Active learning favoring points near the border between clusters

Chunjiang Fu[1, a], Yupu Yang[1, b]

[1]*Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education Shanghai 200240, China*

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

Abstract.

An active learning SVM technique taking advantage of the cluster assumption was proposed. In each active learning iteration, unlabeled instances in the SVM margin were first grouped into two clusters. Then from each cluster, points most similar to the other cluster were selected for labeling. Such points lying near the border between clusters were expected to become support vectors with higher probability. The clustering process was performed in the same kernel space as SVM. With semi-supervised K-medoids, labeled instances were also used to improve the clustering performance. Experiments showed that the proposed method was efficient and robust (to poor initial samples).

Keywords: Active learning; SVM; support vector machine; k-medoids clustering

Introduction

In many machine learning tasks, unlabeled data are relatively easy to collect, but class labels are difficult or time-consuming to obtain. To get the best classification performance with least labeled data, two classes of techniques have been developed by the machine learning community: Semi-supervised Learning [1] and Active Learning [2].

Semi-supervised learning tries to improve the performance using useful information contained in the unlabeled data. This is usually achieved by taking some reasonable assumptions about the data. The most widely used assumption

in semi-supervised learning is the cluster assumption. It assumes that in the feature space, points in the same cluster should have the same class label with high probability and therefore the decision boundary should lie in the low density regions between clusters. This is actually quite intuitive. For datasets that does not meet the cluster assumption, or in other words, points of different classes mingle together in the feature space, many people think that it is the feature extraction strategies to blame.

Instead of utilizing unlabeled data directly, active learning tries to label only the most informative instances. 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 an oracle (e.g., a human annotator) offers class labels of these data samples, they are used to update the learning 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.

In this work, an active learning method taking advantage of the cluster assumption is proposed. We focus on binary classification tasks using SVM (support vector machine). Unlabeled data points lying near the border between clusters are favored over the most uncertain ones.

Support vector machine

With n labeled data points $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i \in R^d$, and $y_i \in \{\pm 1\}$, the SVM training algorithm tries to solve the following optimization problem:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{1}$$

$$\text{s.t.} \quad \forall i \; 0 \le \alpha_i \le C \tag{2}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3}$$

where α_i s are Lagrangian multipliers, *C* is a user chosen constant, and $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a kernel function. Among various kernel functions, the Gaussian kernel is the most widely used one:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|} \tag{4}$$

For a new test sample $\boldsymbol{x}$, sign of the following equation can be used to predict its class label.

$$f(\boldsymbol{x}) = \sum_i \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{5}$$

Semi-supervised K-medoids Clustering

To explore the cluster structure in the dataset, K-medoids clustering is used to group candidate points. K-medoids clustering has been proved to be NP-hard and most implementations are computationally heavy. [3] proposes a very efficient algorithm for K-medoids clustering that runs very similar to the K-means algorithm. We use it instead of K-means in the "Constrain-Kmeans" method proposed in [4] and get a simple semi-supervised K-medoids clustering method.

The clustering process takes place in the same kernel space as SVM. For simplicity, we just use the Gaussian kernel as similarity measure. For a cluster P, its medoid $\boldsymbol{x}_P$ is calculated as:

$$\boldsymbol{x}_P = \arg\max_{\boldsymbol{x}_i \in P} \sum_{\boldsymbol{x}_j \in P} K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{6}$$

The semi-supervised K-medoids clustering method is summarized in table 1.

Table 1 Binary Semi-supervised K-medoids

Algorithm 1: Binary Semi-supervised K-medoids
Input: Labeled dataset *L*, unlabeled dataset *M*
Output: Two clusters *P* and *N*, as well as their medoids $\boldsymbol{x}_P$ and $\boldsymbol{x}_N$
Step 1: $P_0 = \{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in L \text{ and } y_i > 0\}$, $N_0 = \{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in L \text{ and } y_i < 0\}$
Step 2: Calculate medoids of the two clusters using equation (6) $\boldsymbol{x}_P = \arg\max_{\boldsymbol{x}_i \in P_0} \sum_{\boldsymbol{x}_j \in P_0} K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{x}_N = \arg\max_{\boldsymbol{x}_i \in N_0} \sum_{\boldsymbol{x}_j \in N_0} K(\boldsymbol{x}_i, \boldsymbol{x}_j)$
Repeat:
Step 3: $P = P_0$, $N = N_0$
Step 4: For each $\boldsymbol{x}_i \in M$, add $\boldsymbol{x}_i$ into *P* if $K(\boldsymbol{x}_i, \boldsymbol{x}_P) > K(\boldsymbol{x}_i, \boldsymbol{x}_N)$, or

into *N* elsewise

Step 5: $\boldsymbol{x}_P = \arg\max_{\boldsymbol{x}_i \in P} \sum_{\boldsymbol{x}_j \in P} K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{x}_N = \arg\max_{\boldsymbol{x}_i \in N} \sum_{\boldsymbol{x}_j \in N} K(\boldsymbol{x}_i, \boldsymbol{x}_j)$

Until $\boldsymbol{x}_P$ and $\boldsymbol{x}_N$ do not change

Step 6: $P = P - P_0$, $N = N - N_0$

Step 7: Output *P*, *N*, $\boldsymbol{x}_P$, $\boldsymbol{x}_N$

The data selection strategy

The proposed data selection strategy for active learning is summarized in table 2. In each active learning iteration, unlabeled instances in the SVM margin are first grouped into two clusters using a semi-supervised K-medoids clustering algorithm. Then from each cluster, points most similar to the other cluster are selected for labeling. Such points lying near the border between clusters are expected to become support vectors in the final classification model with high probability.

Table 2 Active learning based on semi-supervised clustering

Algorithm 2: Active learning based on semi-supervised clustering

Input: Labeled dataset *L*, unlabeled dataset *U*, batch size *k*
Output: An SVM classifier
Step 1: Train an SVM with initial points in *L*
Repeat:
Step 2: $M = \{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in U \text{ and } |f(\boldsymbol{x}_i)| <= 1\}$, where $f(\boldsymbol{x})$ is in equation (5)
Step 3: If $|M| < 2k$, let $Q = M$ and go to step 6
Else let $Q = \varnothing$
Step 4: Group $\boldsymbol{x}_i$ s in *M* into two clusters *P* and *N* with algorithm 1
Step 5: For $\boldsymbol{x}_i \in P$, select *k* ones that has largest $K(\boldsymbol{x}_i, \boldsymbol{x}_N)$
For $\boldsymbol{x}_j \in N$, select *k* ones that has largest $K(\boldsymbol{x}_j, \boldsymbol{x}_P)$
Add these points into *Q*
Step 6: Randomly delete instances from *Q* as long as $|Q| > k$
Step 7: Label points in *Q* and add them into *L*
Step 8: Retrain the SVM on *L*
Until stopping criterion is met
Step 9: Output the final SVM model

Experiment on the USPS dataset

To test the effectiveness of the proposed method, we compared it with two other traditional techniques listed below. Given a fixed batch size *k*:

(1) Random: randomly select *k* instances in the unlabeled pool U;

(2) Uncertain: select *k* unlabeled samples nearest to the current SVM separating hyperplane;

USPS [5] is a widely used benchmark dataset containing 7291 handwritten digits, among which 645 ones are "8". We set class 8 with label "+1" and the rest "-1". This is a common setting when adopting the "one-vs-all" scheme for employing SVM in multiclass problems.[6]

We run these algorithms 50 times, and draw the average test error rate with respect to the number of iterations. In each run, 50% of data randomly selected are reserved for testing, and the rest 50% as the unlabeled pool. In each run, one instance from each class are selected as initial labeled samples for all the methods. In each iteration, 10 instances are selected for updating the classification model. Parameters for SVM are set as C=10, γ=0.001.

As demonstrated in figure 1, our method get an error rate less than 3% with only 10 iterations on average. Others get this with more than 16 iterations.

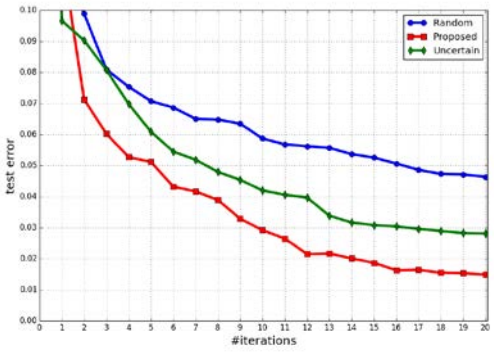

Fig. 1 Average test error rates on USPS

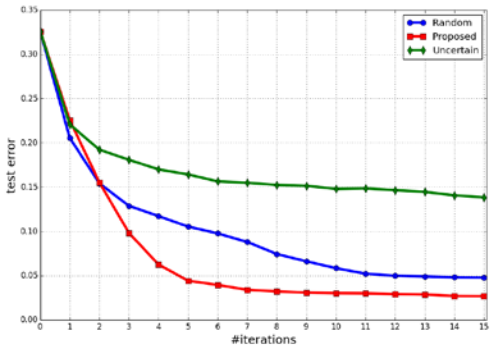

Fig. 2 Average test error rates on w5a

Experiment on the w5a dataset

Another experiment is on text categorization: classifying whether a web page belongs to a category or not. The original dataset consists 49749 web pages, with 300 sparse binary keyword attributes extracted from each. [7] For simplicity, we used only a subset containing 9888 instances, the w5a dataset. [8]

There is some labeling noise in the dataset. To improve robustness of active learning methods, 5 instances from each class are used for initialization. In each iteration, 5 instances are selected. Parameters are set as C=100, γ=0.01. Other settings are the same as the former experiment.

Figure 2 shows average test errors over 50 runs. The proposed method obtains an error rate lower than 3% after 8 iterations, while others failed to achieve this even after 15 iterations.

In terms of average error rate, Uncertain performs worst on this dataset. In fact, in more than 20 runs, Uncertain got smaller final error than the proposed one. But unfortunately, it performs very poor in several runs, with final error rate more than 50%. This indicates that the traditional Uncertain method is vulnerable to poor initial samples. The proposed method is more robust.

Conclusion

By analyzing results of the experiments, we believe that the proposed method is more efficient in terms of label cost, and also more robust to poor initial samples than traditional methods.

K-medoids is adopted to explore the cluster structures in the data. It is both efficient and easy to implement. Maybe the performance of the proposed method

can be further improved by employing more sophisticated clustering algorithms such as spectral clustering [9][10] or maximum volume clustering [11].

Acknowledgement

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

References

[1] Zhu, X., Semi-supervised learning literature survey. Computer Science, University of Wisconsin-Madison, 2006. 2: p. 3.

[2] Settles, B., Active learning literature survey. University of Wisconsin, Madison, 2010.

[3] Park, H. and C. Jun, A simple and fast algorithm for K-medoids clustering. Expert Systems with Applications, 2009. 36(2): p. 3336-3341.

[4] Basu, S., A. Banerjee and R.J. Mooney. Semi-supervised clustering by seeding. in ICML. 2002.

[5] Hull, J.J., A database for handwritten text recognition research. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1994. 16(5): p. 550-554.

[6] Rifkin, R. and A. Klautau, In defense of one-vs-all classification. The Journal of Machine Learning Research, 2004. 5: p. 101-141.

[7] Platt, J., Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[8] Von Luxburg, U., A tutorial on spectral clustering. Statistics and computing, 2007. 17(4): p. 395-416.

[9] Bodó, Z., Z. Minier and L. Csató, Active learning with clustering. Active Learning Challenge Challenges in Machine Learning, Volume 6, 2011: p. 141.

[10] Niu G, Dai B, Shang L, et al. Maximum volume clustering: A new discriminative clustering approach[J]. The Journal of Machine Learning Research, 2013, 14(1): 2641-2687.

[11] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/