STAT 425: Introduction to Nonparametric Statistics **Winter 2018**

Lecture 12: Classification: Density Estimation and Regression Approach

Instructor: Yen-Chi Chen

12.1 Introduction

Classification is one of the most important data analysis problems. Much early work on this topic was done by statisticians but in the past 20 years, computer science and machine learning communities have made much much more progress on this topic.

Here are some classical applications of classification.

- **Email spam.** Email service provider such as Google often faced the problem of classifying a new email. The problem they want to address is like: given an email, how do you decide if it is a spam, an ordinary email, or an important one?
- **Image classification.** If you have used Facebook, you may notice that whenever your photo contains a picture of one of your friends, Facebook may ask you if you want to tag your friend, even if you did not manually tell the computer that this is your friend. How do they know that picture is a human and that guy is your friend?

We consider a simple scenario – binary classification. Namely, there are only two possible classes that we will consider. We will just denote the two classes as 0 and 1.

The classification problem can be formalized as follows. Given a feature vector x_0, we want to create a **classifier** c that maps x_0 into 0 or 1. Namely, we want to find a function $c(x_0)$ that outputs only two possible number: 0 and 1. Moreover, we want to make sure that our *classification error* is small. Let y_0 be the actual label of x_0. We measure the classification error using a **loss function** L such that the the loss of making a prediction $c(x_0)$ when the actual class label is y_0 is $L(c(x_0), y_0)$. A common loss function is the **0-1 loss**, which is $L(c(x_0), y_0) = I(c(x_0) \neq y_0)$. Namely, when we make a wrong classification, we loss 1 point and we do not lose anything if we make the correct classification.

How do we find the classifier c? A good news is: often we have a labeled sample (data) $(X_1, Y_1), \cdots, (X_n, Y_n)$ available. Then we will find c using this dataset.

In statistics, we often model the data as an IID random sample from a distribution. We now define several useful distribution functions:

$$\begin{aligned}
p_0(x) = p(X = x|Y = 0) &: \quad \text{the density of } X \text{ when the actual label is } 0,\\
p_1(x) = p(X = x|Y = 1) &: \quad \text{the density of } X \text{ when the actual label is } 1,\\
P(y|x) = P(Y = y|X = x) &: \quad \text{the probability of being in the class } y \text{ when the feature is } x,\\
P_Y(y) = P(Y = y) &: \quad \text{the probability of observing the class } y\text{, regardless of the feature value.}
\end{aligned} \tag{12.1}$$

Using a probability model, we will define the **risk function**, which is the expected value of the loss function when the input is random. The risk of a classifier c is

$$R(c) = \mathbb{E}(L(c(X), Y)).$$

Ideally, we want to find a classifier that minimizes the risk because such a classifier will minimize our *expected losses.*

Assume that we know the 4 quantities in equation (12.1), what class label will you predict when seeing a feature $X = x$? An intuitive choice is that we should predict the value y that maximizer $P(y|x)$. Namely, we predict the label using the one with highest probability. Such classifier can be written as

$$c_*(x) = \mathsf{argmax}_{y=0,1} P(y|x) = \begin{cases} 0, & \text{if } P(0|x) \geq P(1|x), \\ 1, & \text{if } P(1|x) > P(0|x). \end{cases} \tag{12.2}$$

Is this classifier good in the sense of the classification error (risk)? The answer depends on the loss function. A good news is: this classifier is the optimal classifier for the $0-1$ loss. Namely,

$$R(c_*) = \min_c R(c)$$

when using a $0-1$ loss. However, if we are using other loss function, this classifier will not be the best one (with the smallest expected loss).

Derivation of c_* is optimal under $0-1$ loss. Given a classifier c, the risk function $R(c) = \mathbb{E}(L(c(X), Y))$. Using the property of expectation, we can further write it as

$$R(c) = \mathbb{E}(L(c(X), Y)) = \mathbb{E}(\underbrace{\mathbb{E}(L(c(X), Y)|X)}_{(A)}).$$

For the quantity (A), we have

$$\begin{aligned} \mathbb{E}(L(c(X), Y)|X) &= L(c(X), 1)p(Y = 1|X) + L(c(X), 0)p(Y = 0|X) \\ &= I(c(X) \neq 1)p(Y = 1|X) + I(c(X) \neq 0)p(Y = 0|X) \\ &= \begin{cases} p(Y = 1|X) & \text{if } c(X) = 0 \\ p(Y = 0|X) & \text{if } c(X) = 1. \end{cases} \end{aligned}$$

Thus, seeing a feature X, the expected loss we have when predicting $c(X) = 0$ is $P(Y = 1|X)$ whereas when prediction $c(X) = 1$ is $P(Y = 0|X)$. The optimal choice is predicting $c(X) = 0$ if $P(Y = 1|X) \leq P(Y = 0|X)$ and $c(X) = 1$ if $P(Y = 1|X) > P(Y = 0|X)$ (the equality does not matter), which is the classifier c_*.

When a classifier attains the optimal risk (i.e., having a risk of $\min_c R(c)$), it is called a **Bayes classifier**. Thus, the classifier c_* is the Bayes classifier in $0-1$ loss.

For a classifier c, we define its **excess risk** as

$$\mathcal{E}(c) = R(c) - \min_c R(c).$$

The excess risk is a quantity that measures how the quality of c is away from the optimal/Bayes classifier. If we cannot find the Bayes classifier, we will at least try to find a classifier whose excess risk is small.

12.2 Regression Approach

If we know the $P(y|x)$, we can build the Bayes classifier and this classifier is the optimal one in terms of the risk function. However, $P(y|x)$ is a population quantity, which is often unknown to us. All we have is a random sample $(X_1, Y_1), \cdots, (X_n, Y_n)$. So the question becomes: how do we estimate $P(y|x)$ using the data?

It turns out that this is a problem we know have to solve. Here is just one hint: because the response variable Y only takes two possible values $\{0, 1\}$, it is actually a Bernoulli random variable! Thus, $\mathbb{E}(Y) = P(Y = 1)$, which implies

$$\mathbb{E}(Y|X = x) = P(Y = 1|X = x) = P(1|x). \tag{12.3}$$

Namely, $P(1|x)$ is the regression function! Using the fact that $P(0|x) + P(1|x) = 1$, an estimator of $P(1|x)$ leads to an estimator of $P(y|x)$ for both $y = 0$ and $y = 1$.

Thus, as long as we have a regression estimator, we can convert it into a classifier. Here is one example of using kernel regression. Let

$$\widehat{m}_K(x) = \frac{\sum_{i=1}^n Y_i K\left(\frac{X_i - x}{h}\right)}{\sum_{i=1}^n K\left(\frac{X_i - x}{h}\right)}$$

be the kernel regression. Then

$$\widehat{P}_K(1|x) = \widehat{m}_K(x), \quad \widehat{P}_K(0|x) = 1 - \widehat{m}_K(x).$$

Thus, a classifier based on kernel regression is

$$\begin{aligned}
\widehat{c}_K(x) &= \begin{cases} 0, & \text{if } \widehat{P}_K(0|x) \geq \widehat{P}_K(1|x), \\ 1, & \text{if } \widehat{P}_K(1|x) > \widehat{P}_K(0|x) \end{cases} \\
&= \begin{cases} 0, & \text{if } 1 - \widehat{P}_K(1|x) \geq \widehat{P}_K(1|x), \\ 1, & \text{if } \widehat{P}_K(1|x) > 1 - \widehat{P}_K(1|x). \end{cases} \\
&= \begin{cases} 0, & \text{if } \widehat{P}_K(1|x) \leq \frac{1}{2}, \\ 1, & \text{if } \widehat{P}_K(1|x) > \frac{1}{2}. \end{cases} \\
&= \begin{cases} 0, & \text{if } \widehat{m}_K(x) \leq \frac{1}{2}, \\ 1, & \text{if } \widehat{m}_K(x) > \frac{1}{2}. \end{cases}
\end{aligned}$$

Namely, the classifier will output 1 whenever the estimated regression function is greater than half and 0 otherwise.

Will this classifier be a good one? Intuitively, it should be true. If we have a good regression estimator, the corresponding classifier should also be good. In fact, we have the following powerful result linking the quality of a regression estimator and the excess risk.

Theorem 12.1 *Assume we use the* $0-1$ *loss. Let* $\widehat{m}$ *be a regression estimator and* $\widehat{c}_m$ *be the corresponding classifier. Then*

$$\mathcal{E}(\widehat{c}_m) \leq 2\int |\widehat{m}(x) - m(x)| dP(x) \leq 2\sqrt{\int |\widehat{m}(x) - m(x)|^2 dP(x)}.$$

Namely, if we have a regression estimator whose overall quality is good, the corresponding classifier will also have a small excess risk (i.e., perform comparably well compared to the optimal classifier).

12.3 Density Estimation Approach

In addition to using a regression function to construct a classifier, we can use a density estimator for classification.

A key insight is from the Bayes rule:

$$P(y|x) = P(Y = y|X = x) = \frac{p(x,y)}{p(x)} = \frac{p(x|y)P(Y = y)}{p(x)} = \frac{p_y(x)P_Y(y)}{p(x)},$$

where $p(x) = \sum_y p(x,y) = p(x,0) + p(x,1) = p_0(x)P_Y(0) + p_1(x)P_Y(1)$. Thus, the Bayes classifier can be written as

$$\begin{aligned} c_*(x) &= \begin{cases} 0, & \text{if } P(0|x) \geq P(1|x) \\ 1, & \text{if } P(1|x) > P(0|x) \end{cases} \\ &= \begin{cases} 0, & \text{if } \frac{p_0(x)P_Y(0)}{p(x)} \geq \frac{p_1(x)P_Y(1)}{p(x)} \\ 1, & \text{if } \frac{p_1(x)P_Y(1)}{p(x)} > \frac{p_0(x)P_Y(0)}{p(x)} \end{cases} \\ &= \begin{cases} 0, & \text{if } p_0(x)P_Y(0) \geq p_1(x)P_Y(1) \\ 1, & \text{if } p_1(x)P_Y(1) > p_0(x)P_Y(0) \end{cases}. \end{aligned}$$

Thus, if we can estimate $p_0(x), p_1(x)$, and $P_Y(y)$, we can construct a classifier.

$P_Y(y)$ is very easy to estimate. It is the probability of seeing an observation with label y. As a result, a simple estimator is to use the ratio of observations with this label. Namely,

$$\widehat{P}_Y(y) = \frac{1}{n}\sum_{i=1}^n I(Y_i = y).$$

$p_y(x)$ is just the conditional density of X given the label being y. Thus, we can simply apply a density estimator to those observations with a class label y.

Example: kernel density estimator. Using a kernel density estimator (KDE), we obtain

$$\widehat{p}_{y,kde}(x) = \frac{1}{n_y h}\sum_{i=1}^n I(Y_i = y)K\left(\frac{X_i - x}{h}\right),$$

where $n_y = \sum_{i=1}^n I(Y_i = y)$ is the number of observations with label being y. Note that $\widehat{P}_Y(y) = \frac{n_y}{n}$. Thus, a classifier based on a KDE is

$$\begin{aligned} \widehat{c}_{\mathsf{KDE}}(x) &= \begin{cases} 0, & \text{if } \widehat{p}_{0,kde}(x)\widehat{P}_Y(0) \geq \widehat{p}_{1,kde}(x)\widehat{P}_Y(1) \\ 1, & \text{if } \widehat{p}_{1,kde}(x)\widehat{P}_Y(1) > \widehat{p}_{0,kde}(x)\widehat{P}_Y(0) \end{cases} \\ &= \begin{cases} 0, & \text{if } \sum_{i=1}^n I(Y_i = 0)K\left(\frac{X_i - x}{h}\right) \geq \sum_{i=1}^n I(Y_i = 1)K\left(\frac{X_i - x}{h}\right) \\ 1, & \text{if } \sum_{i=1}^n I(Y_i = 1)K\left(\frac{X_i - x}{h}\right) > \sum_{i=1}^n I(Y_i = 0)K\left(\frac{X_i - x}{h}\right) \end{cases}. \end{aligned}$$

The classifier $\widehat{c}_{\mathsf{KDE}}(x)$ is also called the kernel classifier.

Example: density basis approach. We can use the basis approach as well. Assume that we consider M basis and we use the cosine basis $\{\phi_1(x), \phi_2(x), \cdots\}$. The estimator of $p_y(x)$ will be the density estimator using only those observations with a label y. Let

$$\widehat{\theta}_{y,\ell} = \frac{1}{n_y}\sum_{i=1}^n I(Y_i = y)\phi_\ell(X_i)$$

be the estimator of the ℓ-th coefficient using only those observations with a label y. The corresponding density estimator is

$$\widehat{p}_{y,M}(x) = \sum_{\ell=1}^M \widehat{\theta}_{y,\ell}\phi_\ell(x) = \sum_{\ell=1}^M \frac{1}{n_y}\sum_{i=1}^n I(Y_i = y)\phi_\ell(X_i)\phi_\ell(x) = \frac{1}{n_y}\sum_{i=1}^n I(Y_i = y)\sum_{\ell=1}^M \phi_\ell(X_i)\phi_\ell(x).$$

Thus, the corresponding classifier is

$$\widehat{c}_{\mathsf{M}}(x) = \begin{cases} 0, & \text{if } \widehat{p}_{0,M}(x)\widehat{P}_Y(0) \geq \widehat{p}_{1,M}(x)\widehat{P}_Y(1) \\ 1, & \text{if } \widehat{p}_{1,M}(x)\widehat{P}_Y(1) > \widehat{p}_{0,M}(x)\widehat{P}_Y(0) \end{cases}$$

$$= \begin{cases} 0, & \text{if } \sum_{i=1}^n I(Y_i = 0)\sum_{\ell=1}^M \phi_\ell(X_i)\phi_\ell(x) \geq \sum_{i=1}^n I(Y_i = 1)\sum_{\ell=1}^M \phi_\ell(X_i)\phi_\ell(x) \\ 1, & \text{if } \sum_{i=1}^n I(Y_i = 1)\sum_{\ell=1}^M \phi_\ell(X_i)\phi_\ell(x) > \sum_{i=1}^n I(Y_i = 0)\sum_{\ell=1}^M \phi_\ell(X_i)\phi_\ell(x) \end{cases}.$$

12.4 Confusion Matrix

Given a classifier and a set of labeled data, we can illustrate the quality of classification using a confusion matrix. In binary classification, a confusion matrix is a 2×2 matrix (you can view it as a contingency table) as follows:

	Actual label: 0	Actual label: 1
Predicted label: 0	n_{00}	n_{01}
Predicted label: 1	n_{10}	n_{11}

n_{ij} is the number of instances/observations where the predicted label is i and actual label is j.

The quantity

$$\frac{n_{10} + n_{01}}{n_{00} + n_{01} + n_{10} + n_{11}},$$

is called the misclassification rate and is an empirical estimate of the risk of the classifier.

If the class label 0 stands for ‘normal case’ while the label 1 stands for ‘anomaly’, then we can interpret the confusion matrix as

	Actual label: 0	Actual label: 1
Predicted label: 0	True negative	False negtaive
Predicted label: 1	False positive	True positive

This interpretation is commonly used in engineering problem and medical research for detecting abnormal situation.