MICROECONOMETRICS

Bayesian Classifier and Decision Theory

WISE, XMU

Haihua Xie

27720181153991

Statistics

Oct. 20, 2019

1 Introduction

In this project, I'm going to introduce Bayesian classifier and decision theory. The bayesian decision theory will be introduced first, then the bayesian classifier, there are a large number of bayesian classifiers, we only cover the simple one: naive beysian classifier.

2 Bayesian decision theory

2.1 Bayes estimator for common loss functions

To make a decision, we have to choose an action a from some action space $\mathcal{A}$. Finally we incur some loss, $L(y, a)$, which measures how compatible our action a is with nature' s hidden state y. Our goal is to devise a decision procedure or policy, $\delta : \mathcal{X} \to \mathcal{A}$, which specifies the optimal action for each possible input. By optimal, we mean the action that minimizes the expected loss: $\delta(\mathbf{x}) = \underset{a \in \mathcal{A}}{argmin} \mathbb{E}[L(y, a)]$.

In the Bayesian approach to decision theory, the optimal action, having observed x , is defined as the action a that minimizes the posterior expected loss : $\delta(\mathbf{x}) = \arg\min_{a \in \mathcal{A}} \rho(\mathbf{a}|\mathbf{x})$.

The posterior expected loss: $\rho(a|\mathbf{x}) = \mathbb{E}_{p(y|\mathbf{x})}[L(y, a)] = \sum_y L(y, a)p(y|\mathbf{x})$. In the Bayesian version, we mean the expected value of y given the data we have seen so far. In the frequentist version, we mean the expected value of y and x that we expect to see in the future.

- **MAP estimate minimizes 0-1 loss**

The 0-1 loss is defined by:

$$L(y, a) = \mathbb{I}(y \neq a) = \begin{cases} 0 & \text{if } a = y \\ 1 & \text{if } a \neq y \end{cases}$$

The posterior expected loss is: $\rho(a|\mathbf{x}) = p(a \neq y|\mathbf{x}) = 1 - p(y|\mathbf{x})$. Hence the action that minimizes the expected loss is the MAP estimate: $y^*(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} p(y|\mathbf{x})$.

- **Posterior mean estimate minimizes ℓ_2 (quadratic)loss**

For continuous parameters, a more appropriate loss function is squared error loss, ℓ_2 loss ,or quadratic loss, defined as:$L(y, a) = (y - a)^2$. The posterior expected loss is given by:

$$\rho(a|\mathbf{x}) = \mathbb{E}\left[(y-a)^2|\mathbf{x}\right] = \mathbb{E}\left[y^2|\mathbf{x}\right] - 2a\mathbb{E}[y|\mathbf{x}] + a^2$$
$$\hat{y} = \mathbb{E}[y|\mathbf{x}] = \int yp(y|\mathbf{x})dy$$

- **Posterior median minimizes ℓ_1 (absolute) loss**

The ℓ_1 (absolute) loss is defined by $L(y,a) = |y-a|$. The optimal estimate is the posterior median,i.e., a value of a such that$P(y < a|\mathbf{x}) = P(y \geq a|\mathbf{x}) = 0.5$.

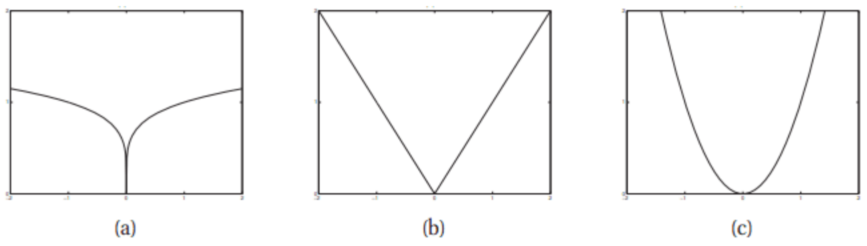


Figure1: Graph of $L(y,a) = |y-a|^q$, for $q = 0.2$, $q = 1$ and $q = 2$

The ℓ_2 loss penalizes deviations from the truth quadratically, and thus is sensitive to outliers. A more robust alternative is the absolute or ℓ_1 loss.

- **Supervised learning**

Consider a prediction function $\delta : \mathcal{X} \to \mathcal{Y}$, and suppose we have some cost function $\ell(y, y')$, which gives the cost of predicting y' when the truth is y. The loss function is defined by

$$L(\boldsymbol{\theta}, \delta) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y|\boldsymbol{\theta})}\left[\ell(y, \delta(\mathbf{x})\right] = \sum_{\mathbf{x}}\sum_{y} L(y, \delta(\mathbf{x}))p(\mathbf{x}, y|\boldsymbol{\theta})$$

This is known as the generalization error . Our goal is to minimize the posterior expected loss, given by:

$$\rho(\delta|\mathcal{D}) = \int p(\boldsymbol{\theta}|\mathcal{D})L(\boldsymbol{\theta}, \delta)d\boldsymbol{\theta}$$

2.2 The false positive vs false negative tradeoff

Binary decision problem, such as hypothesis testing, two-class classification, object/ event detection, etc.

The confusion matrix:

Table 1: Confusion matrix

	$y=1$	$y=0$	Sum
$\hat{y}=1$	TP	FP	$\hat{N}_+$
$\hat{y}=0$	FN	TN	$\hat{N}_-$
Sum	N_+	N_-	N

A general loss matrix

Table 2: Loss matrix

	$y=1$	$y=0$
$\hat{y}=1$	0	L_{FP}
$\hat{y}=0$	L_{FN}	0

where L_{FN} is the cost of a false negative, and L_{FP} is the cost of a false positive. The posterior expected losses for the two possible actions are given by

$$\rho(\hat{y}=0|\mathbf{x}) = L_{FN}\, p(y=1|\mathbf{x}) \text{ and } \rho(\hat{y}=1|\mathbf{x}) = L_{FP}\, p(y=0|\mathbf{x})$$

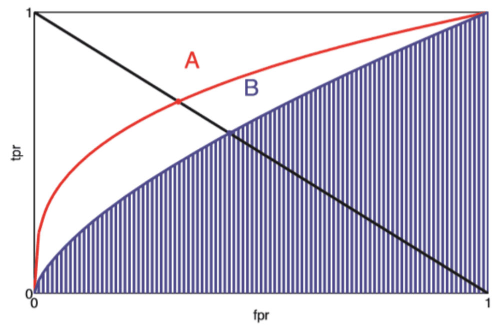
- **ROC curves and Precision-recall curves**

Accuracy: $\frac{TF+TN}{N}$;
Precision: $\frac{TP}{TP+FP}$;
Recall/sensitivity: $\frac{TP}{TP+FN} = \frac{TP}{N_+}$;
Specificity: $\frac{TN}{TN+FP} = \frac{TN}{N_-}$.

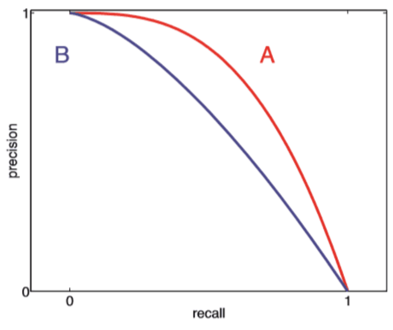
Table 3: Quantities derived from the confusion matrix

	$y=1$	$y=0$
$\hat{y}=1$	TPR=sensitivity=recall	FPR=type I error rate
$\hat{y}=0$	FNR=type II error rate	TNR=specificity

Rather than than computing the TPR and FPR for a fixed threshold τ , we can run our detector for a set of thresholds, and then plot the TPR vs FPR as an implicit function of τ. This is called a receiver operating characteristic or ROC curve.The quality of a ROC curve is often summarized as a single number using the area under the curve or AUC . Higher AUC scores are better; the maximum is obviously 1.



(a)



(b)

Figure2: ROC and Precision-Recall curves

(a) ROC curves for two hypothetical classification systems. A is better than B. We plot the true positive rate (TPR) vs the false positive rate (FPR) as we vary the threshold τ . We also indicate the equal error rate (EER) with the red and blue dots, and the area under the curve (AUC) for classifier B.

(b) A precision-recall curve for two hypothetical classification systems. A is better than B in this case.

- **F-scores**

For a fixed threshold, one can compute a single precision and recall value. These are often combined into a single statistic called the F score, orF1 score, which is the harmonic mean of precision and recall: estimate the precision and recall: $P = \frac{\sum_i y_i \hat{y}_i}{\sum_i \hat{y}_i}$, $R = \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i}$; then F-score or F1-score:

$$F_1 = \frac{2}{1/P + 1/R} = \frac{2PR}{R+P} = \frac{2\sum_{i=1}^N y_i \hat{y}_i}{\sum_{i=1}^N y_i + \sum_{i=1}^N \hat{y}_i}$$

To understand why we use the harmonic mean instead of the arithmetic mean, $(P + R)/2$, consider the following scenario: suppose we recall all entries, so $R = 1$, it's passible to get a very low precision, say, $p(y = 1) = 10^{-5}$, then the arithmetic mean is approximately 50%, while the harmonic mean of this strategy is approximately only 0.2%.

Macro-averaged F1, $\sum_{c=1}^C F_1(c)/C$ where $F_1(C)$ is the F1 score obtained on the task of distinguishing class c from all the others.

Micro-averaged F1, which is defined as the F1 score where we pool all the counts from each class’ s contingency table.

Example:

Table 4: An example: macro-averaged vs. micro-averaged F1

Class 1			Class2			Pooled		
	$y=1$	$y=0$		$y=1$	$y=0$		$y=1$	$y=0$
$\hat{y}=1$	10	10	$\hat{y}=1$	90	10	$\hat{y}=1$	100	20
$\hat{y}=0$	10	970	$\hat{y}=0$	10	890	$\hat{y}=0$	20	1860

We see that the precision of class 1 is 0.5, and of class 2 is 0.9. The macro-averaged precision is therefore 0.7, whereas the micro-averaged precision is $100/(100+20) \approx 0.83$.

- **False discovery rates**

Suppose we are trying to discover a rare phenomenon using some kind of high throughput measurement device, such as a gene expression micro array, or a radio telescope.In such multiple hypotheses testing, we need to make many binary decision of the form $p(y_i = 1|\mathcal{D}) > \tau$, where $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, we are classifying y_i based on all data, not just based on x_i , it's a simultaneous classification problem.

We are trying to minimize the expected number of the false positive to find an appropriate threshold τ.

$$FD(\tau, \mathcal{D}) = \sum_i (1 - p_i)\,\mathbb{I}\,(p_i > \tau)$$

$$FDR(\tau, \mathcal{D}) = FD(\tau, \mathcal{D})/N(\tau, \mathcal{D})$$

Where $p_i = p\,(y_i = 1|\mathcal{D})$ is your belief that this object exhibits the phenomenon in question. $N(\tau, \mathcal{D}) = \sum_i \mathbb{I}\,(p_i > \tau)$ is the number of discovered items. Given a desired FDR tolerance, say $\alpha = 0.05$, one can then adapt τ to achieve this, this is called the direct posterior probability approach to controlling the FDR. There are still a lot of researchers studying in the multiple hypotheses testing by controlling FDR, especially in the health and medicine.

3 Bayesian Classifier

Actually, the bayesian classifier is an application of the bayesian decision theory in solving some classification problems, I mainly introduce the naive bayes classifier, which is relatively simple and easy to understand. More complicated semi-naive bayes classifier and bayes network is left to be further reading.

3.1 Naive bayes classifier

Our goal is to classify vectors of discrete-valued features, $\mathbf{x} \in 1, 2, ..., K^D$, where K is the number of values for each feature, and D is the number of features.

Assumption: the features are conditionally independent given the class label. This allows us to write the class conditional density as a product of one dimensional densities:

$$p(\mathbf{x}|y = c, \theta) = \prod_{j=1}^{D} p(x_j|y = c, \theta_{jc})$$

The resulting model is called a **naive bayes classifier**.

Gaussian naive bayes classifier. In the case of real-valued features, we can use the Gaussian distribution: $p(\mathbf{x}|y = c, \theta) = \prod_{j=1}^{D} \mathcal{N}(x_j|\mu_{jc}, \sigma_{jc}^2)$, where μ_{jc} is the mean of feature j in objects of class c, and $\sigma_{jc}^2)$ is its variance.

Bernoulli naive bayes classifier. In the case of binary features, $x_j \in 0, 1$, we can use the Bernoulli distribution: $p(\mathbf{x}|y = c, \theta) = \prod_{j=1}^{D} Ber(x_j|\mu_{jc})$, where the μ_{jc} is the probability that feature j occurs in class c. This is sometimes called the multivariate Bernoulli naive Bayes model.

Categorical naive bayes classifier. In the case of categorical feature, $x_j \in 1, ..., K$, we can use the multinoulli distribution: $p(\mathbf{x}|y = c, \theta) = \prod_{j=1}^{D} Cat(x_j|\mu_{jc})$, where the μ_{jc} is the histogram over the K possble values for x_j in class c.

3.2 Model fitting

To train a naive bayes classifier. This usually means computing the MLE or the MAP estimates of the parameters. The probability of single data case is given by:

$$p(\mathbf{x}_i, y_i|\boldsymbol{\theta}) = p(y_i|\boldsymbol{\pi}) \prod_j p(x_{ij}|\boldsymbol{\theta}_j) = \prod_c \pi_c^{\mathbb{I}(y_i=c)} \prod_j p(x_{ij}|\boldsymbol{\theta}_{jc})^{\mathbb{I}(y_i=c)}$$

Where π is the class prior. Hence, the likelihood is given by

$$\log p(\mathbf{x}_i, y_i|\boldsymbol{\theta}) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log p(x_{ij}|\boldsymbol{\theta}_{jc})$$

We see that this expression decomposes into a series of terms, one only concerns $\boldsymbol{\pi}$, and the other temr contains $\boldsymbol{\theta}_{jc}$'s, then we can obtain the MLE for the calss prior: $\hat{\pi}_c = \frac{N_c}{N}$, where the $N_c = \sum_i \mathbb{I}(y_i = c)$ is the number of examples in class c.

The MLE for the likelihood depends on the type of distribution we choose for each feature. For simplicity, suppose all features are binary, and $x_j|y = c \sim Ber(\theta_{jc})$, in this case, $\hat{\theta_{jc}} = \frac{N_{jc}}{N_c}$.