STA561: Probabilistic machine learning

Generalized Linear Models (9/16/13)

Lecturer: Barbara Engelhardt *Scribes: Andrew Ang, Shirley Liao, Taylor Pospisil, Zilong Tan*

1 Generative versus discriminative classifiers

Graphical representations of a generative classifier and a discriminative classifier are shown in Figure 1. The discriminative classifier models a single probability distribution $p(y|x)$ whereas the generative classifier models two distributions: $p(y)$ and $p(x|y)$. In general, generative classifiers are less accurate than discriminative classifiers in the limit of infinite data, but generative classifiers are more resilient to missing features and small training data sets, and generative classifiers often have fewer parameters to estimate. In this lecture, we consider discriminative classifiers that are linear in their parameters.

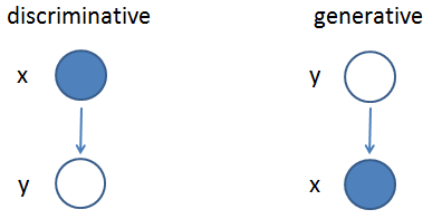

Figure 1: Graphical model representation of discriminative versus generative classifiers.

2 Classification using a generalized linear model (GLM)

2.1 Logistic regression model

Consider a scalar response, $y \in \mathbb{R}$, given a vector of features, $x \in \mathbb{R}^p$. If we consider the response to have a Gaussian distribution, and we include vector $Y = [y_1, \ldots, y_n]^T$ and matrix $X = [x_1^T, \ldots, x_n^T]$ by considering n samples of each y and x, then $Y = X\beta + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is a random noise variable with known variance. This is the model for linear regression, with coefficients $\beta \in \mathbb{R}^p$. We then describe the distribution of the response variable conditional on the other variables and parameters: $Y|X, \beta \sim \mathcal{N}(X\beta, \sigma^2)$.

What if we want a classifier? In other words, what if $y \in \{0, 1\}$? Our first instinct might be to assume that y is from a Bernoulli distribution, $y|x, \beta \sim Ber(x^T\beta)$. But this does not work since $x^T\beta \notin [0, 1]$: the parameter to a Bernoulli is the probability of a 1, which is a probability between zero and one. To ensure $x^T\beta \in [0, 1]$, we will introduce the 'sigmoid' or 'logistic' function, which squashes our variables $x^T\beta$ into the space $(0, 1)$:

$$\pi(x^T\beta) = \frac{1}{1+\exp\left(-x^T\beta\right)}. \tag{1}$$

Thus, we can model $y|x,\beta$ as $\text{Ber}\left(\pi\right)$. Consider our example of classifying emails as 'spam' (1) or 'non-spam' (0), where y_i is the class label of the ith email and x_i contains a vector of features from the email. We use the sigmoid function to map $x_i^T\beta \to [0,1]$. Then, after fitting the model to data, we can select a threshold between 0 and 1 (say, $t = 0.5$), to distinguish between 'spam' and 'non-spam' emails as follows:

For example, say we are given a test sample ($x^\star$), and we have a fitted model $\hat{\beta}$ from training data. We can estimate the value of $y^\star$ and set a threshold for our class labels at 0.5:

$$p\left(y^\star|\hat{\beta},X^\star\right) = \frac{1}{1+e^{-(x^\star)^T\hat{\beta}}}$$

$$y^\star = \begin{cases} 1 & p\left(y^\star|\hat{\beta},X^\star\right) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

It should be noted that

$$x^\star = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_p \end{bmatrix}, \hat{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

where β_0 is the intercept. As $\hat{\beta}_j$ becomes larger in absolute value, the classifier becomes more distinctive (logistic function has a steeper slope) as is depicted in Figure 2. Larger absolute values of the coefficient imply a small uncertainty between the class labels. Whereas, when $\hat{\beta}_j$ is closer to zero, the logistic function has a shallower slope; in other words, there is greater uncertainty about the class label predictions from the features, thus there is a greater range of predictions closer to 0.5 near β_0.

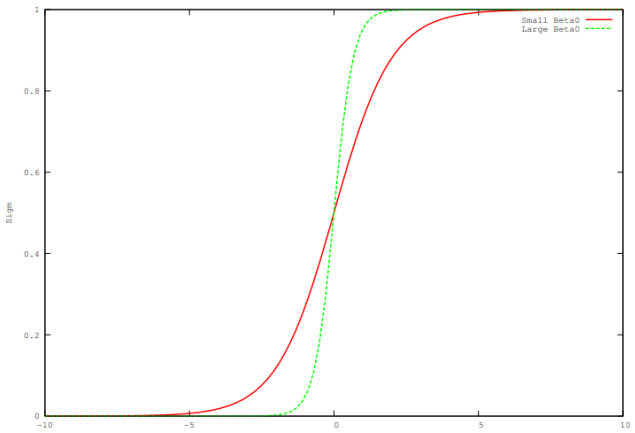

Figure 2: The sigmoid function

It is worth noting that since $\pi\left(x^T\beta\right)$ is constrained within $(0,1)$, outliers in the feature space (x values outside of the general range of the other samples) have a smaller impact on the model than in the linear regression case.

2.2 MLE of β

The next question is, how do we estimate β for logistic regression? Let $\mathcal{D}_{train} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ be the training data set. We want to determine the maximum likelihood estimates for β given the logistic regression model (Eqn. 1). We will start by writing the log likelihood of the response Y:

$$\begin{aligned}
l(\beta|\mathcal{D}) &= \log \prod_{i=1}^{n} \left[p(y_i = 1|x_i, \beta)^{\mathbf{1}(y_i=1)} p(y_i = 0|x_i, \beta)^{\mathbf{1}(y_i=0)} \right] \\
&= \sum_{i=1}^{n} \left[y_i \log \frac{1}{1 + e^{(-x_i^T \beta)}} + (1 - y_i) \log \frac{1}{1 + e^{(x_i^T \beta)}} \right] \\
&= \sum_{i=1}^{n} \left[y_i \log \frac{e^{(x_i^T \beta)}}{1 + e^{(x_i^T \beta)}} - \log \left(1 + e^{(x_i^T \beta)} \right) - y_i \log \frac{1}{1 + e^{(x_i^T \beta)}} \right] \\
&= \sum_{i=1}^{n} \left[y_i \log e^{(x_i^T \beta)} - \log \left(1 + e^{(x_i^T \beta)} \right) \right] \\
&= \sum_{i=1}^{n} \left[y_i x_i^T \beta - \log \left(1 + e^{(x_i^T \beta)} \right) \right]
\end{aligned}$$

Let $\mu_i = \frac{1}{1+e^{(-x_i^T \beta)}}$. Then we can try to take the derivative of this log likelihood, set to zero, and solve for our parameter β as usual to get the MLE:

$$\begin{aligned}
\frac{\partial l(\beta|\mathcal{D})}{\partial \beta} &= \sum_{i=1}^{n} \left[y_i x_i - \frac{e^{(x_i^T \beta)}}{1 + e^{(x_i^T \beta)}} x_i \right] \\
&= \sum_{i=1}^{n} (y_i x_i - \mu_i x_i) \\
0 &= \sum_{i=1}^{n} (y_i - \mu_i) x_i \qquad (2)
\end{aligned}$$

Because there is no closed form solution for β given Eqn. 2, we cannot directly solve for β as in the normal equation for linear regression. Instead we explore optimization methods to approximate β.

2.3 Optimization methods

2.3.1 Stochastic gradient ascent

Stochastic gradient ascent is a simple iterative algorithm to optimize convex and nonconvex functions. For each iteration we randomly pick a sample from our training set and update our parameter, $\beta^{(t)}$, by taking a step in the direction of the gradient, $f'(\beta^{(t)})$. This is summarized by the following rule:

$$\beta^{(t)} \leftarrow \beta^{(t)} + \tau_t f'(\beta^{(t)}) \qquad (3)$$

where $\beta^{(t+1)}$ is the updated parameter, $\beta^{(t)}$ is the previously estimated parameter, τ_t is the step size or learning rate at iteration t, and $f'(\beta^{(t)})$ is the gradient at point $\beta^{(t)}$. One possible strategy for updating the

step size is to pick the τ_t that maximizes $f(\beta^{(t)} + \tau_t f'(\beta^{(t)}))$. This is called a line search method and can be solved by an appropriate one-dimensional method. In general, the trade off is that large step sizes will run fast but may not converge, and small step sizes will be more likely to converge to a locally optimal solution but will be computationally more slow.

The specific update for estimating β in linear regression is:

$$\beta^{(t+1)} \leftarrow \beta^{(t)} + \tau_t(y_i - x_i^T \beta^{(t)})x_i.$$

Estimating β for logistic regression is:

$$\beta^{(t+1)} \leftarrow \beta^{(t)} + \tau_t(y_i - \frac{1}{1 + \exp\left(-x_i^T \beta^{(t)}\right)})x_i.$$

Both of these can be written as follows for appropriately defined $\mu_i^{(t)}$:

$$\beta^{(t+1)} \leftarrow \beta^{(t)} + \tau_t(y_i - \mu_i^{(t)})x_i.$$

In the logistic regression case we are normalizing the feature space $x^T\beta$ to be within $[0, 1]$. This greatly diminishes the effect of outliers in the feature space for the stochastic gradient ascent updates when compared to the updates for the linear regression: the largest that the residual term could possibly be is 1.

2.3.2 Newton-Raphson method

Another more robust approach to estimating the MLE of the logistic regression coefficients is the Newton-Raphson method. The goal of this method is to find β such that $f'(\beta) = 0$ by using the 2^{nd} order Taylor series expansion:

$$\begin{aligned} \beta^{t+1} &\leftarrow \beta^t - \frac{f'(\beta^t)}{f''(\beta^t)} \\ \beta^{t+1} &\leftarrow \beta^t - H^{-1} f'(\beta^t) \end{aligned}$$

Where H is the Hessian matrix given by:

$$H = f''(\beta^t), \qquad H = \frac{\partial^2 f}{\partial\beta\partial\beta^T}.$$

The Newton-Raphson method is similar to the gradient ascent method described above, but the main difference between the two is that, in the Newton-Raphson method, we use the Hessian to dictate learning rate (step size). Specifically, we need to compute the inverse of the Hessian, which may (or may not) be computationally expensive to compute and to invert. On the other hand, the Newton-Raphson method generally requires fewer iterations to converge than the stochastic gradient ascent method, because of this adaptive step size, and is often easier to implement, because the step size is not a parameter that needs to be adjusted carefully.

For logistic regression, our function $f(\cdot)$ that we wish to find the zeros of is the log likelihood function $l(\beta; \mathcal{D})$. We take the partial derivatives of $l(\beta; \mathcal{D})$ with respect to β:

$$\begin{aligned} \frac{\partial^2 l}{\partial\beta\ \partial\beta^T} &= \sum_{i=1}^{n} x_i \frac{\partial\mu_i}{\partial\beta} \\ &= \sum_{i=1}^{n} -x_i x_i^T(\mu_i(1-\mu_i)) = -\sum_{i=1}^{n} x_i^T S x_i \\ &\quad \text{where } S = diag(\mu_i(1-\mu_i)). \end{aligned}$$

Thus, the Newton-Raphson update for logistic regression is:

$$\beta^{t+1} \quad \leftarrow \quad \beta^t + (x_i^T S_t x_i)^{-1} x_i^T (y_i - \mu_i),$$

which is iteratively reweighted least squares (IRLS). Notice that S is the variance term for our predicted response y: this means that, when our uncertainty is greater for our predictions (i.e., when the predictions are closer to 0.5), we will tend to take smaller step sizes, whereas when we are more certain about our predictions (i.e., the are closer to 0 or 1) our step sizes will be greater.

3 Generalized linear models using exponential family

3.1 Canonical link function

Recall for the exponential family,

$$p(y|\eta) = h(y)\ \exp\{\eta^T T(y) - A(\eta)\}$$

Where $h(y)$ is the scaling constant, η are the natural parameters, $T(y)$ are the sufficient statistics, and $A(\eta)$ is the log partition function.

We would like our regression model $p(y|x, \beta)$ to be described in the exponential family form; in other words, $\mu = \mathbb{E}_\eta[y|x] = f(x^T\beta)$. But what function $f(\cdot)$ should we choose? We will show that choosing the *canonical* response function given a specific choice of generalized linear model (GLM) is usually a good starting point for modeling your data. By using the canonical link and response functions for exponential family distributions, we get simple maximum likelihood estimate (MLE) derivations that depend only on the sufficient statistics and response function. We can plug these derivations directly into the stochastic gradient ascent equation to have an general approach for prediction and parameter estimation for GLMs within the exponential family.

Let us set the natural parameter $\eta = x^T\beta$. Then $\mu = g^{-1}(x^T\beta)$ and $x^T\beta = g(\mu)$, where g^{-1} and g are the canonical response and link functions, respectively. The mean parameters, μ, when we model our data using Bernoulli (logistic regression) and Poisson (Poisson linear regression) distributions are as follows. Recall for Bernoulli:

$$\begin{aligned}
P(y|\eta) &= \exp\{y\log(\frac{\mu}{1-\mu}) + \log(1-\mu)\} \\
\Rightarrow \eta = \log(\frac{\mu}{1-\mu}) &\qquad \Rightarrow \mu = \frac{1}{1+e^{-\eta}} \\
\text{If we let } \eta = x^T\beta, \text{ then} & \\
P(y|x,\eta) = P(y|x,\beta) = \mu &= \frac{1}{1+e^{-x^T\beta}}
\end{aligned}$$

which is the logistic regression model we described earlier.

Recall for Poisson:

$$\begin{aligned}
P(y|\eta) &= \frac{1}{y!}\exp\{y\log(\lambda) - \lambda\} \\
\Rightarrow \eta = \log(\lambda) &\qquad \Rightarrow \lambda = e^\eta \\
\text{If we let } \eta = x^T\beta, \text{ then} & \\
\mu &= e^{x^T\beta} \\
P(y|\eta) = P(y|x^T\beta) &= E[y|x] = e^{x^T\beta}.
\end{aligned}$$

Thus, given fitted $\hat{\beta}$ parameters for a Poisson linear model, we can predict y^* given a new data point x^* using the equation above: $y^* = e^{x^{*T}\beta}$. Furthermore, the mean of the Poisson conditional model is given by that same term: $\mu^* = \lambda^* = e^{x^{*T}\beta}$.

3.2 MLE of β for exponential family

Now the MLE derivation for an arbitrary GLM in the exponential family is straightforward.

$$\begin{aligned}
\log\, l(\beta|D) &= \sum_{i=1}^{n} \log(P(y_i|x_i,\beta)) \\
&= \sum_{i=1}^{n} x_i^T \beta T(y_i) - A(x_i^T\beta) \\
\frac{\partial \log\, l(\beta|D)}{\partial \beta} &= \sum_{i=1}^{n} x_i T(y_i) - \left(\frac{\partial A}{\partial \eta_i}\right)\left(\frac{\partial \eta_i}{\partial \beta}\right), \quad \text{recall } \frac{\partial A}{\partial \eta_i} = \mu_i \\
&= \sum_{i=1}^{n} x_i T(y_i) - \mu_i x_i \\
&= \sum_{i=1}^{n} x_i (T(y_i) - \mu_i).
\end{aligned}$$

Recall that $T(y_i)$ is the sufficient statistics and μ_i is the response function. Note $(T(y_i) - \mu_i)$ the residual: the difference between the predicted output and the actual output. Thus, the MLE derivations for the Bernoulli and Poisson regression parameters β are:

Bernoulli:

$$\frac{\partial l}{\partial \beta} = \left(y_i - \frac{1}{1 + e^{-x_i^T\beta}}\right) x_i$$

Poisson:

$$\frac{\partial l}{\partial \beta} = (y_i - e^{x_i^T\beta}) x_i$$

We partial derivatives for GLMs within the exponential family to show a general form of stochastic gradient ascent:

$$\beta^{(t+1)} \leftarrow \beta^{(t)} + \tau_t (y_i - \mu_i) x_i. \tag{4}$$

All of these echo our findings for logistic regression. Iteratively reweighted least squares (or Newton-Raphson) can be generated similarly for each of these GLMs with canonical link functions by computing the Hessian of the log likelihood with respect to the parameters β, and plugging these into the general form of the Newton-Raphson updates.