An improved active learning method based on feature selection

Chunjiang Fu[1, a *], Liang Gong[1,b] and Yupu Yang[1,c]

[1] Department of Automation, Shanghai Jiao Tong University, and

Key Laboratory of System Control and Information Processing, Ministry of Education, China

[a]fcj2519@126.com, [b]Lianggong@sjtu.edu.cn, [c]ypyang@sjtu.edu.cn

Keywords: active learning, support vector machine, principal component analysis, PCA

Abstract. An improved active learning method taking advantage of feature selection technique is proposed. In early stages of active learning, the whole dataset is described using only the few key features, so that its overall distribution characteristic can be learned easily, reducing active learning's possibility of falling into bad local optimum. As active learning proceeds, more and more data get labeled. Only then are detailed features of the dataset gradually added to further enhance the model's classification performance. Experiments show that it is more efficient and more robust than traditional technique.

Introduction

Recently, active learning [1] has become a key technique for dealing with machine learning tasks where unlabeled data are abundant or easy to collect but class labels of each data sample are difficult or costly to obtain. Actually, most real-world classification tasks fits into this category, as the training data need to be labeled by an human annotator.

In the most widely used pool-based active learning scenario, active learning starts with a small set of labeled data *L* and a large pool of unlabeled data *U*. An initial model is trained using the few labeled instances. Then in each consecutive iterations, the active learner selects out a few unlabeled instances that are most informative according to the current model. After an oracle (e.g., a human annotator) offers class labels of these data samples, they are used to update the classification model. This process continues until some stopping criterion is met. In this way, unnecessary and redundant samples are much less likely to be included in the training set, thus reducing the labeling cost and potentially the computational cost greatly.

Active learning methods are usually greedy. The few labeled instances for training the initial classifier can influence the whole process greatly. If not initialized properly, active learning might fall into very poor local minimal solution, which is a common problem of greedy methods. Currently, there's no widely acceptable method for choosing good initial samples for active learning. Usually, they're just randomly selected.

In this paper, an improved active learning method based on feature selection is proposed. In the early stages of the active learning, the number of labeled samples is very small. According to the proposed method, the dataset should be described using only a few key features, so that the classification model is able to grasp the overall structure of the data set easier. As the active learning process continues, more and more data are labeled. Only then detailed features of the dataset are used to further improve the accuracy of the model classification. Without these detailed features, the classifier can't achieve high accuracy due to lacking of useful information.

Active learning with feature selection

Consider a classification task on a dataset containing samples pertaining to 2 categories. Each samples is characterized by three features, F_1, F_2 and F_3. Feature F_1 covers most information

needed for classification. Feature F_2 covers less information than F_1. And information contained in feature F_3 is actually not relevant with classification.

Suppose a labeled data set $L^{(0)}$ containing two samples with different class labels is used to initialize active learning. Besides, the values of feature F_1, F_2 of the two samples are relatively close, but their F_3 values differ greatly. As a result, F_3 will take a heavy weight in the initial classification model $f^{(0)}$. Then active learning will select the most informative unlabeled data sample according to $f^{(0)}$. It will actually focus on the value of feature F_3 when measuring the informativeness of each sample. This makes active learning less effective than expected, even worse than random sampling.

To overcome this problem, we can perform feature selection beforehand and describe the whole dataset using only a few key features in the early stages of active learning, preventing the key features get swamped by less important features.

Usually, the usefulness of each feature describing the dataset is measured by using information gain or mutual information. But with active learning, there's not enough labeled data to do this. So we employ principal component analysis (PCA) [2] to weighting the importance of each feature. PCA requires only unlabeled data.

Details of the improved active learning method is in table 1.

Table 1 Active learning with feature selection

Algorithm 1: Active learning with feature selection
Input: $m \times n$ dimensional data matrix $\boldsymbol{X}$ (each column represents a data sample), Integer k_1 (number of initial samples), Integer k_2 (number of samples to select in each active learning iteration), Integer k_3 (number of active learning iterations to take during each stage). Output: An SVM classifier
Step 1: Perform PCA on the original data set, retaining 70%, 80%, 90%, and 100% variance, getting four different data matrices, denoted as $\boldsymbol{Z}_1$、 $\boldsymbol{Z}_2$、 $\boldsymbol{Z}_3$、 $\boldsymbol{Z}_4$. Step 2: Randomly select k_1 data points, saving their indices in a set I_L. Create another set $I_U = \{0, 1, 2, \cdots, n-1\} - I_L$ for saving indices of unlabeled data. Step 3: Let $t = 0$. **For *j* = 0 to 4:** Step 4: Learn a classification model $f^{(t)}$ from the labeled dataset $L = \{\boldsymbol{z}_i \in \mathrm{Z}_j \mid i \in I_L\}$. Step 5: Actively select the most informative k_2 unlabeled instances. Their indices form a set I_Q. Label these instances and let $I_L = I_L \bigcup I_Q$, $I_U = I_U - I_Q$. Step 6: $t = t + 1$. Step 7: Repeat step 4 ~ step 6 $k_3 - 1$ times. **End for** Step 8: Repeat step 4 ~ step 6 until the stopping criterion is met. Step 9: Return $f^{(t)}$.

Experiments

To evaluate the effectiveness of the proposed active learning method in table 1, we compared it with traditional method on the widely used letter dataset. The letter dataset contains 20,000 instances, each representing a capital English letter. Each instance has 16 features.

Support vector machine (SVM) [3] with Gaussian kernel was employed to be the classifier. A single SVM can only deal with binary classification task. So three groups of experiments were performed. The 1st one aimed to obtain an SVM model for separating letter 'A' from other letters and the 2nd one for separating letter 'B' and the 3rd one 'C'.

The Gaussian kernel is defined as

$$K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = e^{-\gamma\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2}, \tag{1}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two data samples and γ is the kernel function's parameter.

In our experiments, γ is set to be 1.0 and the other parameter of SVM is set as $C = 10$.

Three traditional active learning support vector machine (ALSVM) are used for comparison. They are applied on the $\boldsymbol{Z}_2$、$\boldsymbol{Z}_3$、$\boldsymbol{Z}_4$ data matrices respectively. The dimension of the dataset is fixed during the active learning process.

The PCA implementation used in the experiments is from the mlpack library. [4]

Each experiment is repeated 100 times, each time initialized with 10 randomly selected instances ($k_1 = 10$). All the four ALSVMs always use the same set of instances for initialization. In each active learning iteration, the data sample most nearest to the current separating hyper-plane (of SVM) are selected ($k_2 = 1$). In each stage of active learning, 3 iterations are performed ($k_3 = 3$).

Average classification error rates with respect to the number of iterations are shown in the following three figures. In the figures' legends, 80% means applying PCA on the original dataset, retaining 80% percent of the variance, and then perform ALSVM on the result dataset. "hybrid" indicates using the proposed algorithm 1, adding features gradually as active learning proceeds.

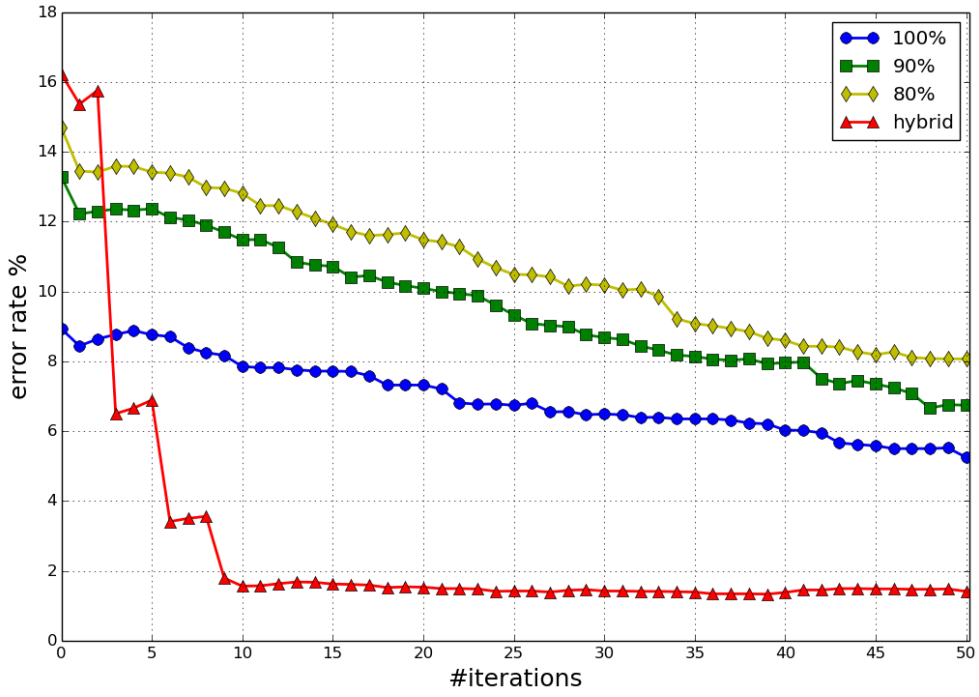

Fig.1 'A' vs others

As shown in the figure, the proposed active learning method reduces the classification error to a level below 2% with only 10 iterations, while traditional methods need more than 50 iterations. A lot of label cost can be saved.

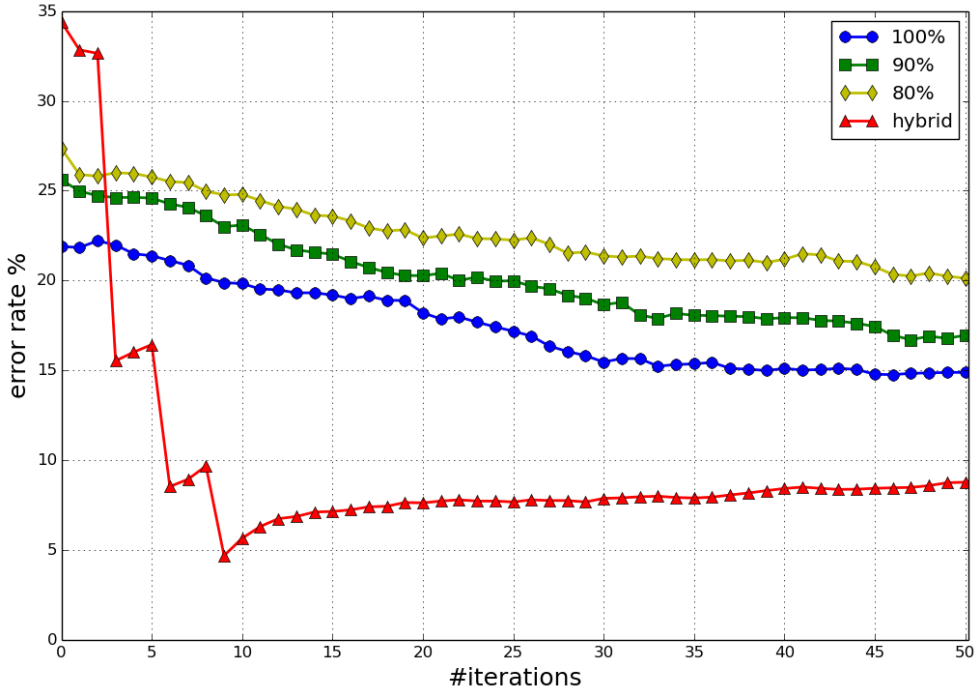

Fig.2 ‘B’ vs others

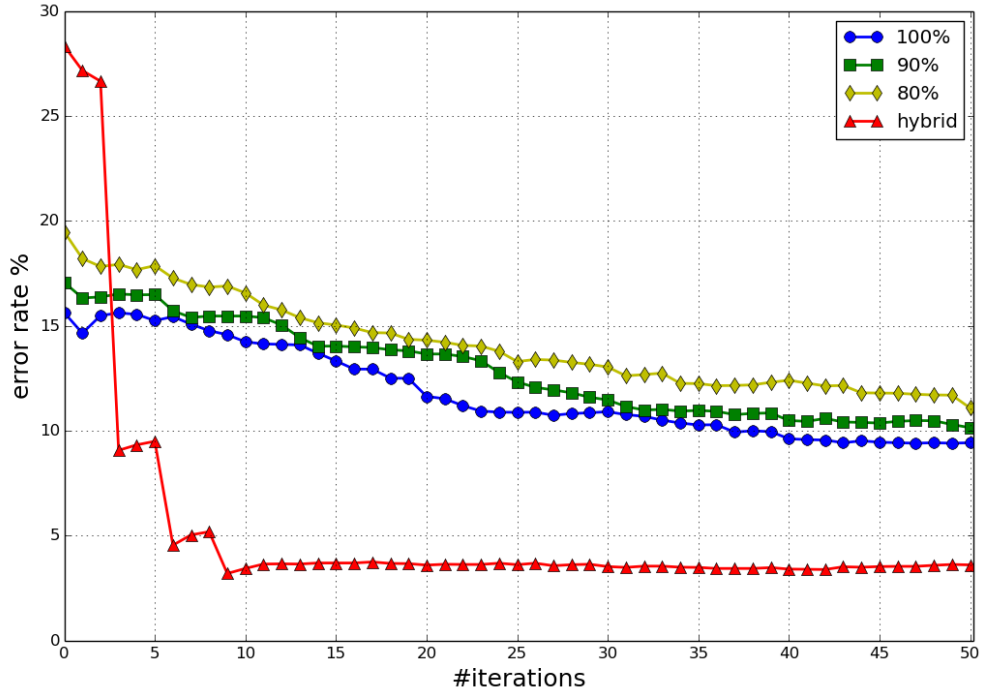

Fig.3 ‘C’ vs others

Statistics of the final error rate among 100 repetitions are shown in table 2, with boldface indicating the best.

We can see from the table that in the best case, all ALSVMs get very good classification performance. The proposed method is much more robust than traditional ones. Generally, it can achieve better performance using less labeled data and is less influenced by the initial samples.

Table 2 Statistics of the final error rate (%)

	features used	MAX	MIN	MEAN	STDEV
'A' vs others	100%	48.62	0.58	5.25	8.13
	90%	45.78	0.79	6.74	9.20
	80%	53.39	1.05	8.07	9.71
	hybrid	**4.78**	**0.49**	**1.41**	**0.94**
'B' vs others	100%	50.22	**3.02**	14.89	10.98
	90%	51.92	4.96	16.95	9.77
	80%	49.11	5.46	20.13	9.59
	hybrid	**19.58**	3.34	**8.77**	**3.43**
'C' vs others	100%	58.80	1.40	9.43	9.60
	90%	49.27	1.93	10.15	9.39
	80%	42.55	2.37	11.09	8.05
	hybrid	**10.17**	**1.34**	**3.61**	**1.80**

Conclusion and discussion

The experiment results shows that the proposed active learning method is superior than traditional ones. Using only the major features of the dataset in early stages of active learning can improve the performance of active learning.

Table 1 actually acts as a framework. In the proposed method, PCA is used for feature selection. This leads to quite good results on the dataset used in the experiments. In practice, with more complicated datasets, we can also use other more advanced feature selection technique, such as Independent Component Analysis (ICA) [5], Non-negative Matrix Factorization (NMF) [6] and Manifold Learning [7] and so on.

Acknowledgements

This work is partly supported by National Nature Science Foundation of China (No. 61273161).

References

[1] Settles B. Active learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2012, 6(1): 1-114.

[2] Abdi H, Williams L J. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, 2(4): 433-459.

[3] Chang C C, Lin C J. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3): 27.

[4] Curtin R R, Cline J R, Slagle N P, et al. MLPACK: A scalable C++ machine learning library. The Journal of Machine Learning Research, 2013, 14(1): 801-805.

[5] Du K L, Swamy M N S. Independent component analysis. Neural Networks and Statistical Learning. Springer London, 2014: 419-450.

[6] Huang K, Sidiropoulos N, Swami A. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. Signal Processing, IEEE Transactions on, 2014, 62(1): 211-224.

[7] Talwalkar A, Kumar S, Mohri M, et al. Large-scale svd and manifold learning. The Journal of Machine Learning Research, 2013, 14(1): 3129-3152.