Machine Learning Report: Supervised Learning

Cindy Cai(18120340), Jane Cao(18120341)
Will Miao(18125242), Zakaria Hasan(18129020), Zper Zhang(18120448)

1 Abstract

Give a brief introduction about this report. (Please delete all of the red text in your final submitted version!) I will write this section at last.

2 Preliminaries

Supervised learning is the machine learning task of learning a function that maps an input to an output based on sample input-output pairs.It infers a function from labeled training data consisting of a set of training samples, which can be used for mapping new samples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.

In Table 1,We introduce some notations in this paper. Obviously, a supervised learning algorithm seeks a function:$g : \mathcal{X} \to \mathcal{Y}$ for mapping new samples.

Table 1: Notations used in the paper

Symbols	Description
$\mathcal{X}$	d-dimensional feature space R^d
$\mathcal{Y}$	label space with q labels $\{1, 2, ..., q\}$
$\mathcal{D}$	training set with n samples $\{(x_i, y_i)\|1 \leq i \leq n\}$
x_i	$x_i \in \mathcal{X}$ is a d-dimensional feature vector$(x_{i1}, x_{i2}, ..., x_{id})^T$
y_i	$y_i \in \mathcal{Y}$ is the label of x_i

3 Algorithm Description

In this section, we will give the detailed description of supervised learning algorithms (including Decision tree, Naive Bayes, Logistic regression, Neural network, Support vector machines).

3.1 Decision Tree

3.2 Naive Bayes

Naive bayes is a probabilistic classifier based on Bayes' theorem. It assumes all features are independent.

For binary classification problem, we need to compare the probability of two different classes.

$$P(y = 1 \mid X) = \frac{P(X \mid y = 1)P(y = 1)}{P(X)} \tag{1}$$

$$P(y = 0 \mid X) = \frac{P(X \mid y = 0)P(y = 0)}{P(X)} \tag{2}$$

If the probability of given X, y equals 1 is larger than given X, y equals 0, then we will say the sample we gave to our classifier is more likely to be class one.

$$P(y=1) = \frac{\#\ of\ samples\ belong\ to\ class\ 1}{\#\ of\ all\ training\ samples} \tag{3}$$

$$P(y=0) = \frac{\#\ of\ samples\ belong\ to\ class\ 0}{\#\ of\ all\ training\ samples} \tag{4}$$

We can use the formula above to compute $P(y=1)$ and $P(y=0)$. Since we assume that all features are independent, so we can use the formula below to compute the probability $P(X \mid y=1)$ and $P(X \mid y=0)$.

$$P(X \mid y=1) = \prod_{i=1}^{m} P(X_i \mid y=1) \tag{5}$$

m denotes the number of features. For discrete features, we can use frequency to compute $P(X_i \mid y=1)$ of each features, this method is also called Bernoulli Naive Bayes. But for continuous features, we cannot use frequency, because the probability of $P(X_i \mid y=1)$ would be very small or even be zero. We need some other techniques to deal with this problem. We assume that all features are normal distributed, so we can choose one feature and use all samples of this feature to compute the parameters of the normal distribution of this feature, such as μ and σ^2. This method is called Gaussian Naive Bayes.

Actually, we can use MLE(Maximum likelihood estimation) to estimate the parameters of the normal distribution, and then we can get the formula of the following:

$$\mu_j = \frac{1}{k}\sum_{i=1}^{k} X_j^{(i)} \tag{6}$$

$$\sigma_j^2 = \frac{1}{k}\sum_{i=1}^{k}(X_j^{(i)} - \mu_j)^2 \tag{7}$$

In the formula above, j denotes the jth feature of samples, k denotes the number of samples which belong to class 1 or class 0.

After computing the value of μ and σ^2, we can use these parameters to compute the $P(X_j \mid y=1)$, the probability of the jth feature given y equals one.

$$P(X_j \mid y=1) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp^{(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2})} \tag{8}$$

After computing all probability of each features given the label, we can times them up and times the probability of the label. So we can get two different probabilities of two different labels, and compare them to determine which label the sample should be.

3.3 Logistic Regression

3.4 Neural Network

3.5 Support Vector Machines

4 Experiments setting

4.1 Dataset

In the experiments, two UCI datasets were used. The detailed information about the datasets are listed in Table 2.

Table 2: Detailed information about datasets used in this paper

Dataset	Number of instances	Number of attributes	Number of classes
Iris	150	4	3
Statlog(Heart)	270	13	2

4.2 Data preprocessing

Give the detailed information about data processing.

4.3 Evaluation metrics

Different evaluation measures are available when computing evaluation scores for classication.

Table 3: Confusion Matrix in Binary Scenario

	True P	True N
Predicted P	TP	FP
Predicted N	FN	TN

In binary scenario, given a Table 3, one can compute:

$$precision = \frac{TP}{TP + FP} \tag{9}$$

$$recall = \frac{TP}{TP + FN} \tag{10}$$

$$f1_score(\beta) = \frac{2 * TP}{2 * TP + FP + FN} \tag{11}$$

One can compute the accuarcy for class A:

$$accuracy = \frac{TP}{N_A} \tag{12}$$

In which TP = true positive for class A, N_A = the total number of instances of class A in test.

When multiple class labels are to be retrieved, averaging the evaluation measures can give a view on the general results. There are two names to refer to averaged results: micro-averaged and macro-averaged results. $L = \{\lambda_j | j = 1, 2, ..., q\}$ is the set of all labels. Consider a binary evaluation measure

$B(TP, TN, FP, FN)$ that is calculated based on the number of true positives, true negatives, false positives and false negatives. Let TP_λ, FP_λ, TN_λ and FN_λ be the number of true positives, false positives, true negatives and false negatives after binary evaluation for a label Λ.

A macro-averaged results can be computed as follows:

$$B_{macro} = \frac{1}{q} \sum_{\Lambda=1}^{q} B\left(TP_\lambda, FP_\lambda, FP_\lambda, TN_\lambda\right) \tag{13}$$

A micro-averaged results can be computed as follows:

$$B_{micro} = B\left(\sum_{\Lambda=1}^{q} TP_\lambda, \sum_{\Lambda=1}^{q} FP_\lambda, \sum_{\Lambda=1}^{q} FP_\lambda, \sum_{\Lambda=1}^{q} TN_\lambda\right) \tag{14}$$

4.4 Parameter Setting

Give the detailed information about how to set optimal parameters for each algorithm.

5 Experimental Results

5.1 Effect of parameters

Investigate the effect of parameters (i.e., the number of nearest neighbors in KNN) for classification performance and give a deep discussion.

5.2 Classification performance

List the classification performance (different evaluation metrics) in difference algorithms and analyze the experimental results.

Table 4: Results on Statlog(Heart)

method	precision	recall	f1_score	accuracy
Decision Tree	0.0	0.0	0.0	0.0
Naive Bayes	0.0	0.0	0.0	0.0
Logistic Regression	0.0	0.0	0.0	0.0
Neural Network	0.0	0.0	0.0	0.0
SVM	0.0	0.0	0.0	0.0

analyze....

Table 5: Macro-Results on Iris

method	macro_precision	macro_recall	macro_f1_score	macro_accuracy
Decision Tree	0.0	0.0	0.0	0.0
Naive Bayes	0.0	0.0	0.0	0.0
Logistic Regression	0.0	0.0	0.0	0.0
Neural Network	0.0	0.0	0.0	0.0
SVM	0.0	0.0	0.0	0.0

analyze....
analyze....

Table 6: Micro-Results on Iris

method	micro_precision	micro_recall	micro_f1_score	micro_accuracy
Decision Tree	0.0	0.0	0.0	0.0
Naive Bayes	0.0	0.0	0.0	0.0
Logistic Regression	0.0	0.0	0.0	0.0
Neural Network	0.0	0.0	0.0	0.0
SVM	0.0	0.0	0.0	0.0

6 Problems and Solutions

List the problems you've ever met and the corresponding solutions

7 Conclusion

Give a brief conclusion about this report.