and both are evaluated at $\theta^{(t)}$).

In the case of linear regression, the cost function, $J = \frac{1}{2}(y - X\theta)^T(y - X\theta)$, is a quadratic function of θ. We calculated the gradient of J in Chapter 6, finding:

$$\nabla_\theta J = -X^T(y - X\theta). \tag{7.67}$$

Taking another derivative we obtain the Hessian:

$$H = -X^T X. \tag{7.68}$$

[3]This statement is not entirely accurate, but it is accurate enough for current purposes. See Chapter 8 for further details.

Thus we can apply the Newton-Raphson algorithm to the problem of minimizing J, obtaining:

$$\theta^{(t+1)} = \theta^{(t)} + (X^T X)^{-1} X^T (y - X\theta^{(t)}) \tag{7.69}$$

$$= (X^T X)^{-1} X^T y, \tag{7.70}$$

where we see that the right-hand-side is the solution to the normal equations. Thus Newton-Raphson hops to the solution in a single step, not a surprise given that J is a quadratic function.

In the logistic regression problem, the function to be optimized is the log likelihood, and this function is not quadratic. Nonetheless it is "nearly" quadratic, and we should not be surprised to see that Newton-Raphson for logistic regression has similarities to the linear regression solution. Indeed, as we will see, the similarity is strong.

The function that we wish to optimize is the log likelihood shown in Eq. (7.61). We have already calculated the gradient of the log likelihood in Eq. (7.64). Writing this result in vector notation, we have:

$$\nabla_\theta l = \sum_n (y_n - \mu_n) x_n = X^T (y - \mu), \tag{7.71}$$

where we have defined $\mu \triangleq (\mu_1, \mu_2, \ldots, \mu_N)^T$. Taking a second derivative, we have:

$$H = -\sum_n \frac{d\mu_n}{d\eta_n} x_n x_n^T \tag{7.72}$$

$$= -\sum_n \mu_n (1 - \mu_n) x_n x_n^T \tag{7.73}$$

$$= -X^T W X, \tag{7.74}$$

where we have defined the diagonal weight matrix:

$$W \triangleq \mathrm{diag}\{\mu_1(1-\mu_1), \mu_2(1-\mu_2), \ldots, \mu_N(1-\mu_N)\}, \tag{7.75}$$

Note that the μ_n values depend on the parameter vector θ, thus the weight matrix W depends on θ. We thus will use the notation $W^{(t)}$ to denote the weight matrix at the tth iteration of the algorithm.

Substituting into Eq. (7.66), we obtain:

$$\theta^{(t+1)} = \theta^{(t)} + (X^T W^{(t)} X)^{-1} X^T (y - \mu^{(t)}) \tag{7.76}$$

$$= (X^T W^{(t)} X)^{-1} \left[X^T W^{(t)} X \theta^{(t)} + X^T (y - \mu^{(t)}) \right] \tag{7.77}$$

$$= (X^T W^{(t)} X)^{-1} X^T W^{(t)} z^{(t)}, \tag{7.78}$$

where we define:

$$z^{(t)} = \eta + [W^{(t)}]^{-1} (y - \mu^{(t)}). \tag{7.79}$$

The algorithm in Eq. (7.78) is the IRLS algorithm.

Inspecting Eq. (7.78) makes it clear why the algorithm is known as the "iteratively reweighted least squares" algorithm. Each iteration of the algorithm involves solving a weighted least-squares problem (recall Eq. (**??**)). Moreover, given that the weight matrix W changes at each iteration, the least-squares problem is "iteratively reweighted."

We can obtain some more insight into the IRLS algorithm, and in particular understand the role played by $z^{(t)}$, if we view the Newton-Raphson algorithm as solving a sequence of linearized problems.

Consider the following (heuristic) argument. For a particular value θ, and a particular vector x_n, let us linearize the logistic function around the "operating point," $\eta_n = \theta^T x_n$. This linearization allows us to convert the value y_n, which is on a nonlinear scale, "backwards" to a value z_n on the linear scale defined by η_n. In paricular, recall that the logistic function can be inverted (cf. Eq. (7.56)) to yield a map from μ_n to η_n. Expanding this inverse function in a first-order Taylor series, we define:

$$z_n \triangleq \eta_n + \frac{d\eta_n}{d\mu_n}(y_n - \mu_n), \tag{7.80}$$

where the derivative is evaluated at η_n, and thus depends implicitly on the parameter vector θ.

This argument suggests using z_n as a surrogate for y_n, in a linearized version of our logistic regression problem. We have another issue to deal with, however, if we wish to use linear regression methods to find parameter estimates: the Bernoulli random variables y_n do not have equal variance. In particular, y_n has variance $\mu_n(1 - \mu_n)$. To deal with this issue, we use weighted least squares. In particular, note that the elements of the weighting matrix W defined in Eq. (7.75) are exactly the Bernoulli variances. Thus we use W as our weight matrix.

We now solve a weighted least squares problem, with data z_n and weight matrix W. Writing the normal equations for this weighted least squares problem, and making the dependence on the iteration number t explicit, we obtain the IRLS iteration in Eq. (7.78).

The Newton-Raphson algorithm is a second-order algorithm and it generally converges rapidly. A small number of iterations of Eq. (7.78) are usually sufficient to obtain convergence of the parameter vector.

7.3.2 Multiway classification

In this section we discuss a generalization of logistic regression to the setting of multiway classification. Recall that in this case the class label Y can take on one of K values.

In Section 7.2.1 we derived the softmax-linear model:

$$p(Y^k = 1 \,|\, x, \theta) = \frac{e^{\theta_k^T x}}{\sum_l e^{\theta_l^T x}} \tag{7.81}$$

as the multiway generalization of the logistic-linear model. In that section, the softmax-linear form for the posterior probability was a consequence of our assumption of Gaussian (or more generally, exponential family) class-conditional probabilities. In the current section, however, we adopt a discriminative perspective in which the softmax-linear form is treated as an assumption, and we make no attempt to specify class-conditional probabilities. In this context, we refer to the model in Eq. (7.81) as a *softmax regression* model. As in the case of logistic regression, the main problem that we face is estimating the parameters θ_k "directly," without making use of an underlying class-conditional model.

We use the notation μ_n^k to denote the posterior probability in Eq. (7.81). We also use $\eta_n^k = \theta_k^T x_n$ to denote the linear component of the softmax-linear model.

Some properties of the softmax function

The softmax function has several properties that are analogs of those of the logistic function that we discussed in Section 7.3.1.

The softmax function can be written as a map from a vector variable η to a vector variable μ. Letting η^i represent the ith component of η and letting μ^i represent the ith component of μ, we write:

$$\mu^i = \frac{e^{\eta^i}}{\sum_k e^{\eta^k}}. \tag{7.82}$$

This function is invertible up to an additive constant. That is, if we add the constant C to each of the components η^i, then the factor e^C cancels in the numerator and denominator of Eq. (7.82), yielding the same value of μ^i. Note in particular that if we take the logarithm of both sides of Eq. (7.82), we obtain the inverse:

$$\eta^i = \log \mu^i + D, \tag{7.83}$$

where $D = \log \sum_k e^{\eta^k}$ is a constant. Any other constant (including zero) will yield an equivalent inverse of the softmax function.

We turn to the calculation of the softmax derivatives. A subtlety in this case is that the derivative of μ_i with respect to η_j is non-zero for $i \neq j$, due to the denominator in Eq. (7.82). The calculation proceeds as follows:

$$\begin{aligned}
\frac{\partial \mu_i}{\partial \eta_j} &= \frac{(\sum_k e^{\eta_k}) e^{\eta_i} \delta_{ij} - e^{\eta_i} e^{\eta_j}}{(\sum_k e^{\eta_k})^2} && (7.84)\\
&= \frac{e^{\eta_i}}{\sum_k e^{\eta_k}} \left(\delta_{ij} - \frac{e^{\eta_j}}{\sum_k e^{\eta_k}} \right) && (7.85)\\
&= \mu_i (\delta_{ij} - \mu_j), && (7.86)
\end{aligned}$$

where δ_{ij} is equal to one if $i = j$ and zero otherwise.

Maximum likelihood estimation

In the multiway classification problem the output Y is a multinomial random variable. Recalling that in softmax regression μ_n^k denotes the posterior probability of the kth class for the nth data point, we can write the multinomial probability distribution in the following form:

$$p(y_n \mid x_n, \theta) = \prod_k \left(\mu_n^k \right)^{y_n^k} \tag{7.87}$$

where $\theta \triangleq (\mu_n^1, \mu_n^2, \ldots, \mu_n^K)^T$ is the multinomial parameter vector. The likelihood is the product of N such probabilities. Taking the logarithm, we obtain:

$$l(\theta \mid \mathcal{D}) = \sum_n \sum_k y_n^k \log \mu_n^k \tag{7.88}$$

as the log likelihood for the multiway classification problem. As in the binary case, this log likelihood has the form of a cross-entropy.

To calculate the gradient of the log likelihood with respect to the parameter vector θ_i, we make use of the intermediate variable $\eta_n^i = \theta_i^T x_n$. Recalling that the derivative of μ_n^k with respect to η_n^i is nonzero because of the shared denominator in the softmax function, we have:

$$\begin{aligned}
\nabla_{\theta_i} l &= \sum_n \sum_k \frac{\partial l}{\partial \mu_n^k} \frac{\partial \mu_n^k}{\partial \eta_n^i} \frac{d\eta_n^i}{d\theta_i} && (7.89)\\
&= \sum_n \sum_k \frac{y_n^k}{\mu_n^k} \mu_n^k (\delta_{ik} - \mu_n^i) x_n && (7.90)\\
&= \sum_n \sum_k y_n^k (\delta_{ik} - \mu_n^i) x_n && (7.91)\\
&= \sum_n (y_n^i - \mu_n^i) x_n, && (7.92)
\end{aligned}$$

where we have used the fact that $\sum_k y_n^k = 1$.

The gradient that we have obtained has the same form as the gradient for logistic regression and linear regression! (Recall Eq. (7.64) and Eq. (6.21)). We will see in Chapter 8 that this result is not a coincidence, but reflects a general property of probability distributions in the exponential family.

As in the case of logistic regression and linear regression, we obtain an on-line parameter estimation algorithm by dropping the sum over n in Eq. (7.92). This algorithm is the analog of the LMS algorithm for multiway classification.

It is straightforward to generalize the IRLS algorithm and thereby obtain a batch algorithm for softmax regression. Rather than pursuing that generalization here, we return to the IRLS algorithm in Chapter 8, where we develop a generic IRLS algorithm for the family of generalized linear models, of which softmax regression and logistic regression are examples.

7.3.3 Probit regression

In this section and the remainder of the chapter, we return to binary classification and consider some alternatives to logistic regression.

Although the logistic regression model arises naturally from a generative perspective—as the posterior probability obtained from a wide class of class-conditional probabilities—there are other choices of class-conditional probabilities that do not yield the logistic-linear form for the posterior probability. Thus, even from a generative point of view there is some motivation for exploring alternative representations for the posterior probability. In this section we engage in such an exploration within the discriminative framework, motivating alternative models "directly," without reference to class-conditional distributions. For simplicity we retain the linearity assumption, and motivate functions other than the logistic function for converting the linear combination $\theta^T x$ to a probability scale.

One natural way to obtain a discriminative classification model is to consider "noisy threshold" models. In particular, we might suppose that a data pair (x, y) is obtained by a process in which some external agent converts the vector x to a scalar value η, defined as a linear combination $\theta^T x$, and compares the resulting value to a threshold. If the value exceeds the threshold, then the label 1 is assigned, otherwise the label 0 is assigned. A probabilistic version of this model can be obtained by assuming that the threshold is stochastic. Thus, let Z be a scalar random variable with a cumulative distribution function $F(z)$. We define:

$$p(Y = 1 \mid x) = p(Z \leq \eta) = F(\eta). \tag{7.93}$$

Making the further assumption that η is parameterized linearly, as $\eta \triangleq \theta^T x$, we obtain a discriminative classification model:

$$p(Y = 1 \mid x, \theta) = F(\theta^T x), \tag{7.94}$$

for a given distribution function F.

The logistic regression model can be interpreted as a special case of this model, given that the logistic function, $1/(1 + \exp(-x))$, is a distribution function. There is no particular reason to use a logistic random variable as the noisy threshold model, however. Indeed, given that many natural sources of "noise" have a Gaussian distribution, a common choice is to take Z to be a Gaussian random variable. This choice yields the *probit regression model*. Thus, in the probit model we have:

$$p(Y = 1 \mid x, \theta) = \Phi(\theta^T x), \tag{7.95}$$

where

$$\Phi(w) = \int_{-\infty}^{w} \frac{1}{(2\pi)^{1/2}} e^{-\frac{1}{2}\xi^2} d\xi \tag{7.96}$$

is the cumulative distribution function of a Gaussian random variable with zero mean and unit variance.[4]

Figure 7.13 shows a graphical model representation of the probit regression model. In this representation, the threshold variable Z is represented as an explicit latent variable. The graphical model requires a marginal distribution for Z, which in the probit model we take as $\mathcal{N}(0, 1)$, and a conditional distribution for Y, given X and Z. This conditional is a degenerate distribution: Y is equal to zero if $\theta^T x$ is less than Z, and one otherwise.

Figure 7.14 shows a plot of the logistic function and the Gaussian cumulative distribution function. As this plot makes clear, there is not a large difference between the two functions, and indeed probit regression and logistic regression generally give rather similar results.

Probit regression is an instance of the family of generalized linear models that we describe in Chapter 8. Maximum likelihood estimates can be obtained via stochastic gradient descent or the general version of the IRLS algorithm that we present in that chapter.

[4] The assumption of zero mean and unit variance is without loss of generality, because any linear transformation of the features can be absorbed in the parameter vector θ.

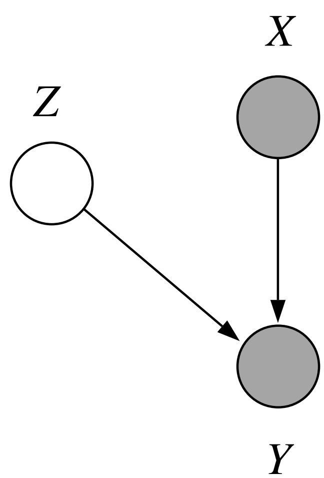

Figure 7.13: A graphical model representation of the probit regression model.

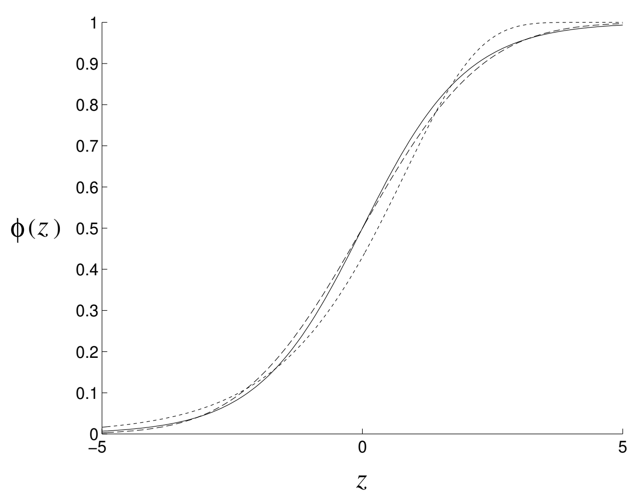

Figure 7.14: Link functions for binary classification. The solid curve is the logistic function (Eq. 7.55), the long-dashed curve is the cumulative Gaussian function (Eq. 7.96), and the small-dashed curve is the complementary log-log function (the inverse of Eq. 7.105).

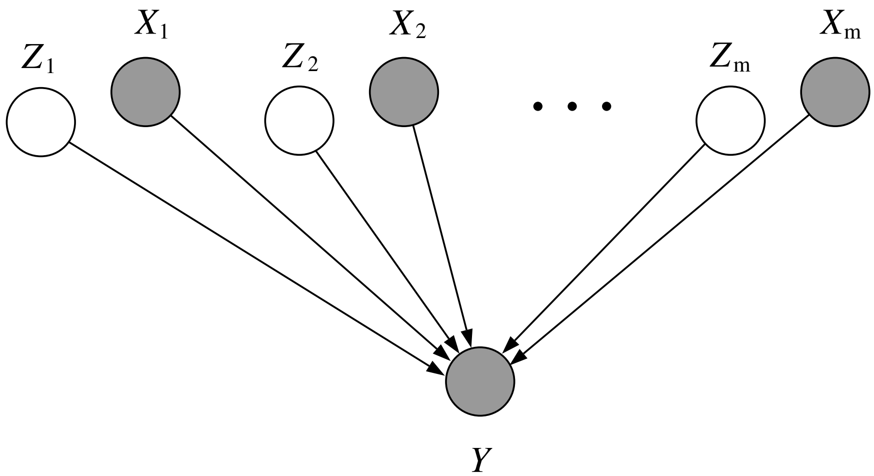

Figure 7.15: A graphical model representation of the noisy-OR model.

7.3.4 The noisy-OR model

A wide range of models can be obtained as "noisy" versions of formulas from propositional logic, in the setting in which the features X_i are binary. In this section we describe an example of this class of models known as the *noisy-OR* model. As with the other models discussed in this chapter the noisy-OR model is a linear classifier.

Let us begin with the Boolean formula:

$$Y = X_1 \vee X_2 \vee \cdots \vee X_m, \tag{7.97}$$

where $X_i \in \{0, 1\}$, for all i. To obtain a "noisy" version of the formula, let us view each variable X_i as encoding a binary "trigger" that can "cause" Y to occur. Eq. (7.97) states that the presence of any single trigger suffices to cause Y to occur. Suppose now that each trigger can "fail" with some probability ξ_i, in that the trigger can be present, but can fail to cause the occurrence of Y. Suppose moreover that the failure probabilities associated with the different triggers are independent. Thus, introducing independent binary random variables Z_i to represent the failure events, we have:

$$Y = \begin{cases} 1 & (X_1 \wedge \neg Z_1) \vee (X_2 \wedge \neg Z_2) \cdots \vee (X_m \wedge \neg Z_m) \\ 0 & \text{otherwise.} \end{cases} \tag{7.98}$$

The graphical model representing this noisy version of the logical OR formula is shown in Figure 7.15.

If we let $\xi_i \triangleq p(z_i = 1)$ denote the Bernoulli parameters associated with the Z_i, we obtain from Eq. (7.98):

$$p(Y = 0 \,|\, x, \xi) = \prod_{i=1}^{m} \{p(z_i = 1)\}^{x_i} = \prod_{i=1}^{m} \xi_i^{x_i}. \tag{7.99}$$

This formula can be interpreted as stating that the probability of Y *not* occurring is the product of the (independent) failure probabilities associated with those features x_i that are present in the input. That is, if all triggers fail, then Y doesn't occur.

To express the noisy-OR model in a linear form, let us rewrite Eq. (7.99):

$$p(Y = 0 \mid x, \xi) = \exp\left\{\sum_{i=1}^{m} x_i \log \xi_i\right\}. \tag{7.100}$$

Letting $\theta_i \triangleq -\log \xi_i$, we obtain our final result:

$$p(Y = 1 \mid x, \theta) = 1 - e^{-\theta^T x} \tag{7.101}$$

for the posterior probability for the noisy-OR model.

7.3.5 Other exponential models

A number of useful classification models are based on the Poisson distribution. Recall that Z is a Poisson random variable with parameter λ if:

$$p(z \mid \lambda) = \frac{\lambda^z e^{-\lambda}}{z!}, \tag{7.102}$$

where z ranges over the nonnegative integers. Poisson variables arise in many contexts, in particular as models of counts of rarely occurring, independent events. For example, in a well-stirred solution that contains a small amount of a virus, the amount of virus in any sample might be a Poisson variable with parameter proportional to the volume of the sample. In such situations, it is often of interest to distinguish between the case in which Z takes on the value zero and the case in which Z takes on a non-zero value. (For example, a model of transmission of viral disease would want to distinguish the case that a sample of the solution contained no viral cells). Defining a binary variable Y that is equal to one in the latter case, we have:

$$p(Y = 1) = 1 - p(Z = 0) = 1 - e^{-\lambda}, \tag{7.103}$$

from Eq. (7.102). If we treat the parameter λ as a linear function of a set of input variables x, we obtain a classification model:

$$p(Y = 1 \mid x, \theta) = 1 - e^{-\theta^T x}. \tag{7.104}$$

This model is identical in form to the noisy-OR model, although the vector x is no longer restricted to be a binary vector.

An awkward aspect of the model in Eq. (7.104) is that the linear combination $\theta^T x$ must be restricted to lie between zero and infinity if we are to obtain a posterior probability that lies between zero and one. To remove this restriction, it is convenient to reparameterize the model so that the argument λ is the exponential function of some underlying variable η. We obtain a linear classification model if we assume that the underlying variable η is linear in x:

$$p(Y = 1 \mid x, \theta) = 1 - e^{-e^{\theta^T x}}. \tag{7.105}$$

An appealing feature of this model is that there are no longer any restrictions on θ. In fact, in situations involving Poisson variables it is often natural to measure the effect of the variables x on a logarithmic scale. In particular, in the example of the viral solution, x might measure the fraction of some diluting agent in the solution.

The model in Eq. (7.105) is referred to as the *complementary log-log model*. (The terminology refers to the inverse of the nonlinear function in Eq. (7.105)). Figure 7.14 includes a plot of the nonlinearity in this model. Note again the similarity to the logistic function.

7.4 Summary

We have presented a number of simple probabilistic models for discrete variables within the framework of binary and multiway classification problems. We discussed generative models, in which the discrete variable is a parent of the feature variables. We also discussed discriminative models, in which the discrete variable is a child of the feature variables. We also focused on some of the relationships between generative and discriminative models.

Maximum likelihood estimates are readily obtained in both cases. In the case of a generative model, maximum likelihood estimation essentially reduces to density estimation. That is, we find estimates of the class-conditional densities separately for each of the classes. In the discriminative setting, we model the class label as a Bernoulli or multinomial variable, which yields a cross entropy for the log likelihood. The IRLS algorithm can be used to maximize this log likelihood in the batch setting. We also presented a stochastic gradient algorithm for the on-line setting, noting the close relationship to the LMS algorithm.

All of the models that we have presented in this chapter are linear classifiers. That is, in all cases the input variable x enters into the model via a linear combination $\eta = \theta^T x$. In the generative setting this linear form was a consequence of the particular kinds of class-conditional densities that we assumed. In the discriminative setting we assumed the linear form at the outset.

Generative and discriminative models have complementary strengths and weaknesses. The generative approach allows knowledge about class-conditional densities to be exploited. If this knowledge is indeed reflective of the true data-generation process, then the generative approach can be more *efficient* than a corresponding discriminative model, in the sense that it will tend to require fewer data points. On the other hand, discriminative approaches tend to be more *robust* than generative approaches, making use of weaker assumptions regarding class-conditional densities. Note also that the discriminative framework presents a straightforward “upgrade path” toward the development of nonlinear classifiers—we can retain the logistic and softmax functions, but replace the linear combination $\eta = \theta^T x$ with a nonlinear function (see Chapter 25).

7.5 Historical remarks and bibliography