A Statistical Approach to Classification: the Generative Story

Guy Lebanon

Experimentation alone does not offer a clear understanding of which classifiers to use and when. Statistics offers a useful theory that can be used to derive optimal classifiers. We review some of its basic ideas here.

We assume our data are pairs (X, Y) where $X \in \mathcal{X}, Y \in \mathcal{Y}$. We also assume that the data is drawn iid from a distribution $p(X, Y)$ which implicitly determines also the distributions $p(X), p(Y), p(X|Y), p(Y|X)$. The first stage is to define a loss function $L(Y, \hat{Y})$ which determines how much we care about predicting $\hat{Y}$ when the true label is Y. Typically $L(Y, Y) = 0$. The popular 0-1 loss is $L(Y, \hat{Y}) = 1$ for $Y \neq \hat{Y}$ and 0 otherwise (all wrong predictions provide equal loss). But in other cases such as spam filtering or cancer detection asymmetric loss should be used (misdiagnosing cancer as no-cancer is much worse than vice verse!).

The next stage is to agree that we measure the quality of a classifier $f : \mathcal{X} \to \mathcal{Y}$ using the expected loss, that is the average loss we incur if we keep using the classifier for a very long time. We call the expected loss the classification risk and denote it as $R(f) = E_{p(X,Y)} L(Y, f(X))$. The classifier $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the risk above is called the optimal classifier or the Bayes classifier and is denoted f^*. It is simple to derive

$$f^*(X) = \arg\min_{y \in \mathcal{Y}} E_{p(Y|x)} L(Y, y) = \arg\min_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} p(y'|X) L(y', y) \tag{1}$$

which simplifies in the 0-1 loss case to $f^*(X) = \arg\max_y p(y|X)$. In the case of binary 0-1 loss ($\mathcal{Y} = \{+1, -1\}$)

$$f^*(X) = \operatorname{sign} \log \frac{p(Y=1|X)}{p(Y=-1|X)} = \operatorname{sign} \log \frac{p(X|Y=1)p(Y=1)}{p(X|Y=-1)p(Y=-1)} \tag{2}$$

$$R(f^*) = E_{p(X)} \min\{p(Y=1|X), 1 - p(Y=1|X)\} = \frac{1}{2} - \frac{1}{2} E_{p(X)} |2p(Y=1|X) - 1|. \tag{3}$$

The problem is that we don't know $p(X, Y)$. Instead we have a training set $(X^{(1)}, Y^{(1)}), \ldots, (X^{(n)}, Y^{(n)}) \overset{\text{iid}}{\sim} p$. A clear way to proceed is to estimate the distribution (say using maximum likelihood) and plug the estimator $\hat{p}(X, Y)$ into (1) or (2) to obtain an estimate for f^*. The quality of the resulting $\hat{f}^*$ would depend on the quality of the approximation $\hat{p} \approx p$.

Gaussian Case (Quadratic Discriminant Analysis)

We assume here $X|\{Y = y\} \sim N(\mu_y, \Sigma_y)$ (μ_y, Σ_y are the mean vector and covariance matrix of $X|Y = y$)

$$p(X = x|Y = y) = (2\pi)^{-d/2} |\Sigma_y|^{-1/2} \exp\left(-\frac{1}{2}(x - \mu_y)^\top \Sigma_y^{-1} (x - \mu_y)\right) \tag{4}$$

where $|A|$ is the determinant of A. In the binary case (2) becomes

$$f^*(X) = \operatorname{sign}\left(\log \frac{|\Sigma_{-1}|}{|\Sigma_1|} + 2 \log \frac{p(Y=1)}{p(Y=-1)} - (X - \mu_1)^\top \Sigma_1^{-1} (X - \mu_1) + (X - \mu_{-1})^\top \Sigma_{-1}^{-1} (X - \mu_{-1})\right). \tag{5}$$

The resulting classifier is called quadratic discriminant analysis since the algebraic form above is quadratic in X: $f^*(x) = \operatorname{sign}(x^\top A x + b^\top x + c)$ and since the decision boundary in $\mathcal{X} = \mathbb{R}^d$ is quadratic (see Figure 1).

As mentioned above the class conditional distributions are typically unknown and in practice the expressions $\mu_y, \Sigma_y, p(y = 1)$ for all y need to be replaced with estimates, for example using MLE. In the Gaussian case these are the empirical mean for μ_y, empirical covariance for Σ_y, and label proportions for $p(y)$.

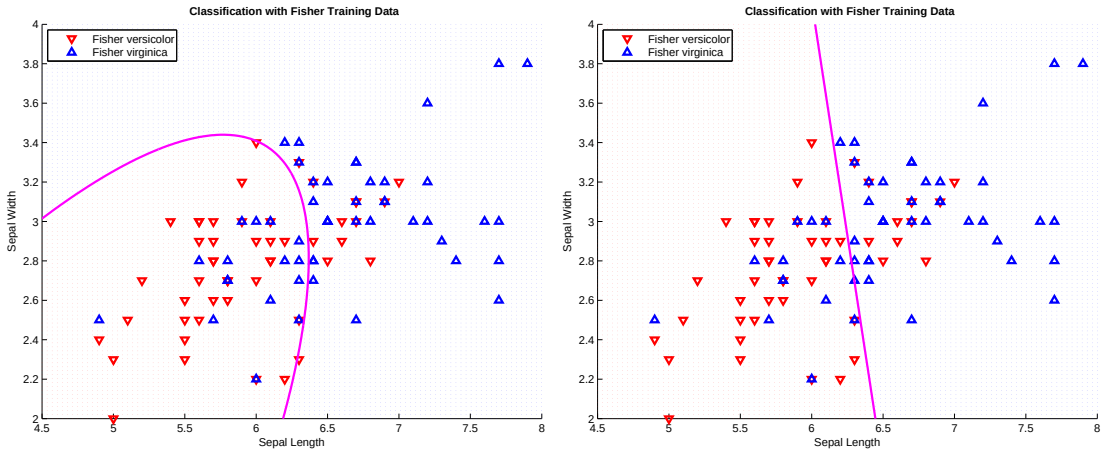

Figure 1: Decision boundary corresponding to optimal classifier f^* (under 0-1 loss) for a two dimensional Iris data when $p(X|Y=y)$ is Gaussian with $\Sigma_1 \neq \Sigma_{-1}$ (left) and $\Sigma_1 = \Sigma_{-1}$ (right).

Gaussian Case (Linear Discriminant Analysis)

The quadratic discriminant above becomes linear when $\Sigma_1 = \Sigma_{-1} = \Sigma$

$$f^*(X) = \text{sign}\left(2\log\frac{p(Y=1)}{p(Y=-1)} - X^\top\Sigma^{-1}(\mu_{-1}-\mu_1) - (\mu_{-1}-\mu_1)^\top\Sigma^{-1}X - \mu_1'\Sigma^{-1}\mu_1 + \mu_{-1}^\top\Sigma^{-1}\mu_{-1}\right) \quad (6)$$

$$= \text{sign}\left(2(\mu_1-\mu_{-1})^\top\Sigma^{-1}X + \mu_{-1}^\top\Sigma^{-1}\mu_{-1} - \mu_1^\top\Sigma^{-1}\mu_1 + 2\log\frac{p(Y=1)}{p(Y=-1)}\right) \quad (7)$$

$$= \text{sign}(w^\top X + c). \quad (8)$$

This linear form shows that the geometric shape of the decision boundary is a linear hyperplane (see Figure 1, right). If we also assume $\Sigma^{-1} = I$, $f^*(X) = \text{sign}(c - (\mu_{-1}-\mu_1)^\top X)$ i.e., the decision boundary is orthogonal to the vector connecting the two means (shifted by c), which makes sense in this spherically symmetric case. The plane will intersect the vector connecting the two means in the middle if $P(Y=1) = 1/2$ but will shift towards the less probable Gaussian otherwise.

Naive Bayes

When $p(X|Y)$ is not Gaussian and X is high dimensional it may be hard to estimate it from the training data. The difficulty is both computational and statistical: estimating a high dimensional distribution is often computationally expensive, and due the curse of dimensionality a small training set may result in very noisy estimate. A popular solution is to assume the factorization $p(X|Y) = \prod_i p(X_i|Y)$ which is equivalent to saying that the dimensions of X are conditionally independent given the label Y. This is called naive Bayes since the assumption is naive (yet nonetheless often useful!).

Perhaps the most popular application of naive Bayes is for categorical data in conjunction with the multinomial model and text documents. In text classification documents are represented by vectors X whose components X_i are either binary (whether word i appeared or not in the document) or a number between 0 and 1 (how many times word i appeared in the document divided by document length). Since the number of possible words is very large the random vector X is very high dimensional. Applying naive Bayes in this way we have $p(X|Y=y) = \prod_i p(X_i|Y=y)$ where $\theta_{iy} = p(X_i|Y=y)$ is a parameter vector defining a multinomial distribution (which may be easily estimated using relative frequency of words in documents associated with the different classes).

In practice, naive Bayes enjoys simplicity and computational efficiency. It also performs similarly for other methods for small training set sizes and high dimensions. But for larger training set sizes, the naive Bayes assumption is not necessary and more flexible models often performs better.