Feature Selection for VIP E-Mail Accounts Analysis

Zhan Zhang, Yingjie Tian^{*}, and Yong Shi

Research Center on Data Technology and Knowledge Economy of CAS,
Beijing 10080,
zhangzhan05@mails.gucas.ac.cn,
tianyingjie1213@163.com,
yshi@gucas.ac.cn,
www.dtke.ac.cn

Abstract. This paper introduces several feature selection methods on the VIP E-Mail account analysis, and then uses SVM to classify the users in order to maintain the ones who may leave in a short time. The websites hosting companies can follow the strategies, which are investigated in this paper, to reduce their labor cost and provide an attractive and simple registration for the users. The result shows that we can also reach the feasible predict accuracy.

Keywords: Support Vector Machine, Feature selection, VIP E-Mail.

1 Introduction

With the development of internet, E-Mail is becoming more and more popular and widely used. Because of its convenience and speediness, it is not only a tool for communication but also has its own commercial characteristics. Many websites hosting companies have developed the charged VIP E-Mail service to meet the demand of the certain persons. According to the statistics, as Chinese network advanced in the past few years, the total market of the Chinese VIP E-Mail service has reached 6.4 hundred million RMB by 2005, which has caused drastic competitions around the IT market. Meanwhile, the providers have to try their best to collect more information about the users and buy more memory to save them. The technical staffs have to face more and more data bases and log files. Therefore, it has become the most important problems for the service providers that how to look for the potential users, how to maintain the current users and how to analysis the behaviors of the users throughout the feature as few as possible.

Data mining means the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. Feature selection is an important part of data mining. The number of features affects the classifier's speed that including a large number of features can result in long training and classification times.

The purpose of this paper is to apply several feature selection methods including Support Vector Machine (SVM), F-score method, and one-norm SVM to reduce the data scale and then use the SVM to deal with the classification on a famous Chinese

^{*} Corresponding author.

website's VIP E-Mail data set. This paper is organized as follows. In Section 2 we introduce the SVM classifier; Section 3 focuses on the different methods that work on the feature selection; Section 4 introduces how to choose satisfying parameters with 10-fold cross validation; In Section 5, we show the experimental results during the development period of the competition; and the last part is the conclusion.

2 Support Vector Classification

The algorithm about SVM is originally established by Vapnik (1998). Since 1990s SVM has been a promising tool for data classification. Its basic idea is to map data into a high dimensional space and find a separating hyper plane with the maximal margin. SVM solves the following optimization problem:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w' w + C \sum_{k=1}^m \xi_k, \\ \text{subject to} \quad & y_k (w^T \phi(x_k) + b) \geq 1 - \xi_k, \\ & \xi_k \geq 0, k = 1, \dots, m, \end{aligned}$$

Where training data are mapped to a higher dimensional space by the function ϕ , and C is a penalty parameter on the error. The decision function (predictor) is $f(x) = \text{sgn}(w^T \phi(x_k) + b)$.

The RBF kernel is used in our experiments: $k(x, x') = \exp(-\gamma \|x - x'\|^2)$. With the RBF kernel, there are two parameters to be determined in the SVM model i.e. C and γ .

3 Feature Selection Strategies

In this section, we will discuss three feature selection strategies respectively: SVM, F-score and one-norm SVM.

3.1 No Feature Selection: Direct SVM Strategy

Firstly, we use the SVM strategy directly without feature selection and choose the linear kernel function.

3.2 SVM Feature Selection Strategy

From 3.1, we can get the decision function (predictor): $f(x) = \text{sgn}(w^T x_k + b)$. If we scale the data by putting each feature to $[0, 1]$, the coefficient $|w|$ can be

considered as the weight of each feature. The larger the lwl is, the more discriminative this feature is. Therefore, we use this score as a feature selection criterion.

3.3 F-Score Feature Selection Strategy[3]

F-score is a simple technique which measures the discrimination of two sets of real numbers. Given training vectors x_k , $k = 1 \dots m$, if the numbers of positive and negative instances are n_+ and n_- , respectively, then the F-score of the i^{th} feature is defined as:

$$F(i) = \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1} \sum_{k=1}^{n_+} (x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n_- - 1} \sum_{k=1}^{n_-} (x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2}$$

Where $\bar{x}_i$, $\bar{x}_i^{(+)}$, $\bar{x}_i^{(-)}$ are the average of the i^{th} feature of the whole, positive, and negative data sets, respectively; $x_{k,i}^{(+)}$ is the i^{th} feature of the k^{th} positive instance, and $x_{k,i}^{(-)}$ is the i^{th} feature of the k^{th} negative instance. The numerator indicates the discrimination between the positive and negative sets, and the denominator indicates the one within each of the two sets. The larger the F-score is, the more discriminative this feature is. Therefore, we use this score as a feature selection criterion.

3.4 One-Norm SVM Feature Selection Strategy

According to SVM, we give the following linear program to calculate the coefficient w :

Given training vectors $x_k \in R^n$, $k = 1 \dots m$, in two classes and a vector of labels $y \in R^m$ such that $y_k \in \{1, -1\}$, and we will solve the following linear program:

$$\begin{aligned} \min_{w, b, \xi, s} \quad & \sum_{i=1}^k s_i + C \sum_j^m \xi_j, \\ \text{subject to} \quad & y_j((w \bullet x_j) + b) \geq 1 - \xi_j, j = 1, \dots, m, \\ & -s_i \leq w_i \leq s_i, i = 1, \dots, k, \\ & \xi_j \geq 0, j = 1, \dots, m. \end{aligned}$$

Where C is a penalty parameter on the training errors.

Actually, the larger the lwl is, the more discriminative this feature is. Therefore, we use this score as a feature selection criterion.

4 K-Fold Cross Validation

When we have finished the feature selection, we use the SVM to do the classification. The cross validation will help to identify good parameters so that the classifier can accurately predict unknown data. In this paper, we use 10 fold cross validation to choose the penalty parameter C and γ in the SVM. When we get the nice arguments, we will use them to train model and do the final prediction.

5 Experimental Results

In this part, we use LIBSVM (Chang and Lin 2001) for SVM classification. Before doing the classification, we perform some data preprocessing. We choose 5499 positive records and 5498 negative records from the data set and then delete the non-numeric features, and scale each feature to $[0,1]$.

Firstly, we use the SVM to do the classification without any feature selection. According to the 10-fold cross validation, we get the high accuracy as follows:

Table 1. Cross Validation with SVM on the Training Set

C	2^{-5}	2^{-4}	2^{-3}	2^{-2}	2^{-1}	2^0
Accuracy	80.53%	81.70%	83.11%	84.33%	85.16%	85.77%
C	2^1	2^2	2^3	2^4	2^5	2^6
Accuracy	86.35%	87.09%	87.55%	88.01%	88.42%	88.89%
C	2^7	2^8	2^9	2^{10}		
Accuracy	89.09%	89.21%	89.23%	89.25%		

Obviously, when $C=2^{10}$, we get the highest accuracy 89.25%.

Then we will focus on the feature selection strategies mentioned in Section 3.2—3.4.

For the SVM feature selection strategy, after the above cross validation, with C equaling to 2^{10} , we can get the model $f(x) = \text{sgn}(w^T x_k + b)$. Then we put the lwl in order, which stands for the weight of each feature. In fact, there are 226 features with $lwl > 0$. In the experiment, we use the half-cut method to choose the features, which means every time we just maintain half features whose lwl are bigger than others. Using the same cross validation method we can select the optimized C , and the following table gives the top five accuracies with each half feature selected.

We find that only when there are still 113 features, the accuracy can be over 80%.

For the F-score feature selection strategy, as we have discussed in Section 3.3, first we calculate the F-score of each feature, and then put them in order, at last use the

half-cut method to choose the features. Using the same cross validation method we can select the optimized C, the following table gives the top five accuracies with half features selected.

Table 2. Cross Validation with SVM after SVM Feature Selection on the Training Set

C	Accuracy (3 features)	Accuracy (7 features)	Accuracy (14 features)	Accuracy (28 features)	Accuracy (56 features)	Accuracy (113 features)	Accuracy (all features)
2^8	66.04%	66.02%	67.26%	72.85%	73.20%	81.90%	89.21%
2^9	67.37%	66.35%	67.88%	73.28%	73.38%	81.91%	89.23%
2^{10}	66.81%	67.37%	68.25%	73.53%	73.84%	81.96%	89.25%

Table 3. Cross Validation with SVM after F-Score Feature Selection on the Training Set

C	Accuracy (3 features)	Accuracy (7 features)	Accuracy (14 features)	Accuracy (28 features)	Accuracy (56 features)	Accuracy (113 features)	Accuracy (all features)
2^8	70.08%	70.62%	78.86%	84.69%	86.20%	88.32%	89.21%
2^9	70.07%	70.62%	78.89%	84.86%	86.33%	88.52%	89.23%
2^{10}	70.07%	70.61%	78.94%	84.99%	86.47%	88.69%	89.25%

From the table we find that when we use the first 28 features, we can get a satisfying accuracy 84.99%.

For the one-norm SVM feature selection strategy, we can also get the l_{w1} from the model the same as the SVM feature selection. In this experiment, we figure out that there are only 26 $l_{w1} > 0$, so with the same method, following table gives the top five accuracies with each half feature selected.

Table 4. Cross Validation with SVM after One-Norm Feature Selection on the Training Set

C	Accuracy (3 features)	Accuracy (7 features)	Accuracy (13 features)	Accuracy (26 features)	Accuracy (all features)
2^8	70.66%	74.01%	74.74%	80.97%	89.21%
2^9	70.66%	74.01%	74.74%	81.06%	89.23%
2^{10}	70.66%	74.01%	74.77%	81.12%	89.25%

All the above strategies are based on the hypothesis that the data set can be classified by a linear decision function. The following experiments are based on the hypothesis that the data set can not be classified by a linear function. As a result, we introduce the RBF kernel to the SVM.

When we use the SVM directly to choose the parameters C and γ , we get the graph as below:

Fig. 1. Direct Use SVM with RBF Kernel on the Training Set

The following table gives the top five accuracies with all features.

Table 5. Cross Validation with RBF Kernel SVM on the Training Set

$\gamma \backslash C$	2^6	2^7	2^8	2^9	2^{10}
2^{-6}	88.11%	88.73%	89.58%	90.21%	90.62%
2^{-5}	88.99%	89.90%	90.49%	90.81%	91.14%
2^{-4}	90.11%	90.62%	90.76%	90.98%	90.84%
2^{-3}	90.27%	90.43%	90.33%	90.23%	90.15%
2^{-2}	89.50%	89.77%	89.56%	89.52%	89.63%

Compared with the other values of C , the value 2^{10} can always get the better accuracy; therefore, we just fix C at 2^{10} to choose the parameter γ . Based on the above three feature selection strategies, we still use the features selected respectively to do the RBF kernel SVM, we can get the following three tables for each strategy.

Table 6. Cross Validation with RBF Kernel SVM after SVM Feature Selection on the Training Set

γ	Accuracy (3features)	Accuracy (7features)	Accuracy (14features)	Accuracy (28features)	Accuracy (56features)	Accuracy (113features)	Accuracy (allfeatures)
2^{-6}	62.17%	62.51%	63.62%	70.64%	71.21%	80.74%	90.62%
2^{-5}	63.47%	63.57%	65.17%	71.38%	71.92%	81.00%	91.14%
2^{-4}	65.88%	65.83%	66.17%	72.44%	72.67%	83.59%	90.84%
2^{-3}	66.02%	66.04%	67.22%	72.91%	73.29%	84.50%	90.15%

Table 7. Cross Validation with RBF Kernel SVM after F-Score Feature Selection on the Training Set

γ	Accuracy (3features)	Accuracy (7features)	Accuracy (14features)	Accuracy (28features)	Accuracy (56features)	Accuracy (113features)	Accuracy (allfeatures)
2^{-6}	69.97%	76.00%	80.97%	86.76%	88.31%	89.33%	90.62%
2^{-5}	69.93%	77.49%	81.12%	87.22%	88.66%	89.71%	91.14%
2^{-4}	69.96%	77.64%	81.70%	88.29%	89.48%	90.20%	90.84%
2^{-3}	69.98%	77.98%	82.94%	88.85%	89.83%	89.71%	90.15%

Table 8. Cross Validation with RBF Kernel SVM after One-Norm SVM Feature Selection on the Training Set

γ	Accuracy (3 features)	Accuracy (7 features)	Accuracy (13 features)	Accuracy (26 features)	Accuracy (all features)
2^{-3}	72.93%	74.38%	75.60%	83.80%	89.97%
2^{-2}	72.92%	74.79%	74.98%	84.88%	90.62%
2^{-1}	72.92%	74.88%	75.56%	85.23%	91.14%

In table 1, it is obvious that the accuracy can be over 80% as long as the parameter C is bigger than 1 with all the 230 features, and also, in table 5, when we introduce the RBF kernel into SVM, the rate can be better than 90% with the proper parameters. In table 2, the accuracy could not be higher than 80% unless we maintain 113 features at least; with the same feature selection strategy, in table 6, we also have to leave no less than 113 features in the model to get the 80% accuracy; In table 3, we just need to keep only 28 features in order to get the accuracy better than 80% as well as 14 features in table 7; Meanwhile, in table 4 and table 8, we can get the 80% rate with 26 features.

6 Conclusion

In this paper, we experiment several feature selection strategies to work on the VIP E-Mail accounts data set. On the whole, the strategies with RBF kernel are better than the ones without it; the F-Score strategy and norm-SVM strategy are better than the SVM strategy. According to the result, the websites hosting companies can reduce the labor cost for information collection. On one hand, the engineers can be released from dealing with the massive log files; on the other hand, the users can get a quick and attractive registration without providing so much personal information.

Acknowledgments. In the process of performing this paper, the authors get much support from the follow members: Zhang Dongling, Liu Rong, Li Xingsen, Zhang Peng, Li Aihua, without their help, we could not get the balanced samples from the VIP E-Mails accounts data bases. We express our esteem with the deepest appreciation. This

research has been partially supported by a grant from National Natural Science Foundation of China (#10601064,#70531040) and NSFB(#9073020).

References

1. Chang C.-C, Lin C.-J. LIBSVM: a library for support vector machines, (2001). Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
2. iResearch, China E-mail Market Research Report 2005 (Online). Available: http://www.iresearch.com.cn/html/email_service/detail_free_id_27497.html. (2005)
3. Chen Y.W., Lin C.J., Combining SVMs with various feature selection strategies. (2005)
4. Chapelle O., Vapnik V., Bousquet O., Mukherjee S.. Choosing kernel parameters for support vector machines. Machine Learning, Submitted. (2000)
5. Cortes C., Vapnik V.: Support-vector networks. Machine Learning, 20(3) (1995):273–297.