Machine Learning Project Report

Abstract

Background: In this report, we performed five classifiers on KDD physics dataset to accomplish the task of labeling class and made comparative analysis on the performance of classifiers.

Methods: Two data reduction techniques were used: singular values decomposition (SVD) and stepwise feature selection. 5-fold cross validation was performed. Final prediction model is based on bagging results of five basic classifiers.

Results: Stepwise selection outperforms SVD in our dataset. Among five basic classifiers, logistic regression returns the lower error rate 0.2909. Neural network perform next best, followed by naïve bayes, which is followed by k-nearest neighbors. After bagging, the error rate is reduced to 0.2855. The prediction accuracy on the testing data is 0.71239 with ACC metric.

1. Introduction

We used KDD 2004 Quantum Physics Dataset [1]. The training part contains totally 50,000 entries with 78 numeric features. The class is set as binary value 0 and 1.

There are some missing values in the feature denoted as special numbers. Missing values: columns 22,23,24 and 46,47,48 use a value of "999" to denote "not available", and columns 31 and 57 use "9999" to denote "not available".

2. Methods

2.1 Outline

We firstly used multiple imputation method to replace the missing data with meaningful values.

Two data reduction methods were performed. One is singular value decomposition (SVD) and another is stepwise feature selection. Comparison of their performance is reported.

To avoid over-fitting of our prediction model, we assessed 5-fold cross validation. The mean of 5-fold cross validation error was calculated and used as the only criteria of our prediction models.

2.3 Description of Classifiers

Artificial Neural Networks

The neural network can be considered as multiple layer perceptron. Each layer in the network is made of neurons and connects with each other via synapses. The previous layer sends data to the node in next node after we have reached the threshold. The learning process is achieved by updating weights of activation function between interconnection. The network is capable of simulating very complex model if we add more hidden nodes and layers in the system. However, determining the optimal number of hidden nodes becomes a question. [2] made experiment on training data with different number and stated that the optimal number of hidden nodes is O(log(n)), where n is the number of training samples. Since we only have at most 80 features, the classification problem can be achieved by using single hidden layer. Thus, we used 15 hidden nodes in one intermediate layer, which is computed based on training sample, as the starting point to explore the optimal model with different hidden nodes number under feature space that we defined in previous step. The result is as Table 1 shows.

It is easy to find that best model has 60 principal components and 5 hidden nodes. As SVD feature number becomes larger, the mean cross validation error on each level of hidden nodes firstly decreases and then increases. It indicates that K with the range 40-60 has best performance. In terms of number of hidden nodes, the model is prone to over fitting after we set model with more hidden nodes. The result agrees on the empirical setting of optimal number of node. We also notice that computation complexity of neural network is higher than other classifiers, in each iteration, we approximately have 1,000 updates on the weight of connection function.

Naïve Bayes Classifier

A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with independence assumption as follows:

$$\text{Bayes' theorem:} \quad P(y \mid x) = \frac{P(x \mid y)P(y)}{\sum_{C} P(x \mid y = c)P(y = c)}$$

$$\text{Independence Assumption:} \quad P(x|y) = \prod_{i} P(x_i \mid y)$$

We applied rule of maximum a posteriori(MAP) for classfication:

$$classify([x_1,\ x_2,\ \ldots,\ x_K]) = \arg\max P(y)P(x \mid y)$$

From training data, we estimate the empirical prior and the likelihood. Then we select the maximum of posterior probability across two categories as the predicted values given input features $x_1,\ x_2,\ \ldots,\ x_K$.

Logistic Regression Model

Logistic regression is a probabilistic model used to predict a binary outcome. Like other regression model, we only need to estimate the coefficients by using maximum

likelihood estimation (MLE). Most time closed form solution is impossible and numerical approximation is performed. If the model doesn't reach convergence, then multicollinearity will appear. $Adjusted\ R^2$ can be used to obtain the goodness of fit.

K-Nearest Neighbors Algorithm (k-NN)

The k-nearest neighbor algorithm is the simplest of all machine learning algorithms. We classify an instance by finding its nearest neighbors and assign it to the class which is most common among those neighbors (majority vote).

The best choice of k depends upon the data; generally, larger values of k reduce the effect of noise on the classification,[3] but make boundaries between classes less distinct.

Decision Tree Learning

Decision tree learning is a method for approximating discrete valued target functions, in which the learned function is represented by a decision tree. Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items.[2] In this study, we chose information gain based on the concept of entropy from information theory.

$$IG(T,a) = H(T) - H(T \mid a),$$

Where H denotes the information entropy

$$H(T) = E[\log \frac{1}{p(x)}] = \sum p_i \log_2 \frac{1}{p_i}$$

2.4 Cross Validation

In 5-fold cross-validation, the training is randomly partitioned into *5* equal size subsamples. Of the *k* subsamples, a single subsample is retained as the validation data for testing the model, and the remaining *k* − 1 subsamples are used as training data. The cross-validation process is then repeated over all folds. The *k* results from the folds can then be averaged to produce a single estimation.

2.5 Bagging

The bagging results are based on majority vote of five basic classifiers. Let $y_{ik} = 0\ or\ 1$ is the prediction values on i^{th} test data from k^{th} classifier. Then bagging prediction

$$z_i = \begin{cases} 1, if \sum_{k=1}^{K} y_{ik} > round(\frac{K}{2}) \\ 0, if \sum_{k=1}^{K} y_{ik} \leq round(\frac{K}{2}) \end{cases}$$

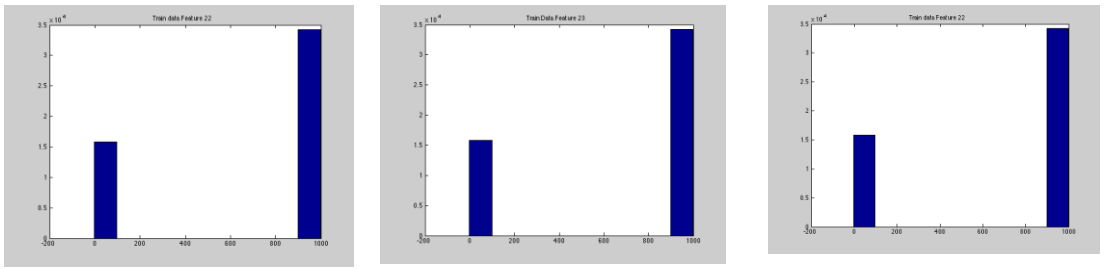
3. Results

3.1 Exploratory Results

Fig.1 shows the missing values completely change the distribution within feature 22, 23, 24, 46, 47 and 48. By the description of the dataset, we can assume the missing columns are missing at random (MAR). The most common way to deal with such missing data is multiple imputations. Here we limit the number of imputation to be 5 times. Fig.2 shows the distribution after we computed the missing values.

Fig.1 Histograms before multiple imputation

Feature 22 Feature 23 Feature 24



Feature 46 Feature 47 Feature 48

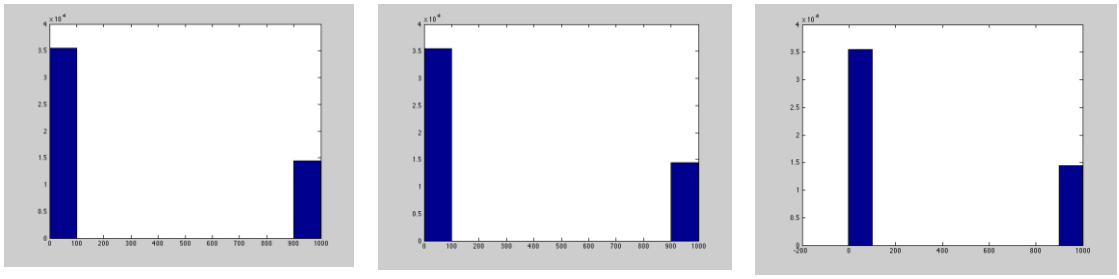
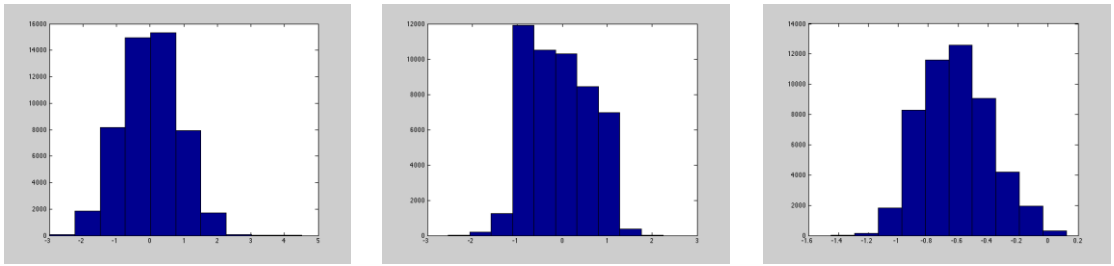
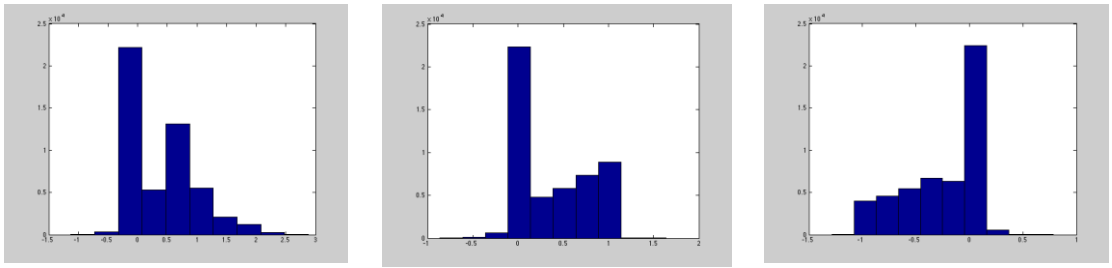


Fig.2 Histograms after multiple imputation

Feature 22 Feature 23 Feature 24



Feature 46 Feature 47 Feature 48



3.2 Classifier Prediction Errors

To evaluate which model is used for prediction on testing data, we used error on validation data. Table 1~5 shows the results with SVD. The best results are 0.3059, 0.3186, 0.2917, 0.2896 and 0.2948 from Naïve Bayes, Decision Tree, Logistic Regression, Neural Network and kNN, respectively.

Table 1. Cross Validation Error (Neural Network Classifier with SVD)

		Number of Hidden Node			
		5	10	20	30
K	10	0.3438	0.3404	0.3422	0.3388
	20	0.3031	0.3027	0.3027	0.303
	30	0.2999	0.3014	0.3007	0.3043
	40	0.2983	0.298	0.2998	0.3056
	50	0.2955	0.2991	0.3027	0.3066
	60	**0.2896**	0.294	0.2992	0.31
	70	0.3048	0.3018	0.3026	0.304

Table 2. Cross Validation Error (Naïve Bayes Classifier with SVD)

K						
10	20	30	40	**50**	60	70
0.3571	0.3091	0.311	0.3076	**0.3059**	0.3102	0.3317

Table 3. Cross Validation Error (Logistic Regression Classifier with SVD)

K						
10	20	30	40	50	60	**70**
0.3543	0.3052	0.3056	0.3002	0.2963	0.2927	**0.2917**

Table 4. Cross Validation Error (K-nearest Neighbors Classifier with SVD)

		Number of Nearest Neighbor						
		5	55	105	**155**	205	255	305
K	10	0.3444	0.3275	0.33	0.3341	0.3357	0.3355	0.33616
	20	0.3352	0.308	0.3054	0.3047	0.3047	0.3041	0.30492
	30	0.3262	0.303	0.302	0.3016	0.3008	0.3014	0.30166
	40	0.3252	0.3034	0.3005	0.2988	0.2996	0.2979	0.2997
	50	0.3275	0.3024	0.2992	0.2987	0.2981	0.2977	0.29864
	60	0.3273	0.3005	0.297	0.296	0.2968	0.2979	0.29766
	70	0.3248	0.2986	0.2972	**0.2948**	0.296	0.2967	0.2978

Table 5. Cross Validation Error (Decision Tree Classifier with SVD)

		Depth					
		5	10	15	20	25	**30**
	10	0.3735	0.3735	0.3615	0.3469	0.3405	0.3356
	20	0.3735	0.3735	0.3457	0.3303	0.3263	**0.3186**
	30	0.3735	0.3735	0.3457	0.3314	0.3272	0.3196
K	40	0.3735	0.3735	0.345	0.3307	0.3268	0.3194
	50	0.3735	0.3735	0.354	0.332	0.3324	0.3256
	60	0.3735	0.3735	0.354	0.332	0.3322	0.3283
	70	0.3735	0.3735	0.354	0.332	0.3322	0.3283

The above results are from SVD features. Now we switch to stepwise features.

Table 6. Cross Validation Error (by Stepwise Selection)

Naïve Bayes	Decision Tree	Logistic Reg	Neural Network	Knn
0.3231	0.3605	0.2909	0.2911	0.2921

3.3 Prediction Results

From above, we should bagged our classifiers with logistic regression (stepwise features), logistic regression (SVD features), KNN (stepwise features), neural network (stepwise features) and neural network (SVD features).

Bagging seems to gives us most reliable prediction results (error rate = 0.2855). Thus we used bagged model for final prediction. Its performance on test data is as Table 7 shows.

Table 7. Prediction on Testing data

	ACC	ROC	CXE	SLQ 0.01	
SVD	0.5568				
Stepwise	0.7124	0.55	5.22	0.17	

4. Conclusions

For our data, stepwise selection is more appropriate. And logistic regression exceeds other methods. With bagging, we average the results from five classifiers, minimizing the pattern recognized by each classifier.

The reason why SVD fails in this case might be the existence of noisy features in the data. SVD will presume them as useful as those "real effort" features. Thus, even we perform SVD, the result is still unsatisfying. But stepwise procedure will consider the interaction between features and use certain cutoff to eliminate the misleading effects.

5. References

[1]http://osmot.cs.cornell.edu/kddcup/datasets.html

[2] Nayer Wanas, Gasser Auda, (1998). On the Optimal Number of Hidden Nodes In A Neural Network, (CCECE'98): 918-921

[3] Rokach, L.; Maimon, O. (2005). "Top-down induction of decision trees classifiers-a survey". *IEEE Transactions on Systems, Man, and Cybernetics, Part C* **35** (4): 476–487.

[4] Everitt, B. S., Landau, S., Leese, M. and Stahl, D. (2011) Miscellaneous Clustering Methods, in Cluster Analysis, 5th Edition, John Wiley & Sons, Ltd, Chichester, UK.