CS281A/Stat241A: Statistical Learning Theory

Binary Classification (9/30/04)

Lecturer: Michael I. Jordan *Scribes: Leon Barrett and Benjamin I. P. Rubinstein*

1 Introduction

Recall our setup for *binary classification* where we are given (a realization of) a *data set* $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{N}$, where the X_i are arbitrary *features* (often vectors in $\mathbb{R}^n$) and the $Y_i \in \{0, 1\}$ are corresponding binary-valued *class labels*. A typical plot of $\mathcal{X}$ is shown in Figure 1.

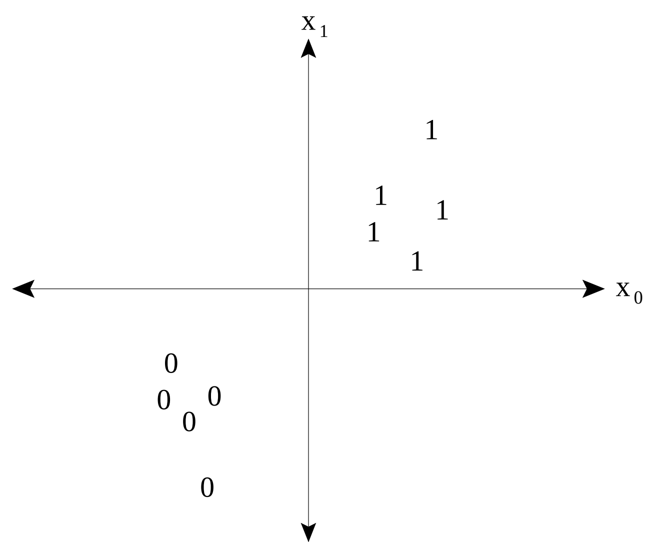

Figure 1: A typical X-space plot of a data set $\mathcal{X}$ with each example $(x_i, y_i) \in \mathcal{X}$ located at its X component and labeled by its Y component.

The goal of binary classification is to find a *decision function* whose level sets may be used to separate the X_i of differing class labels Y_i. This should be viewed probabilistically as computing the posterior $p(y \mid x)$ and performing a subsequent operation like argmax (e.g. classifying points x by $\text{argmax}_{y \in \{0,1\}} p(y \mid x)$). One widely studied binary classification problem is in *intrusion detection*, where each X_i is a set of network packets and the corresponding Y_i denotes whether an intrusion has occurred or not. This lecture will focus on two important models of inference for binary classification.

The *generative model* of inference assumes that X is dependent on Y as in Figure 2. Under this model, calculation of the posterior $p(y \mid x)$ requires an application of Bayes' Rule and access to the prior $p(y)$ (a single $[0, 1]$-valued parameter for binary classification) and the likelihood $p(x \mid y)$ for both classes (i.e. $p(x \mid y = 0)$ and $p(x \mid y = 1)$). This latter quantity is known as the *class-conditional density* of the features on the class and requires that we make assumptions on the form of X. An instance of the generative model of binary classification, which we will work with in the sequel, is the case in which each class-conditional

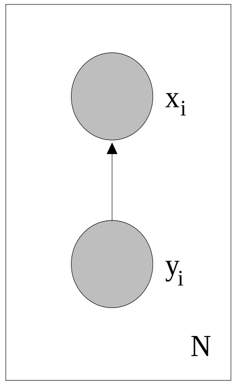

Figure 2: A graphical model representing the generative model of classification/regression on an i.i.d. sample.

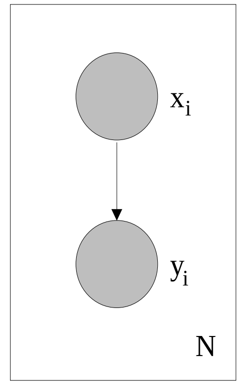

Figure 3: A graphical model representing the discriminative model of classification/regression on an i.i.d. sample.

density is a Gaussian. This model is indexed by up to five parameters: the prior $p(y)$ and the mean vector and covariance matrix for each class-conditional density.

In contrast the *discriminative model* of inference assumes that the classes Y are dependent on the features X as shown in Figure 3. Unlike generative inference, this model does not require the class-conditional densities $p(x \mid y)$ and so we do not need to model X – here we deal with the posterior $p(y \mid x)$ directly and are thus essentially interested in finding a separating boundary between the two classes based on the data set $\mathcal{X}$. An important instance of the discriminative model of binary classification, which we will work with in the sequel, is *logistic regression* in which the posterior is taken to be the logistic function $p(y \mid x) = \frac{1}{1+e^{-\theta^T x}}$ and predictions are made by thresholding at 1/2. We will see that this function is closely related to the Gaussian class-conditional density used in generative inference.

2 Generative Binary Classification with Gaussian Class-conditional Densities

2.1 Gaussian C.C.D.'s – Class-independent Covariance Matrix

Let us consider further the generative example having Gaussian class-conditional densities. As mentioned above, this model can have up to 5 parameters for binary classification. For simplicity we initially assume that the covariance matrices are identical for each class so that our model is parameterized by the prior $\pi = p(Y = 1)$, the means $\mu_1 = E[x \mid Y = 1]$ and $\mu_0 = E[x \mid Y = 0]$, and the (shared) $d \times d$ covariance matrix Σ, where d is the dimension of the feature vectors $X \in \mathbb{R}^d$. With these parameters in hand, the class-conditional densities are:

$$p(x \mid Y = 1) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)} \quad (1)$$

$$p(x \mid Y = 0) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)} \quad (2)$$

Be careful not to confuse the constant $\pi = 3.141...$ in Equations 1 and 2 with the parameter $\pi = p(Y = 0)$ – the latter is the prior and does not appear in the likelihoods $p(x \mid y)$. We wish to 'invert the arrow' in

Figure 2 in order to compute the posterior $p(y \mid x)$. Noting that it is sufficient to compute $p(Y=1 \mid x)$ as $p(Y=0 \mid x) = 1 - p(Y=1 \mid x)$, this is accomplished via Bayes' Rule:

$$\begin{aligned} p(Y=1 \mid x) &= \frac{p(x \mid Y=1)p(Y=1)}{p(x)} \\ &= \frac{p(x \mid Y=1)p(Y=1)}{p(x \mid Y=1)p(Y=1) + p(x \mid Y=0)p(Y=0)} \\ &= \frac{p(x \mid Y=1)\pi}{p(x \mid Y=1)\pi + p(x \mid Y=0)(1-\pi)} \end{aligned}$$

Upon dividing both halves of the fraction by the numerator this becomes

$$p(Y=1 \mid x) = \frac{1}{1 + \frac{p(x|Y=0)}{p(x|Y=1)}\frac{1-\pi}{\pi}}$$

Note that this functional form is starting to look like that of the logistic function, $f(x) = \frac{1}{1+e^{-\theta^T x}}$. Taking $e^{\log(\cdot)}$ of the denominator's second term we reach the logistic function

$$p(Y=1 \mid x) = \frac{1}{1 + \exp\left(\log(\frac{p(x|y=0)}{p(x|y=1)}) + \log(\frac{1-\pi}{\pi})\right)}$$

Can the exponent be simplified with the function being made to be linear in x? In particular, we'd like the function to be of the form $\frac{1}{1+e^{-\theta^T x}}$. It turns out that the Gaussian class-conditional densities will do this for us – the leading terms of the two Gaussians cancel, and the exponents combine, giving

$$\begin{aligned} p(Y=1 \mid x) &= \frac{1}{1 + e^{-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) + \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) + \log\frac{1-\pi}{\pi}}} && (3) \\ &= \frac{1}{1 + e^{-(\mu_1-\mu_0)^T\Sigma^{-1}x - \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 + \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \log\frac{1-\pi}{\pi}}} && (4) \\ &= \frac{1}{1 + e^{-(\mu_1-\mu_0)^T\Sigma^{-1}\left(x - \frac{\mu_0+\mu_1}{2}\right) + \log\frac{1-\pi}{\pi}}} && (5) \end{aligned}$$

where (4) results from canceling the quadratic $x^T x$ terms in (3) – which relies on the fact that each class-conditional density shares the same covariance matrix Σ – and (5) comes from factoring (4).

The exponent in the denominator of (5) is now linear in a transformed version of X: after the d-dimensional feature vector is shifted by the average of $\{\mu_0, \mu_1\}$, it is augmented with a $(d+1)^{th}$ component equal to 1.

2.1.1 Gaussian C.C.D.'s – the Identity Class-independent Covariance Matrix

Consider the geometry of the posterior (5) for the case of $\Sigma = \mathcal{I}$, the $d \times d$ identity matrix. As shown in Figure 4, both class-conditional densities take on circular level-sets of equal radius (across classes) centered around μ_1 and μ_0. As Equation 5 becomes

$$p(Y=1 \mid x) = \frac{1}{1 + e^{-(\mu_1-\mu_0)^T\left(x - \frac{\mu_0+\mu_1}{2}\right) + \log\frac{1-\pi}{\pi}}}$$

we can see that the exponent is equal to $x \in \mathbb{R}^d$ shifted by the vector $\frac{\mu_0-\mu_1}{2}$ (half-way along the line-segment from μ_0 to μ_1), projected onto this same line segment, and then thresholded against the real valued $\log\frac{1-\pi}{\pi}$. Indeed, along lines perpendicular to $(\mu_1 - \mu_0)$, all x are shifted by the same vector and projected to the same value, thus resulting in the same posterior probability $p(Y=1 \mid x)$.

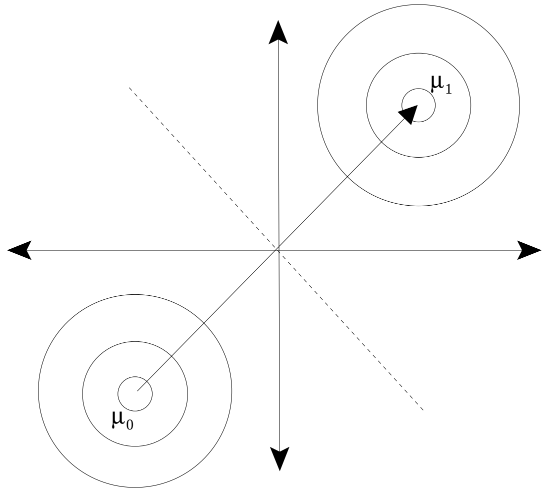

Figure 4: Generative modeling using Gaussian class-conditional densities with shared identity covariance matrix. C.C.D. level sets (posterior decision boundary) are displayed as solid (dashed) lines.

If, for an x on a given $(\mu_1 - \mu_0)$-orthogonal line, the projected length is equal to $-\log \frac{1-\pi}{\pi}$ then the thresholded value is 0 and the posterior evaluates to $\frac{1}{2}$. For positive thresholded values ($p(Y=1 \mid x) < \frac{1}{2}$), $Y=0$ is predicted for the given feature; for negative values ($p(Y=1 \mid x) > \frac{1}{2}$), $Y=1$ is predicted. Thus, as we sweep orthogonal lines from μ_0 to μ_1, we pass from predicting class 0 to predicting class 1 at the *decision boundary* $\log \frac{\pi}{1-\pi}$ from the mid-way point $\frac{\mu_0+\mu_1}{2}$. For $\pi = \frac{1}{2}$, this reduces to a linear decision boundary intersecting $\frac{\mu_0+\mu_1}{2}$.

How does this geometry change for $\pi \neq \frac{1}{2}$? As the prior π increases to 1, $\log \frac{\pi}{1-\pi}$ increases to infinity and the posterior $p(Y=1 \mid x)$ goes to 0 – we predict 0. Similarly, as π decreases to 0, the posterior goes to 1, and we predict class 1. Thus the geometry, matching our intuition, implies that the prior π affects a shift in the decision boundary away from the more likely class.

2.1.2 Gaussian C.C.D.'s – Arbitrary Class-independent Covariance Matrix

If $\Sigma \neq \mathcal{I}$, then the previous derivation of the posterior holds up until Equation 5, but no further. The covariance matrix does not vanish and our lines are not necessarily perpendicular to the line segment $(\mu_1 - \mu_0)$, although the posterior's level sets are still a set of parallel lines (see Figure 5).

2.2 Gaussian C.C.D.'s – Class-dependent Covariance Matrix

Consider now the case where each Gaussian class-conditional density is parameterized by a different covariance matrix Σ_1 and Σ_0. In this setting the contours of equal posterior probability are no longer lines as the quadratic $x^T x$ terms in (3) do *not* cancel. Instead the level sets are curves as in Figure 6.

The contour equation,

$$\frac{1}{1+e^{-Q(x)}} = C$$

can induce any quadratic form of level set, depending on the particular Q used. Examples include hyperbolas,

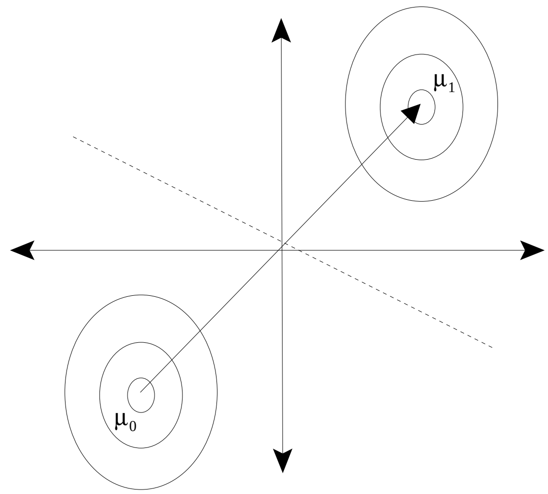

Figure 5: Generative modeling using Gaussian class-conditional densities with shared arbitrary covariance matrix. C.C.D. level sets (posterior decision boundary) are displayed as solid (dashed) lines.

parabolas and ellipses. Elliptical curves (centered around the μ_i) are somewhat of a defect rather than a feature of this *strong* parametric model: as the level sets loop back on themselves a flat C.C.D. for class 1 and concentrated C.C.D. for class 0 can result in points on the far side of class 0 being labeled (clearly) incorrectly as belonging to class 1, as shown in Figure 7. The additional power of this model with class-dependent covariance matrices may not necessarily reflect reality (and the underlying class-conditional densities may not even be Gaussian); furthermore, assuming $\Sigma_0 = \Sigma_1 = \Sigma$ is often a good idea computationally as we typically face problems with 100,000 of features which results in an estimation problem with a covariance matrix of $100,000^2$ elements – this becomes much worse when allowing class-dependent matrices. The simple discriminative logistic regression framework does not allow the unwanted situation of Figure 7 to occur – therefore, it is more *robust*.

2.3 Summary

We have shown that Gaussian class-conditional densities in the generative model reduces to logistic regression in the discriminative model with certain $Q(x)$ functions. The reverse is not necessarily true. Many different class-conditional densities (including non-Gaussian members of the exponential family, and others) can produce the same logistic posterior function.

We already know how to estimate the parameters of this model (e.g., with $\Sigma_1 = \Sigma_0 = \Sigma$) – μ_0, μ_1, Σ and π. As we will see in the next section, we can estimate π by treating Y as a multinomial of two classes, estimating the Gaussian parameters μ_1 and μ_0 separately by partitioning the examples in $\mathcal{X}$ by their class labels, and the covariance matrix Σ by pooling all the data together.

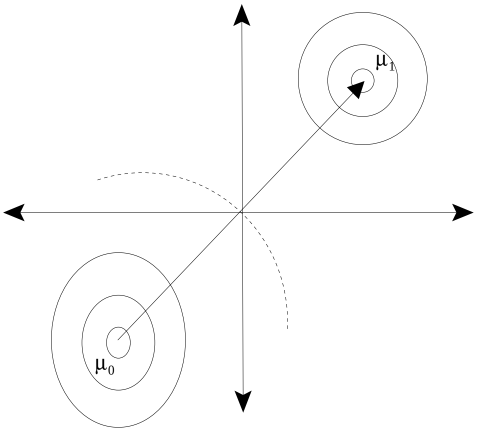

Figure 6: Generative modeling using Gaussian class-conditional densities with class-dependent covariance matrices. C.C.D. level sets (posterior decision boundary) are displayed as solid (dashed) lines.

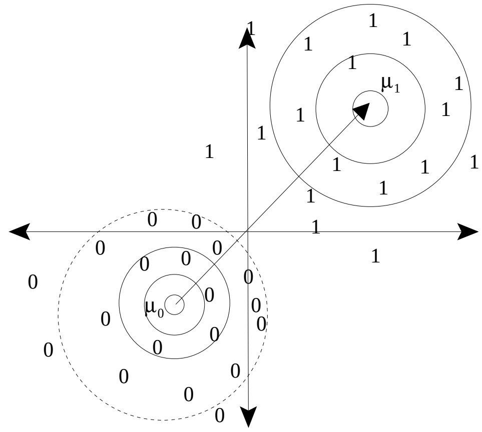

Figure 7: The problem with elliptical posterior level sets. The Gaussian C.C.D.'s shown result in a posterior that can incorrectly label points on the far side of class 0 as belonging to class 1.

3 Maximum Likelihood Estimation

3.1 The Generative Model – Estimating μ_0, μ_1, Σ and π

Let us consider maximum likelihood estimation for the generative model of binary classification with Gaussian class-conditional densities that have a class-independent covariance matrix. As we shall see in subsequent lectures, the ML estimates for the 4 parameters turn out to be

$$\begin{aligned}\hat{\pi}_{ML} &= \text{proportion of examples with } Y_i = 1\\ \hat{\mu}_{0,ML} &= \text{sample mean of } \{x_i \mid Y_i = 0\}\\ \hat{\mu}_{1,ML} &= \text{sample mean of } \{x_i \mid Y_i = 1\}\\ \hat{\Sigma}_{ML} &= \text{pooled sample covariance matrix}\end{aligned}$$

all of which are easy to compute given a data set $\mathcal{X}$. Now, to perform prediction on an new feature $X_{N+1} \in \mathbb{R}^d$ via the posterior $p(Y_{N+1} \mid x_{N+1})$, we simply plug the above maximum likelihood estimates into the logistic function derived in (5).

3.2 The Discriminative Model – Estimating θ

How do these ideas transfer to logistic regression for the discriminative model? Logistic regression uses the now familiar logistic posterior

$$p(Y = 1 \mid x) = \frac{1}{1 + \exp -\theta^T x} \tag{6}$$

We might try to induce $\hat{\theta}$ in (6) by 'plugging-in' the maximum likelihood estimates $\hat{\pi}_{ML}$, $\hat{\mu}_{0,ML}$, $\hat{\mu}_{1,ML}$ and $\hat{\Sigma}_{ML}$ obtained for the generative model above. However as we shall see, this estimate of θ may not correspond to $\hat{\theta}_{ML}$.

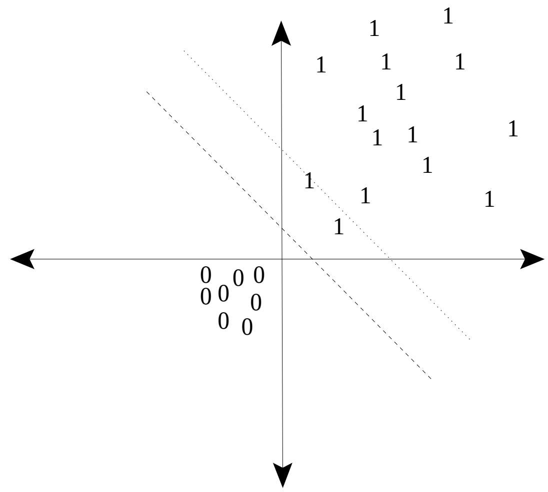

Figure 8: Estimated decision boundaries for generative (dotted line) and discriminative (dashed line) models when the data is generated with Gaussian class-conditional distributions with class-dependent covariance matrices. Having to model the features X means that the generative model can lead to incorrect boundaries when the assumption of class-independent covariance matrices is incorrect. The discriminative model cannot make this mistake as X is not modeled.

To see why this might be the case, consider the following situation depicted in Figure 8. Suppose that, contrary to our assumptions in the generative model, the true (Gaussian) class-conditional densities do *not* have a single class-independent covariance matrix. As shown in the figure, while the scatter plot of the data has class-clusters of different sizes (corresponding to the different underlying covariance matrices), maximum likelihood estimation of a single class-independent matrix Σ results in a $\hat{\Sigma}_{ML}$ which is 'smaller' than Σ_1 while 'larger' than Σ_0. This ill-fitted covariance matrix results in the decision boundary being too close to the 1 class, in turn resulting in a disproportionate number of *false negatives*[1].

The discriminative model, on the other hand, simply tries to find a *separating decision boundary* – a boundary where as many as possible of the examples of differing class are separated by the boundary. As demonstrated in Figure 8, this more direct goal can result in better classification accuracy – in general this improved real-life performance is partly due to the fewer assumptions made by discriminative approaches about the data at hand.

We next see a concrete example of the natural deterministic logistic model being more general than the generative Gaussian model.

4 The Exponential Family of Distributions

The *exponential family* of probability distributions have densities of the form:

$$p(x \mid y) \quad = \quad h(x)e^{\eta^T T(x) - A(\eta)}$$

This family includes many special cases including the Gaussian, Binomial, Gamma, Poisson, Dirichlet distributions, and many others. The exponential family is relatively easy to work with in its general form (later

[1]Interpreting the 0 (1) class as 'negative' ('positive'), a false negative is a feature x_i lying on the 0 side of the decision boundary, with $(x_i, 1) \in \mathcal{X}$.

we will look at identifying family members), and in fact naturally gives rise to a logistic posterior.

Consider a binary generative model in which the class-conditional densities are instances of the same distribution in the exponential family (with possibly different parameter values):

$$\begin{aligned} p(x \mid Y=0) &= h(x)e^{\eta_0^T T(x) - A(\eta_0)} \\ p(x \mid Y=1) &= h(x)e^{\eta_1^T T(x) - A(\eta_1)} \end{aligned}$$

Applying Bayes rule with prior $\pi = p(Y=1)$ achieves the posterior

$$p(Y=1 \mid x) = \frac{1}{1 + e^{-(\eta_1-\eta_0)^T T(x) - A(\eta_0) + A(\eta_1) + \log\frac{1-\pi}{\pi}}} \tag{7}$$

Consider the exponent in this expression. The term $-A(\eta_0) + A(\eta_1) + \log\frac{1-\pi}{\pi}$ is independent of x. The remainder of the exponent, $-(\eta_1 - \eta_0)^T T(x)$, is linear in $T(x)$. Therefore if we let z be $T(x)$ augmented with a constant 1 and θ be $\eta_1 - \eta_0$ augmented with an appropriate constant value, the exponent becomes linear in z and the posterior (7) is transformed into the familiar logistic function

$$p(Y=1 \mid x) = \frac{1}{1 + e^{-\theta^T z}} \tag{8}$$

Estimation under the generative model (with a fixed exponential family C.C.D.) involves maximum likelihood estimation of each parameter (i.e. η_0, η_1 and π in (7)); under discriminative classification θ is estimated directly. Thus, unless correct generative model assumptions can be made, the discriminative model (via logistic regression) will lead to more robust classification.

The importance of logistic regression in discriminative classification, particularly the connections it makes between generative and discriminative models – as seen in the present lecture – was historically neglected in the literature until very recently.

Logistic regression with an un-transformed x (i.e. where $x = z$) does not have the expressive power of the full exponential family – often we have no reason to expect any particular transformation of x. However certain problems require nonlinear decision boundaries. For example, see Figure 9. In this case one could use elliptical (or general quadratic) transformations to tackle the problem effectively. Indeed, logistic regression (with transformed features) is more broadly applicable than the exponential family; while all members of the exponential family produce logistic posteriors, logistic posterior can be induced by non-exponential family distributions. Logistic regression generalizes to kernelized logistic regression which is more powerful that the present case – such techniques will be covered in the spring sequel to this course.

Logistic regression also works naturally with discrete feature vectors by treating them in the same way as real-valued features.

On the downside, higher-dimensional transformations of x (e.g. $z = (x, x^2)$) require more data to achieve good estimation. However even if the given data is not linearly separable, the untransformed logistic function may well provide reasonable discriminative power.

5 Estimation under the Discriminative Model

Let us consider maximum likelihood estimation of θ in (8) as a simple case to demonstrate the method of estimation for discriminative models. Being frequentist, inference under the discriminative model of Figure 3 and the logistic function (8) involves estimating θ from $\mathcal{X}$ and then plugging the estimate into the logistic to make subsequent predictions; so estimation is clearly an important task. We start by re-writing the posterior

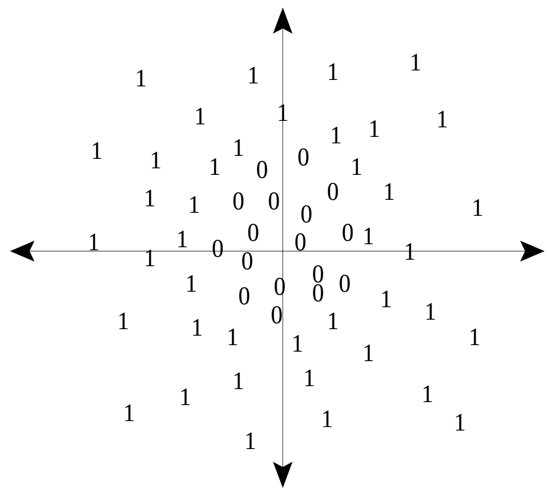

Figure 9: A binary classification problem with non-linearly separable $\mathcal{X}$.

so that it depends on the values of both x and y:

$$p(y \mid x) \quad = \quad \frac{e^{y\theta^T x}}{1 + e^{\theta^T x}}$$

Next we define the likelihood and log-likelihood functions $L(\theta)$, $l(\theta)$ respectively:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{N} p(y_i \mid x_i, \theta) \\ l(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^{N} \log p(y_i \mid x_i, \theta) \\ &= \sum_{i=1}^{N} y_i \theta^T x_i - \sum_{i=1}^{N} \log(1 + e^{\theta^T x_i}) \end{aligned}$$

And the ML estimator of θ is

$$\hat{\theta}_{ML} \quad = \quad \operatorname*{argmax}_{\theta} l(\theta)$$

Setting $\frac{dl}{d\theta}(\hat{\theta}_{ML}) = 0$ does not yield a closed-form expression for $\hat{\theta}_{ML}$ as it has on several previous occasions. Thus we must turn to a general-purpose optimizer, such as the widely popular Newton-Raphson algorithm. Let us then consider how *gradient ascent* would proceed for this problem, starting with the derivative:

$$\begin{aligned} \frac{dl}{d\theta} &= \sum_{i=1}^{N} \left(y_i x_i - \frac{e^{\theta^T x_i}}{1 + e^{\theta^T x_i}} x_i \right) \\ &= \sum_{i=1}^{N} (y_i - \mu_i)\, x_i \end{aligned}$$

where[2] $\mu_i = p(Y_i = 1 \mid x_i, \theta)$.

[2] We use the symbol μ as the stated probability is in fact a mean as in $\mu_i = p(Y_i = 1 \mid x_i, \theta) = E[1_{\{Y=1\}} \mid x, \theta]$.

5.1 Stochastic Gradient Ascent

A variant of gradient ascent particularly well-suited to large-scale problems (those with large N) is *stochastic gradient ascent*:

$$\Delta\theta \quad = \quad \rho(y_i - \mu_i)x_i \tag{9}$$

where $\rho > 0$ is the *step-size*. In the usual gradient ascent algorithm we run through all the data in $\mathcal{X}$, summing and making one step $\Delta\theta$ up-hill, repeating until some stopping criterion is met. For (typically) large datasets with $N = 10^6$ this would involve a million computational steps before we can increment $\hat{\theta}$. In contrast, stochastic gradient ascent takes a subsample of the points in $\mathcal{X}$ and proceeds over these rather than the (much larger) $\mathcal{X}$, thereby reducing the number of required computations for each increment of $\hat{\theta}$. The limiting case of this method is of course stochastic LMS where each increment of $\hat{\theta}$ is made with a single randomly selected example from $\mathcal{X}$.

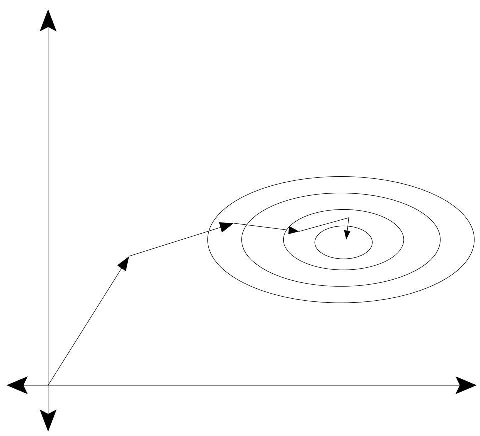

Figure 10: A trajectory of $\hat{\theta}^{(t)}$ in stochastic gradient ascent.

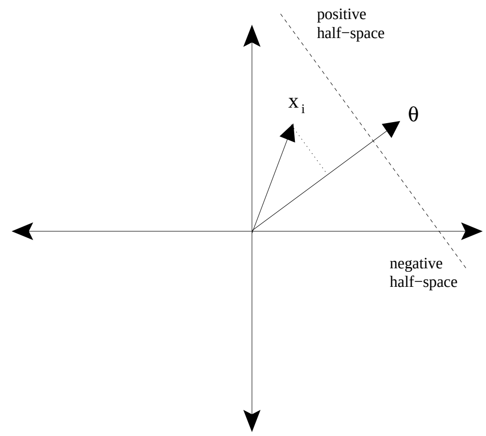

Figure 11: Updating $\hat{\theta}$ with an example whose feature x_i falls on the 0-class side of the logistic decision boundary.

What is the intuition behind the gradient ascent update rule of (9)? Given an example $(x_i, y_i) \in \mathcal{X}$ which has a posterior value $\mu_i = 0.1 < \frac{1}{2}$ (see Figure 11), consider the effect of different values that y_i would take on. If $y_i = 0$ then $\Delta\theta = \rho(0 - 0.1)x_i$ and the boundary goes towards x_i but does not cross it (we are making correct predictions). If $Y_i = 1$ then $\Delta\theta = \rho(1 - 0.1)x_i$ and the boundary tilts strongly towards x_i – we see that the update rule (9) is a simple error correction procedure.

How does this procedure compare to linear regression? In linear regression, outliers be have a large effect on updates, since in that method

$$\Delta\theta = \rho(y_i - \theta^T x_i)x_i$$

while in logistic regression

$$\Delta\theta = \rho\left(y_i - \frac{1}{1 + e^{-\theta^T x_i}}\right)x_i$$

Due to the sigmoidal shape of the logistic function, $\mu_i = \frac{1}{1+e^{-\theta^T x_i}}$ is bounded between 0 and 1, and the effect of an outlier is far less than in linear regression.