A Comparison Study of Semi-supervised SVM Algorithms for Small Business Credit Prediction

Jie Zhang [#1], Lin Li [#2], Ge Zhu [#3], Xiangfu Meng [*4], Qing Xie [#5]

[#] *School of Computer Science & Technology, Wuhan University of Technology, Wuhan, China*

[1] xiaoyejiu@yeah.net

[2] cathylilin@whut.edu.cn

[3] ziwuyoulin@foxmail.com

[5] felixxq@whut.edu.cn

[*] *School of Electronic and Information Engineering, Liaoning Technical University, Huludao, China*

[4] marxi@126.com

***Abstract*—The small companies become increasingly important in bank's lending business. But the challenge is how bank's credit assessment is made in a small amount of time and money. Compare with the big companies, the small companies often need a small amount of cash flow. They may not provide the complete certificates or documents, so that the bank has to collect information of the companies and evaluate their credit rating especially by experts. For the bank, it is worthless to spend time and money to investigate a small company, especially just to lend several hundred thousand dollars. In the real life, credits of most the companies are good, while only small of them cannot repay for some reasons. The few number of small companies' credit data is valuable while considerable unknowing credit data of small companies is within reach. Therefore, the binary classification of the good credit and the bad credit is asymmetry. we choose supervised learning algorithm (Regularized Least Squares Classification and SVM) and semi-supervised learning algorithm (Transductive SVM and Deterministic Annealing Semi-supervised SVM) to predict the credits of small companies. In this paper, we conduct a series of experiments on credit datasets with different proportion classification and the results show that the Deterministic Annealing Semi-supervised SVM (DAS3VM) performance better when the data set is rare and asymmetry.**

I. Introduction

If a big company needs funds, it tends to get a bank loan. The company has to fill out the loan application about loan amount, loan purpose, repayment ability and repayment. There are many basic data and supporting information submitted along with an application, such as the corporate information, enterprise information assets, corporate financial statements, etc. Then, the bank's credit evaluation experts will select investigation items and formulate examine plans. They will evaluate the enterprise by its strength of the economic, capital structure, financial, operational efficiency, business prospects, etc.

But for many small companies, they may not provide the complete certificates or documents and they just need some funds for a short-time liquidity [1, 2]. The bank has to gather the information of the companies and evaluate their credit rating especially by the experts. For banks, it is unworthy to spend the same time and money on investigating a small company which will just borrow several hundred thousand dollars for a short time.

In the real life, credits of most the companies are good, while only small of them cannot repay for some reasons. We can only acquire a company's credit standing after it's loan from the bank. The few number of small enterprise' credit data is valuable while considerable unknowing credit data of small companies is within reach. It is convenient to acquire a company's basic information, online transaction data, product comments, etc. Therefore, the binary classification of the good credit and the bad credit is asymmetry.

In this paper, we briefly introduce the L_2-SVM-MFN, Transductive SVM (using L_2-SVM-MFN) and Deterministic Annealing Semi-supervised SVM (using L_2-SVM-MFN). The key idea of SVM is the classification hyperplane has to pass through the low data density region. And another constraint is that data points in each cluster on the same side of the hyperplane have the same labels. It bases on the assumption that points in a same cluster should have the similar labels. In the semi-supervised SVM, the role of the unlabeled data is to identify clusters and high density regions in the input space.

We conduct the experiment study on three different scale data sets and it clearly shows that the DAS3VM behaves better than the other three algorithm when the labeled data are small and asymmetry.

This paper is arranged as follows. In section 2 we describe the L_2-SVM-MFN algorithm and present semi-supervised SVM in section 3. In section 4 we illustrate the experiment. Section 5 contains our conclusion.

II. L_2-SVM-MFN

The problem is a binary classification with l labeled examples $\{x_i, y_i\}_{i=1}^{l}$, the input $x_i \epsilon R^n$ and the output $y_i \epsilon \{-1, +1\}$ and the classifier is $y = w^T x + b$. The original optimization problem to solve the standard SVM [3] is:

$$\min_{(w,b)} \frac{1}{2}(||w||^2 + b) + \frac{C}{2}\sum_{i}^{l} \xi^2$$
$$s.t. y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1, 2, \ldots, l \quad (1)$$

Where C is the regularization parameter.

L_2-SVM-MFN provides the following SVM optimization problem:

$$w^* = \underset{w\epsilon R^n}{\operatorname{argmax}} \frac{1}{2}\sum_{i}^{l} l_2(y_i w^T x_i) + \frac{\lambda}{2}||w||^2 \tag{2}$$

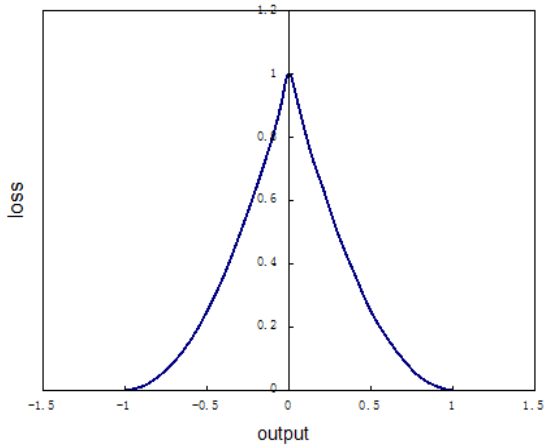

Fig. 1. l_2 loss function

Where l_2 is the loss function given by $l_2(f(x)) = \max(0, 1 - f(x))^2$, λ is a regularization parameter given by $\lambda = C^{-1}$ and the final classifier is $y = {w^*}^T x$.

In this function, we use the square of the Hinge Loss rather than the Hinge Loss for easier derivation and the b is regularized.

$$\min_w f(w) = \frac{1}{2}\sum_{i\epsilon I(w)} c_i l_2(y_i w^T x_i) + \frac{\lambda}{2}||w||^2 \tag{3}$$

Here we add constraint of the support vectors set $I(w) = \{i : y_i({w^*}^T x_i) < 1\}$ and the loss cost c_i. At the same time, if the index set $I(w)$ are independent of w and run over all data points, this would simply be the objective function for weighted linear regularized least squares (RLS) [4].

The gradient of f is given by X

$$\nabla f(w) = \lambda w + X_{I(w)} C_{I(w)}[X_{I(w)} w - Y_{I(w)}] \tag{4}$$

Here, $X_{I(w)}$ is a matrix and it's rows are the feature vectors of training examples with respect to the index set $I(w)$, $Y_{I(w)}$ is a column vector which composition labels of the examples, and $C_{I(w)}$is a diagonal matrix which diagonal are the costs c_i for these examples.

With the set $I \subset \{1, \ldots, m\}$, we define the function f_I as

$$f_I(w) = \frac{\lambda}{2}||w||^2 + \frac{1}{2}\sum_{i\epsilon I} c_i l_2(y_i w^T x_i) \tag{5}$$

We know that the f_I is a strictly convex quadratic function, so it has a unique minimizer.

The Newton method in [5] does the iterations of the form

$$w_{k+1} = w_k + \delta_k n_k \tag{6}$$

Here, the $\delta_k \epsilon R$, the Newton search direction $n_k \epsilon R^n$ is given by w_k:

$$n_k = -\nabla f(w_k)/[\nabla^2 f(w_k)] \tag{7}$$

Here, the $\nabla f(w_k)$ is the gradient vector and the $\nabla^2 f(w_k)$ is the Hessian matrix of f at w^k.

Algorithm 1 L_2-SVM-MFN

Input: Training set $\{x_i, y_j\}_{i=1}^{l}$.
Output: The optimize result w.

1.Choose a suitable w_0, set $k = 0$.

2.Check if w_k is the optimal solution of (3). If it is, end the algorithm with return w_k.

3.$I_k = I(w_k)$.

$$\overline{w} = \underset{w}{\operatorname{argmin}} f_{I_k}(w)$$

4. $L = \{w = w_k + \delta(\overline{w} - w_k) : \delta \geq 0\}$.

$$\delta^* = \underset{w\epsilon L}{\operatorname{argmin}} f(w)$$

Let $w_{k+1} = w_k + \delta^*(w - w_k)$, $k = k + 1$, go back to the step 2.

Algorithm L_2-SVM-MFN would converges in a limited number of iterations, it can be proved at [4].

III. SEMI-SUPERVISED LINEAR SVMS

Our data sets are l labeled examples $\{x_i, y_i\}_{i=1}^{l}$ and u unlabeled examples $\{x'_j\}_{j=1}^{u}$, the x_i,$x'_j \epsilon R^n$, $y_i \epsilon \{+1, -1\}$, $l \ll u$. Our goal is using the labeled examples and the unlabeled examples to construe a liner classifier sign $(w^T x)$.

A. Transductive SVM

We assume that the examples x_j are labeled with $y'_j \epsilon \{+1, -1\}$. The assumption that the classification hyperplane has to pass through the low data density region $\{x_i, y_i\}$, at the same time the unlabeled examples x_j all have the real labels, which means that the classification hyperplane also has to pass through the low data $\{x_i, y_i\}$ and $\{x_j, y'_j\}$. So the unlabeled examples can help to display the data distribution. Transductive SVM uses the unlabeled examples to help to drive the classification hyperplane to pass through the real low data density regions [6, 7].

The following optimization problem is the standard TSVM:

$$\begin{aligned} \min_{w,\{y'_j\}_{j=1}^{u}} & \frac{\lambda}{2}||w||^2 + \frac{1}{2l}\sum_{i=1}^{l} l(y_i w^T x_i) \\ & + \frac{\lambda'}{2u}\sum_{j=1}^{u} l(y'_j w^T x'_j) \\ s.t. & \frac{1}{u}\sum_{j=1}^{u} max[0, sign(w^T x'_j)] = r \end{aligned} \tag{8}$$

Here, we use the Hinge Loss, $l(f(x)) = max(0, 1 - f(x))$. λ is a regularization parameter, λ' is a parameter provided

by the users and it controls the influence of unlabeled data. For example, if we set λ' to 0, it would the standard SVM. The initial value of r can be get from the training set of the positive class in labeled examples and can be adjusted by the validation performance.

The optimization is implement in [8] by first using the inductive SVM to label the unlabeled data and designation a temp factor λ'^* then iteratively switching labels to minimize (8). Second uniformly increasing the value of the λ'^*. Then retraining SVM to improve the objective function until $\lambda'^* \geq \lambda'$ and the algorithm ends and output the result.

To use the L_2-SVM-MFN, we consider the TSVM object function with the L_2-SVM loss function, $l = l_2$. We can know from the [9, 10] about the TSVM with L_2-SVM-MFN and we use the L_2-SVM-MFN to train a classifier on labeled data, the unlabeled data are temporary labeled based on the classifier. Then we start from a small value of λ'^*, and pairs of unlabeled data with opposite temporary labels switching these labels to decrease the object function. We gradual increase λ'^* by a certain factor until $\lambda'^* \geq \lambda'$, the algorithm ends and output the result.

B. Deterministic Annealing

The TSVM loss function over the unlabeled examples is non-convex which makes it do not have a global optimal solution and has a high time complexity in the solving model. Deterministic Annealing [6, 11] based on the annealing process makes the optimization problem process into a series of temperature-dependent physical systems minimal of free energy function. Therefore, it can avoid local minimum and obtain the global minimum.

We first rewrite the TSVM object function:

$$\begin{aligned} w^* = &\underset{w,u_j{}_{j=1}^{u}}{\operatorname{argmin}} \frac{\lambda}{2}||w||^2 + \frac{1}{2l}\sum_{i=1}^{l} l_2(w^T x_i) \\ &+\frac{\lambda'}{2u}\sum_{j=1}^{u}(u_j l_2(w^T x'_j) + (1-u_j) l_2(-w^T x'_j)) \end{aligned} \tag{9}$$

Here, $u_j = (1+y_j)/2$, we relax the unlabeled data by $max[0, 1-|w^T x|]^2 = min[l_2(w^T x), l_2(-w^T x)]$. Then we rewrite the object function as following:

$$\begin{aligned} w_T^* = &\underset{w,p_j{}_{j=1}^{u}}{\operatorname{argmin}} \frac{\lambda}{2}||w||^2 + \frac{1}{2l}\sum_{i=1}^{l} l_2(y_i w^T x_i) \\ &+\frac{\lambda'}{2u}\sum_{j=1}^{u}(p_j l_2(w^T x'_j) + (1-p_j) l_2(-w^T x'_j)) \\ &+\frac{T}{2u}\sum_{j=1}^{u}[p_j \log(p_j) + (1-p_j)\log(1-p_j)] \\ &s.t. \frac{1}{u}\sum_{j=1}^{u} p_j = r \end{aligned} \tag{10}$$

Here, we relax the binary variables u_j to probability variables p_j and include entropy terms for the distributions defined by p_j. The r is the ratio of positive class in unlabeled examples. And we use the DA loss function replace the l_2 loss function, with the decrease of the temperature T. The loss function changes the shape from a squared-loss shape to the TSVM loss function. The minimizer is slowly obtained as the temperature is reduced to 0.

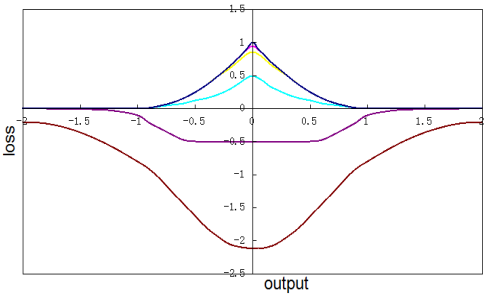

Fig. 2. DA loss function parameterized by T

The optimization is done with the decreasing of the T from a high value towards 0. For each T, we first fix the p and optimize the w with the L_2-SVM-MFN; next, we fix the w, construct the Lagrangian of (10) and let the partial derivative of p_j to 0 to optimize the p_j with a hybrid combination of Newton-Raphson iterations and the bisection method [12].

IV. EXPERIMENTAL RESULT AND ANALYSIS

We conduct supervised learning experiments and semi-supervised learning experiments on three different credit data sets.

A. Preparation for the Experiment

The data is about the enterprise credit information. We crawl the information form the National Business Credit Information publicity system and Alibaba[1]. We have collected millions of companies' basic information and product comments and others. The data sets we have collected are usually noisy. We find that during the crawling, some valuable companies' basic information has not proper storage or missing, such as the financial reports and product comments. When the companies information has been stored in our database, we randomly select some companies with credit history for experiment. We select some attributes of the companies and conduct the numerical operation. Some large attributes may affect the small attributes. In order to avoid this, we conduct normalization on some attributes.

TABLE I
TWO-CLASS CREDIT DATA SETS

data sets	$l+u$	r	f
credit-one	4255	0.674	17
credit-two	2589	0.464	17
credit-three	1924	0.289	17

We randomly divide the examples into three different class proportion data sets in Table I. $l+u$ represents the number

[1]http://gsxt.saic.gov.cn/, https://s.1688.com/company/company_search.htm

of labeled examples and unlabeled examples, f represents the numbers of attributes, r represents the positive class ratio.

In our experiment, we use the reverse k-fold cross validation to effectively avoid over-learning and the learning owe state so that the final result is also more persuasive. Reverse k-fold cross validation is a method similar to the k-fold cross validation where the data sets are divided into k groups and each group take turns to be chosen as the training set while the remaining $k-1$ groups together act as the test set. We have collected the accuracy evaluate result, the recall of positive examples, computation time and calculated the F-1 measure. The final experiment result is the average of the k times test results.

At this paper, we conduct experiments by using the Regularized Least Squares (RLS) Classification, SVM (L2-SVM-MFN), Transductive SVM (using L2-SVM-MFN), Deterministic Annealing Semi-supervised SVM (using L2-SVM-MFN) while set the $k = 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$ on every data sets.

B. Experimental Result

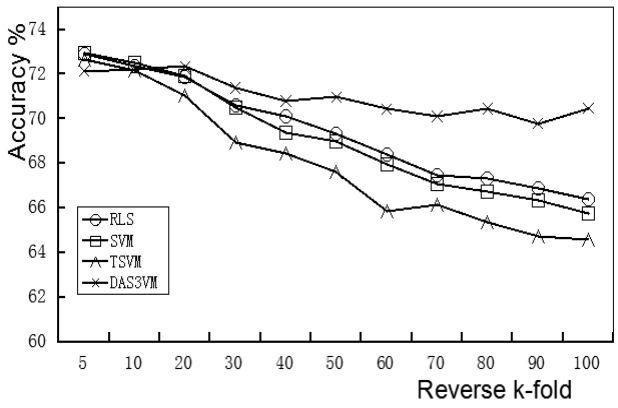

Fig. 3. RLS, SVM, TSVM, DASVM: The Accuracy evaluate result on credit-one for different k.

Here, we first conduct experiment on the data set credit-one and the accuracy evaluate results are shown in Figure 3. From the Figure 3, while the $k = 5, 10, 20$, the resultS of the four algorithms have the similar results. But with the increase of k, we can obviously find that the performance of the DAS3VM is better than other three algorithms. We consider the reason for this phenomenon is while the $k = 5, 10, 20$, the number of labeled examples is enough for the supervised learning algorithms; While the $k \geq 30$, the number of labeled examples is very few, especially when the value of $k = 50$, the training set only have 2% of the data set (85 labeled examples), the DAS3VM obviously performs better than other three algorithms. The reason is that it can gradually use the unlabeled examples to optimize the object function.

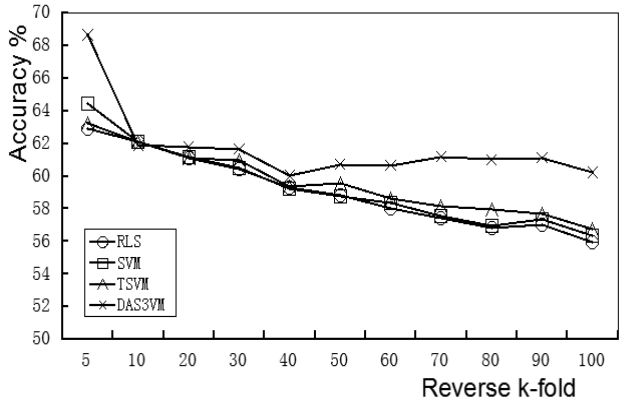

Fig. 4. RLS, SVM, TSVM, DASVM: The Accuracy evaluate on credit-two for different k.

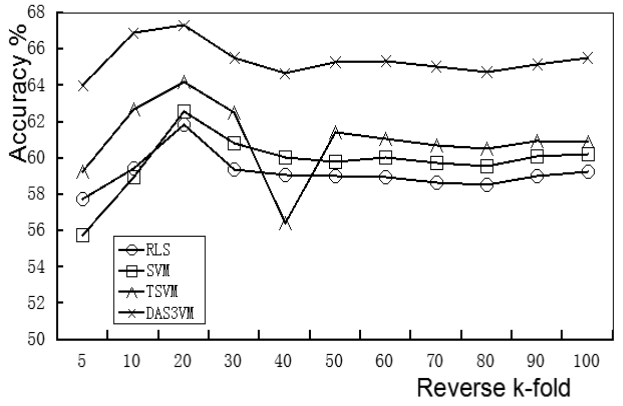

Fig. 5. RLS, SVM, TSVM, DASVM: The Accuracy evaluate on credit-three for different k.

In the Figure 4 and Figure 5, we find the same result as in Figure 3. DAS3VM performs better than other three algorithms if the number of labeled examples in training set is much less than than the test examples. Different with the Figure 3, TSVM performs better than other supervised learning algorithms in the most time.

The Figure 6 shows the accuracy of credit evaluate on the three different proportion of the two class. Obviously, the results on credit-one and credit-three are better than the credit-two. We hold the view that the DASVM performs better on the asymmetry data set (the majority of the examples are one of the class). And the three curves are very flat even if the labeled examples in the training set are difference in many times (if the $k = 10$, there are 10% labeled examples; if the $k = 50$, there are only 2% labeled examples.). So we believe that the DAS3VM is a very good algorithm on credit data.

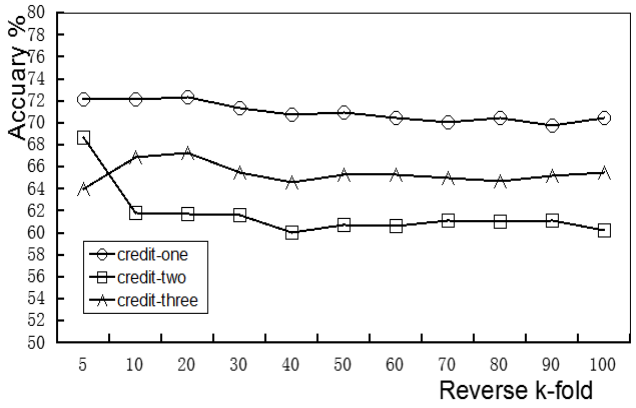

Fig. 6. DAS3VM:The Accuracy evaluate on credit-one, credit-two and credit-three for different k.

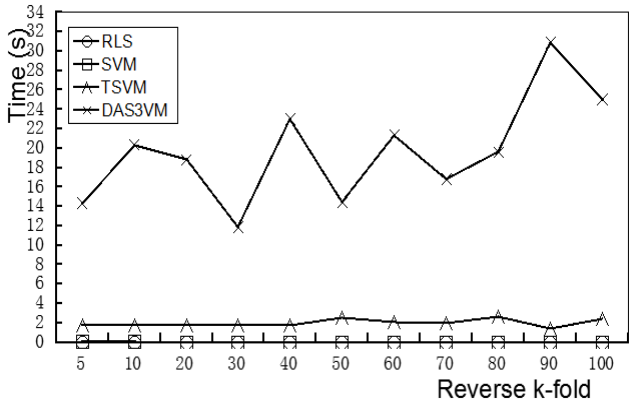

Fig. 7. RLS, SVM, TSVM, DASVM: The computation time on credit-one for different k.

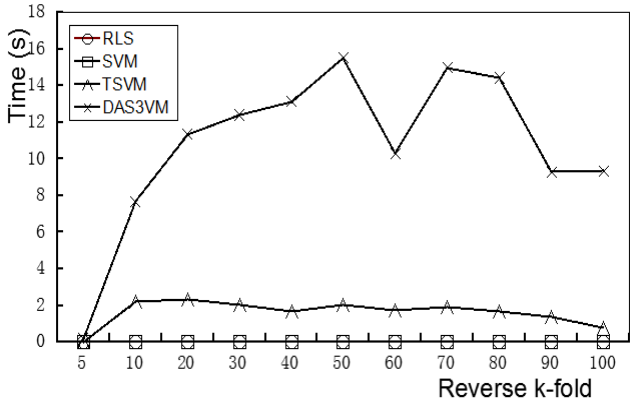

Fig. 8. RLS, SVM, TSVM, DASVM: The computation time on credit-two for different k.

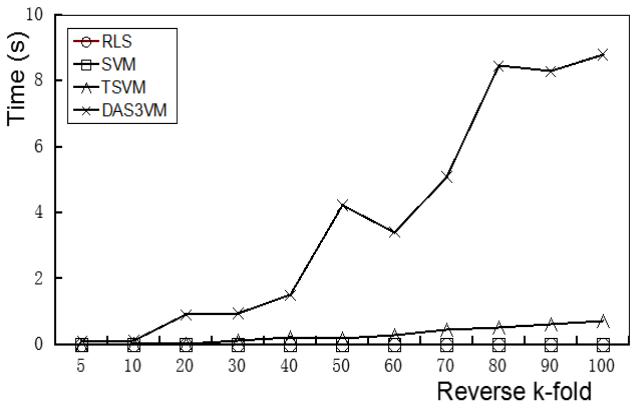

Fig. 9. RLS, SVM, TSVM, DASVM: The computation time on credit-three for different k.

TABLE II
RLS, SVM, TSVM, DASVM: THE RECALL ON THE CREDIT-ONE, CREDIT-TWO AND CREDIT-THREE

credit-one	RLS	SVM	TSVM	DASVM
k=5	93.28	93.16	85.87	88.37
10	92.15	91.98	85.14	86.31
20	90.89	90.67	84.05	85.71
30	89.20	88.68	81.80	87.05
40	88.29	86.69	81.04	89.54
50	87.56	86.66	80.19	87.01
60	85.21	83.39	77.68	87.60
70	83.79	82.46	78.11	87.95
80	83.6	81.48	77.32	86.64
90	81.82	80.36	76.26	87.42
100	81.47	79.53	76.05	87.59
credit-two	RLS	SVM	TSVM	DASVM
k=5	67.97	62.52	62.48	71.54
10	66.05	66.12	66.01	67.60
20	62.55	62.58	63.77	67.30
30	61.38	61.24	63.13	64.93
40	60.26	59.80	60.88	55.81
50	59.36	59.07	60.14	62.08
60	59.00	58.73	59.17	62.74
70	58.10	58.27	58.90	62.99
80	56.08	55.87	57.89	62.53
90	57.56	56.82	58.08	60.55
100	55.89	55.48	56.96	58.78
credit-three	RLS	SVM	TSVM	DASVM
k=5	54.14	52.31	39.22	48.05
10	44.83	50.28	43.27	43.33
20	43.61	47.85	42.52	48.63
30	41.97	43.66	39.87	33.62
40	41.74	42.50	52.18	33.07
50	41.2	42.06	38.16	33.56
60	40.52	41.34	37.38	32.90
70	39.70	40.49	36.84	32.13
80	38.53	39.46	36.48	30.05
90	38.75	39.77	37.03	31.44
100	37.93	38.62	36.50	29.02

The Figure 7, Figure 8 and Figure 9 show that no matter what value of k is, the computation time of DAS3VM is the longest followed by the TSVM. The supervised learning algorithms are the shortest and coincidence together in the vicinity of 0 seconds. In the section 3, we have briefly introduced the TSVM (using L_2-SVM-MFN) and the Deterministic

TABLE III
TWO-CLASS DATASETS

credit-one	$k=5$	10	20	30	40	50	60	70	80	90	100
RLS	81.84	81.06	80.26	78.81	78.17	77.37	75.90	74.76	74.59	73.60	73.15
SVM	81.83	81.10	80.21	78.57	77.09	76.83	74.89	73.97	73.37	72.69	71.98
TSVM	78.71	78.13	77.01	74.83	74.21	73.36	71.28	71.63	70.85	70.02	69.84
DAS3VM	79.43	78.63	78.46	78.44	79.07	78.18	**78.08**	**78.02**	**77.71**	**77.62**	**78.10**
credit-two	$k=5$	10	20	30	40	50	60	70	80	90	100
RLS	65.33	64.00	61.82	60.90	59.77	59.09	58.50	57.76	56.44	57.29	55.92
SVM	63.45	64.05	61.85	60.87	59.50	58.92	58.54	57.91	56.41	57.10	55.92
TSVM	62.87	64.01	62.40	62.04	60.10	59.84	58.90	58.52	57.91	57.87	56.84
DAS3VM	70.07	64.61	64.41	63.25	57.84	61.40	**61.68**	**62.07**	**61.76**	**60.84**	**59.50**
credit-three	$k=5$	10	20	30	40	50	60	70	80	90	100
RLS	55.89	51.11	51.16	49.18	48.91	48.53	48.04	47.35	46.47	46.78	46.25
SVM	53.98	54.27	54.23	50.82	49.76	49.39	48.96	48.27	47.46	47.88	47.06
TSVM	47.20	51.19	51.16	48.69	54.23	47.07	46.38	45.85	45.52	46.06	45.64
DAS3VM	54.89	52.59	56.47	44.43	43.76	44.34	43.77	43.01	41.05	42.42	40.22

Annealing Semi-supervised SVM (using L_2-SVM-MFN). We know that the two semi-supervised learning algorithms are more complex in object optimize compare with the RLS and SVM. And the experiment results also prove the DAS3VM cost more time in computation.

Here, we display the recall and F-1 measure of the four algorithms on three data sets in Table II and Table III. As shown in Table II shown, the recall of the DAS3VM on the credit-one is better than the three algorithms if $k \geq 40$. And on the credit-two it perform better whether the $k = 5, 10$ or 50. But on the credit-three, the supervised learning algorithms perform better. The F measure take the both accuracy and recall into account, so it can better show the experiment results of the algorithms. In the Table III, the F-1 measure also indicate the DAS3VM perform good on the small business credit prediction of the credit-one and credit-two.

V. CONCLUSION

In this paper, we present the algorithm Deterministic Annealing Semi-supervised SVM and conduct several experiment on the small business credit data sets. The results shown in this paper all indicate that the DAS3VM performs good on the small business credit data sets, especially in the situation that the labeled examples are few.

REFERENCES

[1] A. N. Berger and W. S. Frame, "Small business credit scoring and credit availability," *Journal of small business management*, vol. 45, no. 1, pp. 5–22, 2007.

[2] A. N. Berger and G. F. Udell, "Small business credit availability and relationship lending: The importance of bank organisational structure," *The economic journal*, vol. 112, no. 477, pp. F32–F53, 2002.

[3] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998.

[4] S. S. Keerthi and D. DeCoste, "A modified finite newton method for fast solution of large scale linear svms," *Journal of Machine Learning Research*, vol. 6, no. Mar, pp. 341–361, 2005.

[5] O. L. Mangasarian, "A finite newton method for classification," *Optimization Methods and Software*, vol. 17, no. 5, pp. 913–929, 2002.

[6] V. Sindhwani, S. S. Keerthi, and O. Chapelle, "Deterministic annealing for semi-supervised kernel machines," in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006)*, 2006, pp. 841–848.

[7] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Advances in Neural Information Processing Systems 11*, 1998, pp. 368–374.

[8] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, AISTATS 2005*, 2005.

[9] O. Chapelle, V. Sindhwani, and S. S. Keerthi, "Optimization techniques for semi-supervised support vector machines," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 203–233, 2008.

[10] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[11] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, 1998.

[12] V. Sindhwani and S. S. Keerthi, "Large scale semi-supervised linear svms," in *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 477–484.