CO902
Supporting Lecture Notes:
Decision Theory Background of Classification
Lecture 4 — 28 Jan 2012

1 Decision Theory for Classification

In labs we have been building classifiers and prediction tools motivated by informal justification. Here is an attempt to formally derive why we build classifiers the we way we do. This follows roughly from section 1.5 of Bishop's text.

We are concerned with problems that take the form $\{\mathbf{X}_i, Y_i\}$, $i = 1, \ldots, n$, for data $\mathbf{X}_i \in \mathcal{X}$ and a class membership $Y_i \in \{1, 2, \ldots, K\}$. $\mathcal{X}$ can be a discrete or continuous sample space but usually have multiple dimensions, but for simplicity here we'll assume X_i is a discrete random variable.

In a decision theoretic approach, we wish to build a decision rule $D_{\mathbf{x}} : \mathcal{X} \to \{1, 2, \ldots, K\}$, a mapping from data to class, that minimizes a loss function. Loss functions can have very complicated forms, but in classification we simply want to minimize the chance of being *wrong*, i.e. misclassifying, or, equivalently, maximize the chance of being *right*.

First, it is helpful to define decision regions $\mathcal{R}_k$ that partition $\mathcal{X}$, such that if $\mathbf{x}$ falls in $\mathcal{R}_k$ the predicted class is k, i.e.

$$D_{\mathbf{x}} = \{k : x \in \mathcal{R}_k\}$$

The "best" decision rule $D_{\mathbf{x}}$ is then the one that maximizes the probability of being "correct", i.e. $P(D_{\mathbf{x}} = Y)$. Using the sum rule and then the product rule, we can see the answer:

$$\begin{aligned}
P(D_{\mathbf{x}} = Y) &= \sum_{k=1}^{K} P(D_{\mathbf{x}} = k, Y = k) \\
&= \sum_{k=1}^{K} P(\mathbf{x} \in \mathcal{R}_k, Y = k) \\
&= \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{R}_k} P(\mathbf{X} = \mathbf{x}, Y = k) \\
&= \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{R}_k} P(Y = k | \mathbf{X} = \mathbf{x}) P(\mathbf{X} = \mathbf{x})
\end{aligned}$$

Remember that a decision rule is built for each particular $\mathbf{x}$: For each $\mathbf{x}$, we have no control over $P(\mathbf{X} = \mathbf{x})$, but we can maximize this expression by making sure x belongs in the $\mathcal{R}_k$ that has the largest $P(Y = k | \mathbf{X} = \mathbf{x})$. (Note that the exact same logic works for continuous $\mathbf{X}$; pointwise, for each $\mathbf{x}$, we need the k that maximizes $P(Y = k | \mathbf{X} = \mathbf{x})$).

So there we go. By just using sum and product rules, the probability of being "right" is optimized by

$$D_x = \operatorname*{argmax}_k P(Y = k | \mathbf{X} = \mathbf{x})$$

That is, to find the "correct" class we just need to compute all K of the conditional distributions of Y given $\mathbf{x}$ to find the most likely class.

Now, let's see how this played out in our lab examples.

2 Markov Chain Prediction

In Lab 2 & 3 we considered predicting the final observation in a binary, time-invariant Markov Chain. Here, our data were $\mathbf{X} = (X_1, X_2, \ldots, X_{n-1})$, and our classes were $Y = X_n$. In this case, the conditional distributions to base our decision on are

$$\begin{aligned} P(Y = k|\mathbf{X} = \mathbf{x}) &= P(X_n = k|X_1, X_2, \ldots, X_{n-1}) \\ &= P(X_n = k|X_{n-1}) \end{aligned}$$

for $k = 0, 1$. We of course recognize these as the elements of the transition matrix

$$\mathbf{T} = \begin{bmatrix} P(X_n = 0|X_{n-1} = 0) & P(X_n = 1|X_{n-1} = 0) \\ P(X_n = 0|X_{n-1} = 1) & P(X_n = 1|X_{n-1} = 1). \end{bmatrix}$$

But, let's be clear: In our exercise, we *didn't* use this to make predictions! We used the data at hand to estimate $\hat{\mathbf{T}}$! So, to be very precise, our decision rule was

$$D_{\mathbf{x}} = \begin{cases} 0 & \text{if } \widehat{P}(X_n = 0|X_{n-1}) > \widehat{P}(X_n = 1|X_{n-1}) \\ 1 & \text{if } \widehat{P}(X_n = 0|X_{n-1}) < \widehat{P}(X_n = 1|X_{n-1}) \end{cases}$$

Many of you thought the prediction came from sampling a Bernoulli with probability $P(X_n = k|X_{n-1})$. This is bad for two reasons: First, it isn't the "optimal decision rule". The optimal decision rule $D_{\mathbf{x}}$ doesn't specify *generating* random data, it gives a specific *deterministic* mapping between data space $\mathcal{X}$ and class membership. Second, a randomized rule, as this is known, will give different (random) answers for different data, a very unwelcome attribute.

3 Spam prediction

In Lab 3 you were asked to build a spam classifier. You were given two vectors of data, X presence of the word "free" in the email, and Y hand-labeled classification of email as spam. Specifically, $X_i = 1$ if "free" was in the email, 0 otherwise, and $Y_i = 1$ if the email was spam, 0 otherwise. The optimal decision rule must be the one based on $P(Y|X)$, i.e. e.g. if $P(Y = 1|X) > 0.5$ call it spam, not otherwise. In such a simple problem, you could compute $P(Y = y|X = x)$ directly for all four combinations of x and y, *however* you were instructed to assume $P(Y = 1) = P(Y = 0) = 0.5$. This is the hint that you should use Bayes Rule:

$$P(Y = 1|X = x) = \frac{P(X = x|Y = 1)P(Y = 1)}{P(X = x|Y = 0)P(Y = 0) + P(X = x|Y = 1)P(Y = 1)}.$$

Thus, what was needed was $P(X = x|Y = 0)$ and $P(X = x|Y = 1)$, the "class conditional" distributions (the distribution of the data conditional on a given class). For some training data, these are easily computed...

$$\begin{aligned}
\widehat{P}(X = 1|Y = 0) &= \text{"proportion of 'free' emails among all non-spam emails"} \\
&= \frac{\#\{X_i = 1, Y_i = 0\}}{\#\{Y_i = 0\}} \\
\widehat{P}(X = 1|Y = 1) &= \text{"proportion of 'free' emails among all spam emails"} \\
&= \frac{\#\{X_i = 1, Y_i = 1\}}{\#\{Y_i = 1\}}
\end{aligned}$$

Using these *estimated* probabilities we then define our decision rule for a new email $(X_{\text{new}}, Y_{\text{new}})$

$$D_{\mathbf{x}} = \begin{cases} 0 & \text{if } \widehat{P}(Y_{\text{new}} = 0|X_{\text{new}}) > \widehat{P}(Y_{\text{new}} = 1|X_{\text{new}}) \\ 1 & \text{if } \widehat{P}(Y_{\text{new}} = 0|X_{\text{new}}) < \widehat{P}(Y_{\text{new}} = 1|X_{\text{new}}) \end{cases}$$

4 Optimal Classifier & Bayes Theorem - Shortcuts

For a general K class classifier problem, the optimal decision rule specifies that we must find k that maximizes

$$P(Y = k|\mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X} = \mathbf{x}|Y = k)P(Y = k)}{\sum_{k'=1}^{K} P(\mathbf{X} = \mathbf{x}|Y = k')P(Y = k')}.$$

However, note that the denominator is independent of k, thus

$$P(Y = k|\mathbf{X} = \mathbf{x}) \propto P(\mathbf{X} = \mathbf{x}|Y = k)P(Y = k)$$

and so we simply need to find k that optimizes the numerator (the joint distribution evaluated at $\mathbf{x}$ and k). Also, if we further assume, as we often do, that all classes are equally likely then

$$P(Y = k|\mathbf{X} = \mathbf{x}) \propto P(\mathbf{X} = \mathbf{x}|Y = k)$$

and the optimal decision rule reduces to evaluating the K class conditional distributions at $\mathbf{x}$ and finding the largest one!

And, finally, of course it might be easier take a log transformation to simplify the computations, i.e. build the decision rule using

$$\underset{k}{\text{argmax}} \log P(\mathbf{X} = \mathbf{x}|Y = k).$$