Semi-supervised Learning for SVM-KNN

Kunlun Li
College of Electronics and information Engineering, Hebei University, Baoding, 071002,china
Email: likunlun@hbu.edu.cn

Xuerong Luo and Ming Jin
College of Electronics and information Engineering, Hebei University, Baoding, 071002,china
School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876,china
Email: {baoding0312@163.com, jinan527@yahoo.com.cn}

***Abstract*—Compared with labeled data, unlabeled data are significantly easier to obtain. Currently, classification of unlabeled data is an open issue. In this paper a novel SVM-KNN classification methodology based on Semi-supervised learning is proposed, we consider the problem of using a large number of unlabeled data to boost performance of the classifier when only a small set of labeled examples is available. We use the few labeled data to train a weaker SVM classifier and make use of the boundary vectors to improve the weaker SVM iteratively by introducing KNN. Using KNN classifier doesn't enlarge the number of training examples only, but also improves the quality of the new training examples which are transformed from the boundary vectors. Experiments on UCI data sets show that the proposed methodology can evidently improve the accuracy of the final SVM classifier by tuning the parameters and can reduce the cost of labeling unlabeled examples.**

***Index Terms*—semi-supervised learning, support vector machine, K-nearest neighbor, boundary vectors**

I. Introduction

In this paper we focus on solving the classification problem by using semi-supervised learning strategy. Traditional classifiers are constructed based on labeled data in supervised learning. Labeled examples, however, are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile it is relatively easier to collect unlabeled examples and there have been a few classification approaches using unlabeled data in recent years. Semi-supervised learning addresses this problem by using large number of unlabeled data, together with the small number of labeled data, to construct better classifiers. Because semi-supervised learning requires less human effort and gives higher accuracy, it is of great interest both in theory and in practice [1].

Traditional supervised learning needs sufficient labeled data as training sets, or else can't get a supervised learning method with strong generalization [2], but obtaining lots of labeled data is difficult in practice, even can't come true. Unsupervised learning tries to find the inner structure of the unlabeled data to construct the corresponding learning machine, so it leads to unsupervised learning can't ensure high learning accuracy usually [3]. In this case, only using traditional machine learning strategy can't gain a learning machine with strong generalization and high accuracy if there is inadequate labeled data.

Semi-supervised learning is brought forward as a learning strategy in recent years, which not only makes use of the labeled data and unlabeled data but also supplements the shortages of supervised learning and unsupervised learning. Semi-supervised learning theory and algorithm developed quickly in the recent years [4], because it has been become research focus in the field of machine learning, attracting much more scholars devote themselves to the further study.

The central issue that this paper addresses is how to use information from unlabeled data to enhance the predictability of classification. In this paper, we propose a novel SVM-KNN classification method based on semi-supervised learning, which makes full use of unlabeled data. Our experimental results support the statistical learning theory showing that incorporating unlabeled data improves accuracy of the classifier when insufficient training information is available.

This paper is organized as follows. In section II we describe the related works for our method. In Section Ⅲ we introduce the proposed semi-supervised learning methodology. In section Ⅳ we present some experimental results, using a tuning method that utilizes both labeled and unlabeled data to enhance the accuracy of classification. We gieve conclusion in section Ⅴ.

II. Related work

A. Semi-supervised learning

Semi-supervised learning (SSL) is halfway between supervised and unsupervised learning [3]. In addition to

This work is supported by the National Natural Science Foundation of China (No.60773062 & No. 60873100), the Science and Technology Supporting Program of Hebei Province (No. 072135188) and Scientific research plan projects of Hebei Educational Bureau (No. 2008312).

Manuscript received January 1, 2008; revised June 1, 2008; accepted July 1, 2008. The corresponding author is Kunlun Li.

doi:10.4304/jcp.5.5.671-678

unlabeled data, this kind of methodology is provided with some supervised information – but not necessarily for all examples. Often, this information will be the labels associated with some of the examples. In this case, the data of SSL set $X = (x_i)_{i\in[n]}$ can be separated into two parts: the points $X_h = (x_1,\ldots,x_h)$, for which labels $Y_h = (y_1,\ldots,y_h)$ are provided, and the points $X_t = (x_{h+1},\ldots,x_{h+t})$, the labels of which are not known. This is the normal form of semi-supervised learning set [4].

Semi-supervised learning will be most useful whenever there are much more unlabeled data than labeled. This is likely to occur if obtaining data points is cheap, but obtaining the labels costs a lot of time, effort, or money [5]. This is the instance in many application areas of machine learning, for example, in speech recognition, it costs almost nothing to record large amounts of speech, but labeling it requires many people to listen to it and type a transcript. Since unlabeled data contain less information than labeled data, they are required in large amounts in order to increase prediction accuracy significantly [4].

A number of classification algorithms that uses both labeled and unlabeled data have been proposed, for example, self-learning or self-labeling is the earliest semi-supervised learning method Probably, which is still extensively used in the processing of natural language. S ^{3} VM, originally called Transductive SVM, they are now called Semi-Supervised SVM to emphasize the fact that they are not capable of transduction only, but also can induction. The idea is to find a decision boundary in 'low density' regions. Graph-based algorithms, one can build a weighted graph over the labeled and unlabeled examples, and assume that two strongly-connected examples tend to have the same label and solve an optimization problem. Generative models, Mixture of Gaussian or multinomial distributions, and pretty much any generative model can do semi-supervised learning. Especially for EM algorithm, which is often used for training generative models when there is unlabeled data [1].

One of the many approaches to semi-supervised learning is to first train a weaker predictor, which is then used in exploiting the unlabeled examples. For instance, in content-based image retrieval (CBIR), a user usually poses several example images as a query and asks a system to return similar images. In this situation there are many unlabeled examples, i.e. images that exist in a database, but there are only several labeled examples, i.e. the query images. Another instance is online web page recommendation. When a user is surfing the Internet, he may occasionally encounter some interesting web pages and may want the system bring him to similarly interesting web pages. It will be difficult to require the user to confirm more interesting pages as training examples because the user may not know where they are. In this instance, although there are a lot of unlabeled examples, i.e. web pages on the Internet, there are only a few labeled examples, i.e. the current interesting web pages [6]. In these situations, there are very few labeled training examples to rely on. The initial weaker predictor may not classify the other examples correctly, so we propose a novel SVM-KNN classification method based on semi-supervised learning to solve these cases.

B. Support vector machine

Support vector machine (SVM) is the youngest part in the statistical learning theory [7], whose dominating content is accomplished from 1992 to 1995 and developed quickly at present because of its solid theory and widespread applications. SVM is based on the structural risk minimization principle (SRM), which was proposed by Vapnik in 1998. Comparing with other learning methods, its generalization is optimal.

SVM is proposed through the optimal hyperplane in the linear partition case [8], the optimal hyperplane is depicted in the figure1, the pentacles and squares denote the training examples of two classes respectively, L is the separated line which partitions two classes correctly, L_1 and L_2 are the separate lines nearest to and parallel with L, the distance between L_1 and L_2 is called classification margin. The optimal classification line requests to partition two classes correctly and make the margin up to maximum so as to ensure structural risk minimization [9]. Extend to high dimension feature space, the optimal line becomes the optimal hyperplane.

The basic classification task is to estimate a classification function $f : R^N \to \{\pm 1\}$ using input-output training examples [10] from two classes

$$(x_{i,}y_i)\ \ i=1,\ldots,\ \mathrm{n}, \mathrm{x}\in R^d, y\in\{+1,-1\} \qquad (1)$$

The function f should correctly classify unseen examples (x, y) i.e. $f(x) = y$ if (x, y) have the same probability distribution with the training data. In this work we will discuss binary classification [10]. If the points are linearly separable, then there exist an n-vector w and scalar b such that

$$y_i\left[\left(\omega\cdot x_i\right)+b\right]-1\geq 0,\ i=1,2,\ldots,\ \mathrm{n} \qquad (2)$$

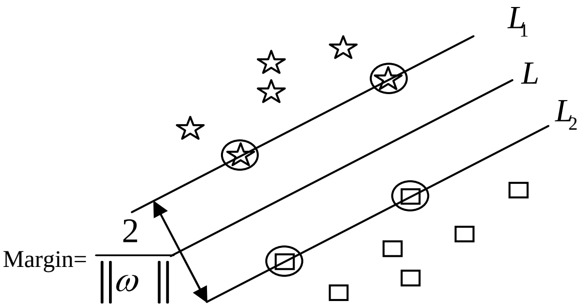


Figure1.The optimal hyperplane

The “optimal” separating plane, $\omega \cdot \mathrm{x+b=0}$, is the one which is furthest from the closest points in the two classes [11]. Geometrically this is equivalent to maximizing the separation margin or distance between the two parallel planes $w \cdot x + b = \pm 1$ i.e. L_1, L_2 (see Figure 1.)

The “margin of separation” in Euclidean distance is $2/\|\omega\|_2$ where $\|w\|_2 = \sum_{i=1}^{n} w_i^2$ is the 2-norm. To maximize the margin, we minimize $\|w\|$ subject to the constraints (2). According to structural risk minimization[12], for a fixed empirical misclassification rate, larger margins should lead to better generalization and prevent over-fitting in high-dimensional feature spaces, so the task of standard SVM is:

$$\min \Phi(\omega) = \frac{1}{2}\|\omega\|^2 = \frac{1}{2}(\omega \cdot \omega)$$
$$s.t. y_i \left[\left(\omega \cdot x_i \right) + b \right] - 1 \geq 0,\ i = 1, 2, \ldots,\ \mathrm{n} \quad (3)$$

The Lagrangian equation is:

$$L(w, b, \alpha) = \frac{1}{2}(w \cdot w) - \sum_{i=1}^{n} \alpha_i \{ y_i [(w \cdot x_i) + b] - 1 \} \quad (4)$$

Then the former of the optimal problem accordingly becomes to:

$$\begin{cases} \min L(\alpha) = 1/2 \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N} \alpha_i \\ s.t \quad \alpha_i \geq 0, \quad i = 1, 2, \ldots, N \\ \sum_{i=1}^{N} y_i \alpha_i = 0 \end{cases} \quad (5)$$

At last it calculates the decision function[13]:

$$\begin{aligned} f(x) &= \mathrm{sgn}\{(w^* \cdot x) + b^*\} \\ &= \mathrm{sgn}\{\sum_{i=1}^{n} \alpha_i^* y_i (x_i \cdot x) + b^*\} \end{aligned} \quad (6)$$

The above only considers the linearly separable case. But in practice the most data is nonlinearly separable. In order to solve the nonlinearly separable cases, we introduce the kernel function into SVM. Generally to say, through the space mapping of the data, the dimensional-low data will be mapped into a sufficiently high dimensional space. Then the data can be linearly separable in the high dimensional space. To avoid the complex inner product operation of the high dimensional space, the kernel function uses the simple operation of the original space to replace it [14] [15]. Then:

$$K(x, x_i) = \varphi(x) \cdot \varphi(x_i) \quad (7)$$

The quadratic programming of the classical SVM is:

$$\begin{cases} \min_{\alpha_i} Q(\alpha_i; \varphi(x_i)) = \frac{1}{2} \sum_{i,j=1}^{N} y_i y_l \varphi(x_i)^T \varphi(x_j) \alpha_i \alpha_j - \sum_{i=1}^{N} \alpha_i \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq c, \qquad i = 1, 2, \cdots, N \end{cases} \quad (8)$$

The final classifier is：

$$f(x) = \mathrm{sgn}\left[\sum_{i=1}^{N} \alpha_i y_i K(x \cdot x_i) + b \right] \quad (9)$$

C. Selection of Kernel function

There are four common kernel functions depicted as below, we must decide which one to try first. Then the penalty parameter C and kernel parameters are chosen.

- Linear: $K(x_i, x_j) = x_i^T x_j$.
- Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$.
- Radial basis function (RBF):
  $$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$$
- Sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$.

In our experiment, we choose RBF kernel as our kernel function. The RBF kernel nonlinearly maps examples into a higher dimensional space, unlike the linear kernel, it can handle the situation when the relation between class labels and attributes is nonlinear. What is more, the linear kernel is a special case of RBF as [16] [17] shows that the linear kernel with a penalty parameter has the same performance as the RBF kernel with some parameters. In addition, the sigmoid kernel behaves like RBF for certain parameters [17] and the number of hyper-parameters which influences the complexity of model selection. The polynomial kernel has more hyperparameters than the RBF kernel [18].

D. K-Nearest neighbor

The K-Nearest Neighbor (KNN) algorithm is proposed by Cover and Hart in 1968 [19], whose theory has been developed maturely. K nearest neighbors are calculated using Euclidean distance, though other measures are available, Euclidean distance offers a fine mix of ease and efficiency. The classification of the example is determined by a majority vote of the labels of the *k*-near neighbors [20]. Intuitively, This method is very simple: for instance, if example x_1 has k nearest examples in the feature space and a majority of them have the same label y_1, then example x_1 belongs to y_1.

Although KNN method depends on utmost theorem in the theory, during the decision course it is only related to small number of nearest neighbors, so adopting this method can avoid the problem of examples imbalanced, otherwise, KNN mainly depends on limited number of nearest neighbors around not a decision boundary, so it is suitable for classifying the case of examples set of boundary intercross and examples overlapped.

Euclidean distance is calculated as follows [21]: suppose two vectors x_i and x_j, $x_i = (x_i^1, x_i^2, \dots, x_i^n)$, $x_j = (x_j^1, x_j^2, \dots, x_j^n)$,

the distance between x_i and x_j is :

$$D(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_i^k - x_j^k)^2} \qquad (10)$$

In our experiment, we estimate the nearest neighbor of an example according to this formula.

III. SVM-KNN METHODOLOGY BASED ON SEMI-SUPERVISED LEARNING

A. The motivation of the methodology

In many pattern classification problems, if there is plenty of unlabeled data while only a small number of labeled data is available, then we should adopt semi-supervised learning strategy, the existing methods have different kinds of restrictions at present, so how to seek an approach to solve the classification problem extensively is a troublesome problem. We design a semi-supervised learning methodology by combining SVM and KNN algorithm, SVM fails to provide an accurate estimate of the true decision boundaries, because of the small size of labeled data. In contrast, we can utilize the information from the unlabeled data, which can help to recover the true decision boundaries for classification. As for SVM classification, the support vectors decide the decision boundaries directly, while the boundary vectors stand a good chance to be the support vectors, so we can choose the boundary vectors to rectify the decision boundaries iteratively. We employ KNN algorithm to label the boundary vectors because KNN mainly depends on limited number of nearest neighbors around, so it is suitable for classifying the case of examples set of boundary intercross and examples overlapped. At last the boundary vectors are mingled with the initial training examples to improve the accuracy of classification.

The primary goal of this paper is to develop a semi-supervised learning methodology to show high performance of classification by utilizing unlabeled data. Unlike existing methods, our methodology is planed to adapt to a variety of cases unlike other approaches have many restrictions. It yields an improvement when unlabeled data can help to reconstruct the optimal classification boundary by tuning three parameters. The three parameters can be depicted in the section

B. Proposed methodology

When we classified a data set including large number of unlabeled data, if only utilize the few training examples available, then we can't obtain a high accuracy classifier with inadequate training examples; if we want to obtain a classifier of high performance, then labeling the unlabeled data is necessary, but labeling vast unlabeled data wastes time and consumes strength. In this paper, we propose a novel method which uses SVM cooperated with KNN for classification based on semi-supervised learning theory. The general model is depicted as above (See Figure 2). To begin with, we construct a weaker classifier SVM according to the few training examples available, then using the weaker SVM classifies the remaining large number of unlabeled data in the data set, picking out n examples belonging to each class around the decision boundary by calculating Euclidean distance in the feature space, because the examples located around the boundary are easy to be misclassified, but they are likely to be the support vectors, we call them boundary vectors, so picking out these boundary vectors whose labels are fuzzy labeled by the weaker classifier SVM. Secondly we recognize these boundary vectors as testing set while recognize initial training examples as training set, use KNN method to classify them and recognize the results as the labels for boundary vectors. In the end, we put these boundary vectors and their labels into initial training set to enlarge the number of the training examples then retrain a SVM, iteratively until the number of the training examples is m times of the whole data set. The experimental results on three UCI data sets indicate that the final classifier SVM has significant improvement on accuracy.

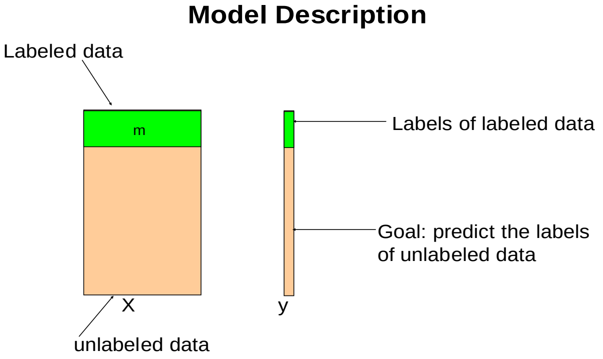


Figure2. The general description of the proposed model

The detailed steps of our method are as follows:

1) Utilize the labeled data available in a data set as initial training set and construct a weaker classifier SVM1 based on this training set.
2) Utilize SVM1 to predict the labels of all the remaining unlabeled data in the data set, then pick out 2 n examples located around the decision boundary as boundary vectors.
    a) Choose an example x_i from the class of A (A is the label) and calculate the distance between x_i and all the examples of class B (B is the label) using Euclidean distance subsequently pick out n examples of B corresponding to the n minimum distances.
    b) Choose an example y_i from the class of B (B is the label) and calculate the distance between y_i and all the examples of class A (A is the label) using Euclidean distance subsequently pick out n examples of A corresponding to

the n minimum distances.

c) We call the $2n$ examples as boundary vectors, make the $2n$ boundary vectors together as a new testing set.

3) KNN classifier classifies the new testing set with the initial training set, the boundary vectors get new labels.

4) Put the boundary vectors and their new labels into initial training set to enlarge the training set, then retrain a new SVM2.

5) Iteratively as above until the number of the training examples is m times of the whole data set.

The final SVM predicts the initial unlabeled data and the remaining unlabeled data, the results indicating that it has significant improvement.

Ⅳ. EXPERIMENTAL RESULTS

Our experiment was carried out on three publicly available labeled data sets. Unlabeled data was simulated by dropping labels from some points in a given data set. In a given data set, some examples are randomly picked out to be used as the labeled training examples, while the remaining data are used as the unlabeled examples. The procedure is repeated ten times with random data partitions and reports the average result.

The three benchmark data sets used in the following experiments are all chosen from the UCI machine learning repository [22], they are Iris data set, Breast cancer data set and Ionosphere data set. Iris data set: This data set consists of 150 four-dimensional examples. It is divided into three classes of equal size 50, but we only choose the examples of the two non-linear classes. The four features of this data set are: sepal length, sepal width, petal length, and petal width. Breast cancer data set: This is a nine-dimensional data set with 683 examples in two classes. There are 444 examples in class ‘‘benign’’, and 239 examples in class ‘‘malignant’’. The nine features are: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses. Ionosphere data set: This is a 34-dimensional data set with 351 examples in two classes. All 34 attributes are continuous and the two classes are “good” and “bad”.

1） Experiment on Iris data

In the experiment, we choose all the examples of the two non-linear classes as the data set in advance, subsequently choose 10 labeled examples randomly from the data set as initial training set and get rid of the labels of the other 90 examples as unlabeled data set. We repeat the procedure ten times and report the average result.

Table 1 Experiment result on Iris data when m =0.26 n =2

k	Training examples	Testing examples	Accuracy on initial unlabeled data (%)
	10	90	91.11%
2	26	74	95.65%
4	26	74	94.20%
6	26	74	95.65%

Table 2 Experiment result on Iris data when k =1 n =2

m	Training examples	Testing examples	Accuracy on initial unlabeled data (%)
	10	90	91.11%
0.26	26	74	95.65%
0.34	36	64	96.88%
0.40	36	64	94.92%

Table 3 Experiment result on Iris data when m =0.4 k =1

n	Training examples	Testing examples	Accuracy on initial unlabeled data (%)
	10	90	91.11%
3	22	78	98.15%
5	40	60	93.10%
8	26	74	90.00%

It is shown in the three Tables that we choose 10% labeled data of the whole data set as initial training set, after iterations the final training examples become more at a different degree so that it improves the classification accuracy on the 90% unlabeled data, during the course of the experiment we can tune the three parameters to obtain the optimal result as Table 3 shows when m =0.4 k =1 n =3.

2） Experiment on Breast cancer data

In the experiment, we choose 100 data randomly from the data set as initial training set and get rid of the labels of the other 583 examples as unlabeled data set. We repeat the procedure ten times and report the average result.

Table 4 Experiment result on breast cancer data when m =0.3 n =3

k	Training examples	Testing examples	Accuracy on initial unlabeled data (%)
	100	583	78.22%
1	203	480	80.34%
5	203	480	80.77%
8	203	480	80.77%

Table 5 Experiment result on breast cancer data when k =5 n =3

m	Training examples	Testing examples	Accuracy on initial unlabeled data (%)
	100	583	78.22%
0.3	203	480	80.77%
0.4	272	411	80.69%
0.5	335	348	84.55%

Table 6 Experiment result on breast cancer data when m =0.5 k =5

n	Training examples	Testing examples	Accuracy on initial unlabeled data (%)
	100	583	78.22%
2	341	342	84.91%
5	331	352	90.06%
8	341	342	86.24%

It is shown in the three Tables that we choose 15% labeled data of the whole data set as initial training set, after iterations the final training examples become more at a different degree so that it improves the classification accuracy on the 85% unlabeled data, during the course of the experiment we can tune the three parameters to obtain the optimal result as Table 3 shows when m =0.5 k =5 n =5.

3） Experiment on Ionosphere data

In the experiment, we choose 70 data randomly from the dataset as initial training set, using the other 281data remained as unlabeled data. We repeat the procedure ten times and report the average result.

Table 7 Experiment result on Ionosphere data when m =0.3 n =3

k	Training examples	Testing examples	Accuracy on initial unlabeled data (%)
	70	281	69.40%
2	100	251	72.24%
3	100	251	71.72%
5	100	251	72.65%

Table 8 Experiment result on Ionosphere data when k =3 n =3

m	Training examples	Testing examples	Accuracy on initial unlabeled data (%)
	70	281	69.40%
0.3	100	251	72.65%
0.4	136	215	77.14%
0.45	152	199	77.84%

Table 9 Experiment result on Ionosphere data when m =0.45 k =3

n	Training examples	Testing examples	Accuracy on initial unlabeled data (%)
	70	281	69.40%
1	156	195	80.14%
2	154	197	79.27%
5	150	201	79.69%

It is shown in the three Tables that we choose 20% labeled data of the whole data set as initial training set, after iterations the final training examples become more at a different degree so that it improves the classification accuracy on the 80% unlabeled data, during the course of the experiment we can tune the three parameters to obtain the optimal result as Table 3 shows when m =0.45 k =3 n =1.

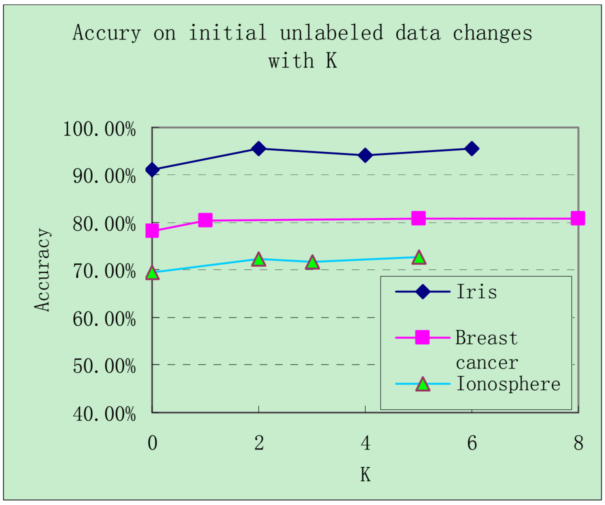


Figure3. When the other two parameters are fixed, the accuracy on initial unlabeled data changes along with the change of the K and we can choose the optimal value according to the above figure.

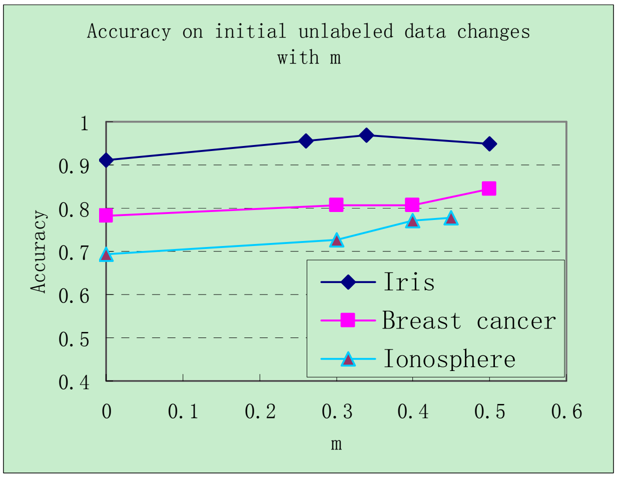


Figure4. When the other two parameters are fixed, the accuracy on initial unlabeled data changes along with the change of the m and we can choose the optimal value according to the above figure.

In all the experiments, the classes of unlabeled data are binary and the initial training set consists of the examples of two classes, during the course of experiment we set three parameters including k, m, n , k is the number of nearest neighbors, m is the percentage controlling the number of training data account for the number of whole data, n is the number of boundary vectors picked out from every class for each iteration. As depicted above we can tune the three parameters in order to gain better performance.

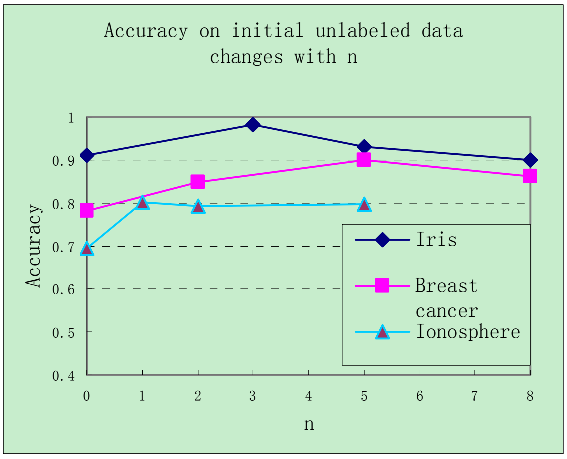


Figure5. When the other two parameters are fixed, the accuracy on initial unlabeled data changes along with the change of the n and we can choose the optimal value according to the above figure.

In a word, we propose a novel SVM-KNN classification methodology based on semi-supervised learning theory improve the accuracy of the final classifier. We only introduce part of the unlabeled-labeled data into the training set, but these unlabeled-labeled data are boundary vectors picked out from the classification boundary, the boundary vectors may be the support vectors so the final SVM classifier can predict the examples excellently.

Ⅴ. CONCLUSION

In this work we have described a classification method based on semi-supervised learning theory in which unlabeled data can be used to augment the training set and improve the accuracy of the final classifier. Our results support the statistical learning theory results that incorporating unlabeled data improves accuracy of the classifier when insufficient training information is available. In order to obtain better results the three parameters can be tuned conveniently according to practical environment. The preliminary experimental results presented suggest that this method of utilizing large number of unlabeled data has a potential for significant benefits in practice [6].

Many research questions remain. In the future we will study the classification problem with unlabeled data set in which the labels of unlabeled examples are unbalanced distributed and solve the multi-category unlabeled data classification, so the further work are clearly required.

ACKNOWLEDGMENTS

We would like to express our gratitude to all those who gave us help to complete this paper, especially for Mr Wei Zhang, Zheng Cao and Chao Zhang whose constructive suggestions and encouragements help us in all the time of research and writing of this paper.

REFERENCES

[1] X.J. Zhu. Semi-supervised learning literature survey[R]. Technical Report 1530, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, December, 2007.

[2] Miller, D. J., and Uyar, H. S. A mixture of experts classifier with learning based on both labeled and unlabeled data. Advance in NIPS 9.571–577, 1997.

[3] Fung, G., & Mangasarian, O. Semi-supervised support vector machines for unlabeled data classification (Technical Report 99-05). Data Mining Institute, University of Wisconsin Madison, 1999.

[4] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien. Semi- Supervised Learning [M]. The MIT Press, 2006

[5] Blum, A., and Mitchell, T. Combining labeled and unlabeled data with co-training. In COLT, 92–100, 1998.

[6] Zhou, Z.-H., Zhan, D.-C., & Yang, Q. Semi-supervised learning with very few labeled training examples. Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07), 2007.

[7] Bennett, K., & Demiriz, A. Semi-supervised support vector machines.Advances in Neural Information Processing Systems, 11, 368–374, 1999.

[8] Vapnik V. The Nature of Statistical Learning Theory [M]. New York: Springer-Verlag, 2000.

[9] P. S. Bradley and O. L. Mangasarian. Massive data discrimination via linear support vector machines. Technical Report Mathematical Programming Technical Report 98-05, University of Wisconsin-Madison, 1998.

[10] C. Burges, "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, vol. 2, 1998.

[11] V. N. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation and signal processing", Advances in Neural Information Processing Systems, vol. 9, Cambridge, Mass: MIT Press, 1997.

[12] Y. Q. Zhang and D. G. Shen, "Design efficient support vector machine for fast classification", Pattern Recognition, vol. 38, 157-161, 2005.

[13] G. Fung, O.L. Mangasarian, Semi-supervised support vector machines for unlabeled data classification, Optim. Methods Software15 (1) (2001) 29–44.

[14] S.R. Waterhouse, A.J. Robinson, Classification using hierarchical mixtures of experts, in: Proceedings of the 1994 IEEE Workshop on Neural Networks for Signal Processing, pp. 177–186, 1994.

[15] Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods [M]. Cambridge University Press, 2000.

[16] Keerthi, S. S. and C.-J. Lin (2003). Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation 15 (7), 1667{1689.

[17] Lin, H.-T. and C.-J. Lin (2003). A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Technical report, Department of Computer Science, National Taiwan University.

[18] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification.Technical report, National Taiwan University,2003.

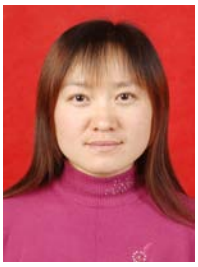
[19] Dasarathy, B. V., Nearest Neighbor (NN) Norms, NN Pattern Classification Techniques. IEEE Computer Society Press, 1990.

[20] Wettschereck, D., Dietterich, T. G. "An Experimental Comparison of the Nearest Neighbor and Nearest-hyperrectangle Algorithms," Machine Learning, 9: 5-28, 1995.

[21] Platt J C. Fast Training of Support Vector Machines Using Sequential Minimal Optimization [M]. Advances in Kernel Methods:Support Vector Machines (Edited by Scholkopf B,Burges C,Smola A)[M]. Cambridge MA: MIT Press, 185-208, 1998.

[22] UCI repository of machine learning databases: http://www.ics.uci.edu/mlearn/MLRepository.

KunLun Li, born in 1962, received the PhD degrees in Signal & Information Processing from Beijing Jiaotong University, China, in 2004 and join College of Electronic and Information Engineering of Hebei University as the associate professor at present. His main research interests include machine learning, data mining, intelligent network security and biology information technology. In these areas, he has published over 20 technical papers in refereed international journals or conference proceedings.



Xuerong Luo born in 1982, received her B.Sc. degree in College of Physics Science & Technology, Hebei University, Baoding, China. Currently she is a M.Sc. candidate in College of Electronics and information Engineering, Hebei University, Baoding, 071002, china. Her main research interests include pattern recognition and artificial intelligent, machine learning and data mining, information security.

Ming Jin born in 1983, received his B.Sc. degree in College of Physics Science & Technology, Hebei University, Baoding, 071002, China. Currently he is a M.Sc. candidate in School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, china. His main research interests include mobile telecommunication and signal process especially about system level simulation in TD-SCDMA and HSDPA.