Transaction Cost Analysis via Label-Spreading Learning

Pangjing Wu(✉) and Xiaodong Li

College of Computer and Information, Hohai University, Nanjing, China
{pangjing.wu,xiaodong.li}@hhu.edu.cn

Abstract. When the investment institution analyzes the transaction cost of stock orders, it is costly to obtain the transaction cost of the stock orders by trading it. In contrast, many simulated trading orders cannot get the exact transaction cost. Due to the lack of enough labeled data, it is usually hard to use a supervised learner to estimate accurate transaction cost of stock orders. Label-spreading, a graph-based semi-supervised learner, can integrate a small number of labeled real orders and a large number of unlabeled simulated orders, and train a learner simultaneously. Using a RBF kernel, the learner constructs a graph structure through the spatial similarity measure between the transaction cost samples, and propagates the label through edges of graph in high-dimensional space. The results of experiments show that the label-spreading learner can make full use of the information of unlabeled data to improve classification of transaction cost.

Keywords: Label-spreading · Semi-supervised learning · Algorithm trading

1 Introduction

When investment institutions analyze the transaction cost of the stock orders, accurate data of transaction cost is required for investment institution to analyze the transaction cost. However, it is costly to obtain the transaction cost of stock orders by trading it. Besides, the large amount of simulated data generated by the stock trading simulator is simulated estimation, which lacks real value and cannot obtain the exact transaction cost of stock orders. Therefore, transaction cost data is not enough and accurate for investment institutions to analyze the order transaction cost by supervised learner.

This paper intends to provide the solution to the problem that when the transaction data of stock orders is scarce and hard to mark, the graph-based

This work was supported in part by the National Key R&D Program of China under Grant 2018YFC0407901, in part by the National Natural Science Foundation of China under Grant 61602149, and in part by the Fundamental Research Funds for the Central Universities under Grant 2019B15514.

X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 444–454, 2019.
https://doi.org/10.1007/978-3-030-32388-2_38

semi-supervised learner can be applied to study transaction data of stock orders, which make full use of the readily available unlabeled data and generalize a well-performed classifier to predict transaction cost.

Semi-supervised learning can learn both labeled data and unlabeled data. Its basic principle is to improve the learning effect by using many unlabeled data to assist a few labeled data. As early as 1970s, some scholars tried to use the unlabeled samples to improve the performance of the classifier. Based on generative model theory, they proposed the first semi-supervised learning model and EM algorithm [4] to solve the model. Subsequently, in 1990s, Vapnik proposed TSVM [9] based on the idea that the decision boundary should keep the maximum distance from the labeled and unlabeled samples, which had a greater impact on the early semi-supervised learning model. And then, Blum and Mitchell proposed co-train model [2] from the perspective of learning-views. Limited to TSVM is a non-convex optimization problem and the assumptions of co-train are harsh, these methods have difficulties in practical applications, so people began to try other methods for semi-supervised learning. Based on the graph theory, scholars proposed a series of graph-based semi-supervised learners such as min-cut [1], local and global consistency theory [12], label propagation [13]. Compared with the earlier methods, most of the learners are convex, which means that the global optimal solution can be easily obtained. In addition, the calculation is based on matrix operations, which is efficient and easy to understand and implement. In recent years, graph-based semi-supervised learners have received extensive attention in research and application [3,5,7].

The rest of this paper will be organized as follows. Section 2 describe the dataset and select features and labels. We build the label-spreading model in Sect. 3 and predict the transaction cost label for the stock orders in Sect. 4. Section 4.2 analyzes the properties of the label-spreading learner further. And the conclusions will be presented in Sect. 5.

2 Data Sets of Stock Orders

The data sets used for the experiment is consisted of a transaction data set of real stock orders $L = \left\{ (\boldsymbol{x}_1, y_1), \ldots, \left(\boldsymbol{x}_{|L|}, y_{|L|}\right) \right\}$ and a transaction data set of simulation stock orders $U = \left\{ \boldsymbol{x}_{|L|+1}, \ldots, \boldsymbol{x}_{|L|+|U|} \right\}$. L contains 93 items of transaction information, and U contains 3776. The data in L and U are derived from Charles River Advisors Ltd. [6]. For ease of calculation, let $l = |L|, u = |U|, \ n = l + u$.

2.1 Features Selection

As shown in Table 1, the transaction data sets of stock orders contain six trading features.

Since the deviation between the weighted price and average price can reflect the trading market of the order. The Intraday VWAP, TWAP and Full Day VWAP are the weighted prices of stocks. Therefore, we calculate the relative

Table 1. Features in transaction data sets of stock orders

Feature	Meaning
Average	Average transaction price per share
Liq. Consumption	Order volume accounts for the size of stock liquidity
Execution Time	The time elapsed from the start to the completion of execution
Intraday VWAP	Intraday volume weighted average price
TWAP	Time-weighted average price
Full day VWAP	Full day volume weighted average price

deviation between the three features and average price according to Eq. 1, which is used as the features for learning. Where features include Intraday VWAP, TWAP, Full day VWAP. Thus, the features for semi-supervised learning are $\boldsymbol{x} = \{$Liq. Consumption, Execution Time, Δ Intraday VWAP, Δ TWAP, Δ Full day VWAP$\}$.

$$Features = \frac{Features - Average}{Average} \tag{1}$$

To eliminate the deviation generated by different dimensions and of unit each feature, this paper use the Z-score equation to standardize the data. Calculate the mean value $\bar{\boldsymbol{x}}$ and the variance $\boldsymbol{\sigma}$ of the features in L. Then, according to Eq. 2, obtain the standardized features L_x of data set L and the standardized features U_x of data set U.

$$Z(\boldsymbol{x}_i) = \frac{x_i - \bar{\boldsymbol{x}}}{\boldsymbol{\sigma}},\ i = 1, \ldots, n. \tag{2}$$

2.2 Label of Transaction Cost

Intraday VWAP slippage is used in data set L to describe the transaction cost of stock orders, which indicates the deviation of Intraday VWAP between the trading algorithm transmit a buy or sell signal and actually completes the buy or sell.

For the buyer, intraday VWAP slippage greater than 0 means that the intraday VWAP when actually completes the buy transaction is lower than the trading algorithm transmit signal. Stock is bought at a lower than expected price, and the transaction cost decreases. On the contrary, the transaction cost increases when slippage is less than 0. The situation is reversed for the seller. Slippage greater than 0 means stock is sold at a lower than expected price, that the transaction cost increases. On the contrary, when slippage less than 0, the transaction cost decreases. In the data set mentioned above, we have changed the sign of slippage according to the direction of buying and selling. So that intraday VWAP slippage greater than 0 indicates the transaction cost of decreases, and less than 0 indicates the transaction cost increases.

Because the intraday VWAP slippage is the actual value, so the transaction cost label y needs to be initialized according Eq. 3 before classifying.

$$y = \begin{cases} 1, & \textit{Intraday VWAP Slippage} \geq 0; \\ 0, & \textit{Intraday VWAP Slippage} \leq 0; \\ -1, & \textit{Unlabeled.} \end{cases} \tag{3}$$

3 Label-Spreading Learner

Based on L and U, we construct graph $G = (V, E)$. Where, $V = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is consisted by $\boldsymbol{x}$, and edge set is represented by the affinity matrix $\mathbf{W}$.

In the graph-based semi-supervised learner, the k-nearest neighbor (k-NN) kernel and the radial basis function (RBF) kernel are commonly used as graph kernel. In the graph using the k-NN kernel, the node only establishes the joint edge with its k neighbors, and only reflects the local relationship of each node. The affinity matrix $\mathbf{W}$ is a sparse matrix, and the calculation speed is relatively fast. In the graph of RBF kernel, the node establishes edge with all other nodes to form a complete graph, which fully reflects the global relationship of each node, but the defect is that the calculation speed is slow. In order to ensure that the semi-supervised learner has good generalization, this paper uses radial basis function as graph kernel. Substitutes $L \cup U$ into Eq. 4 to calculate affinity matrix $\mathbf{W}$ and construct graph structure. Where, let RBF parameter $\gamma = \frac{1}{2\sigma^2} = 1$ according to experiment.

$$(\mathbf{W})_{ij} = \begin{cases} \exp(\dfrac{-\|x_i - x_j\|_2^2}{2\sigma^2}), & i \neq j; \\ 0, & \textit{Otherwise.} \end{cases} \tag{4}$$

We assume that in the complete graph $G = (V, E)$ constructed by $L \cup U$, a real-valued function $f : V \rightarrow R$ can be obtained by learning. So that f approaches the true label at the labeled nodes and has smoothness on entire graph. According to the study by Zhu et al. [14], adjacent nodes on the feature space should have similar labels, so the energy function of f is defined using the quadratic energy function. Where, $\boldsymbol{f}$ is the prediction of the transaction cost. The diagonal matrix $\mathbf{D} = diag\,(d_1, \ldots, d_n)$ whose element $d_i = \sum_{j=1}^{n} (\mathbf{W})_{ij}$ is the sum of the elements of the i-th row of affinity matrix $\mathbf{W}$.

$$\begin{aligned} E(f) &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{W}_{ij})(f(x_i) - f(x_j))^2 \\ &= \boldsymbol{f}^T (\mathbf{D} - \mathbf{W}) \boldsymbol{f}. \end{aligned} \tag{5}$$

By minimizing the energy function (Eq. 5), f can approaches the true label at the labeled nodes and has smoothness across the graph, i.e. $\forall \boldsymbol{x}_i \in L,\ f(\boldsymbol{x}_i) = y_i,\ \boldsymbol{\Delta f} = \mathbf{0},\ \boldsymbol{\Delta} = \ \mathbf{D} - \mathbf{W}$. In order to minimize the energy function easily, we split affinity matrix $\mathbf{W}$ into 4 blocks after l-th row and column,

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{ll} & \mathbf{W}_{lu} \\ \mathbf{W}_{ul} & \mathbf{W}_{uu} \end{bmatrix},\ \mathbf{D} = \begin{bmatrix} \mathbf{D}_{ll} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{uu} \end{bmatrix}. \tag{6}$$

The energy function (Eq. 5) can be expressed as

$$E(f) = \boldsymbol{f}_l^T(\mathbf{D}_{ll} - \mathbf{W}_{ll})\boldsymbol{f}_l - 2\boldsymbol{f}_u^T\mathbf{W}_{ul}\boldsymbol{f}_l + \boldsymbol{f}_u^T(\mathbf{D}_{uu} - \mathbf{W}_{uu})\boldsymbol{f}_u. \tag{7}$$

Let

$$\mathbf{P} = \mathbf{D^{-1}W} = \begin{bmatrix} \mathbf{D}_{ll}^{-1}\mathbf{W}_{ll} & \mathbf{D}_{ll}^{-1}\mathbf{W}_{lu} \\ \mathbf{D}_{uu}^{-1}\mathbf{W}_{ul} & \mathbf{D}_{uu}^{-1}\mathbf{W}_{uu} \end{bmatrix}, \tag{8}$$

there is $\mathbf{P}_{uu} = \mathbf{D}_{uu}^{-1}\mathbf{W}_{uu}, \mathbf{P}_{ul} = \mathbf{D}_{uu}^{-1}\mathbf{W}_{ul}$. From $\frac{\partial E(f)}{\partial \boldsymbol{f}_u} = 0$ can get

$$\begin{aligned} \boldsymbol{f}_u &= (\mathbf{D}_{uu} - \mathbf{W}_{uu})^{-1}\mathbf{W}_{ul}\boldsymbol{f}_l \\ &= (\mathbf{D}_{uu}(\mathbf{I} - \mathbf{D}_{uu}^{-1}\mathbf{W}_{uu}))^{-1}\mathbf{W}_{ul}\boldsymbol{f}_l \\ &= (\mathbf{I} - \mathbf{D}_{uu}^{-1}\mathbf{W}_{uu})^{-1}\mathbf{D}_{uu}^{-1}\mathbf{W}_{ul}\boldsymbol{f}_l \\ &= (\mathbf{I} - \mathbf{P}_{uu})^{-1}\mathbf{P}_{ul}\boldsymbol{f}_l \end{aligned} \tag{9}$$

Then, according to the established graph structure, take the label information $\boldsymbol{f}_l = (y_1; y_2; , \ldots ; y_l)$ into Eq. 9, and the prediction result $\boldsymbol{f}_u$ of the unlabeled data can be obtained.

Furthermore, Zhou et al. [11] proposed a regularization framework equivalent to Eq. 9, i.e. label-spreading learner.

$$\min_F \frac{1}{2}\left(\sum_{i,j=1}^{l+u}(\mathbf{W})_{ij}^2\left\|\frac{1}{\sqrt{d_i}}\mathbf{F}_i - \frac{1}{\sqrt{d_j}}\mathbf{F}_j\right\|^2\right) + \mu\sum_{i=1}^{l}\|\mathbf{F}_i - \mathbf{Y}_i\|^2. \tag{10}$$

The learner learns labels by minimizing the loss function with regularization characteristics, which is more robust to noise in most cases. Where, $\mathbf{F}$ is a non-negative label matrix. $\mathbf{Y}$ is the matrix with actual label of data set L. the regularization parameter $\mu = \frac{1-\alpha}{\alpha}, \alpha \in [0, 1]$ is clamping factor specified by user. The first item in Eq. 10 forces similar samples to have similar labels, and the second item forces the learning result to be as identical as possible to the real label on the labeled sample.

4 Experiments

4.1 Transaction Cost Classification

Let $\hat{y}$ be the transaction cost label predicted by the label-spreading learner. Because the actual transaction cost label y_U in U is unknown, the learning effect of the learner cannot be evaluated by comparing y_U and $\hat{y}_U$. So that, this paper compares the prediction labels of the test set $\hat{y}_{TE}$ and the actual label y_{TE} to evaluate the learning effect of the learner. 80 data are randomly extracted from L as the labeled part of training set L_{TR}, and the remaining 13 data are used as the test set L_{TE}.

In order to reduce the of influence of special case, we slack the limit of the label of labeled samples, making the clamping factor $\alpha = 0.1$ (i.e., the regularization parameter $\mu = 9$) in Eq. 10. Some labeled samples are allowed to be assigned incorrect labels, so that the labels in graph are smoother.

Take L_{TR} and U into the label-spreading learner (Eq. 10) to learn and predict the label $\hat{y}_{TE}$ of L_{TE}. Because the graph-based semi-supervised learning is transductive, the process of graph construction can only consider the training set $L_{TR} \bigcup U$ and cannot judge the position of the new sample in graph. Using label-spreading learner to predict the label of newly added test data L_{TE} is essentially added L_{TE} to the training set as an unlabeled sample, then reconstruct the graph and propagate label information on the graph to obtain the test set label $\hat{y}_{TE}$. Therefore, based on the above experimental ideas, we can simplify the process of graph construction. Set the label in L_{TE} to unlabeled state and participate in the construction of the graph structure with the training set $L_{TR} \bigcup U$. When the iteration of learner converges, $\hat{y}_{TE}$ is obtained, which saves the tedious calculation of reconstructing, and the experiment takes about 0.803s.

To avoid accidental errors, the experiment was repeated 100 times, and the mode of the transaction cost label $\left\{ \hat{y}_{TE}^{(i)} \middle| , i = 1, 2, \ldots, 100 \right\}$ predicted in 100 experiments was taken as the transaction cost prediction label $\hat{y}_{TE}$. The learning effect of the label-spreading is evaluated by calculating the f1-score value of $\hat{y}_{TE}$, which shown in Table 2.

Table 2. Label-spreading classification report

Label	Precision	Recall	F1-score
0	0.73	1.00	0.84
1	1.00	0.40	0.57
Weight average	0.83	0.77	0.74

From the experimental results, the weighted average f1-score reached 0.74, which means the label $\hat{y}_{TE}$ predicted by label-spreading for the test set L_{TE} is more consistent with the actual label y_{TE}. Because the stock market fluctuations have complex properties such as uncertainty, chaos, and abruptness, the internal mechanism relationship is very complicated [8]. Moreover, in practical applications, when the weighted average f1-score is greater than 0.5, the classifier can be considered to have a good learning effect. Therefore, it is feasible to use label-spreading to classify and predict the transaction cost of stock orders.

Among the results, the recall of the class 0 is 1, indicating that all transaction cost loss orders in L_{TE} have been correctly labeled, and the precision of the class 1 is 1, indicating all transaction cost data labeled as 1 in L_{TE} are classified correctly. This learner can effectively reduce the risk of loss of order transaction and be sensitive to loss risk when predict the transaction cost. It can correctly label all orders with rising transaction costs and ensure that the actual situation marked as a profit order sample is profitable.

4.2 Analysis

Graph Kernel. In label-spreading learner, the k-nearest neighbor kernel and the radial basis function kernel are commonly used as graph kernel. In Sect. 3, we use the RBF kernel of the graph to improve the learning effect of label-spreading learner. Compared with the complete graph based on the RBF kernel, the graph structure based on the k-NN kernel is sparse and has certain advantages in calculation speed. Below we will build a graph based on the k-NN neighbor kernel and analyze the prediction effect of the label-spreading based on k-NN kernel in the transaction cost prediction experiment.

We take the nearest neighbors $k = \{5, \ldots, 9\}$ and train label-spreading learner based on different k. To avoid accidental errors, the experiment was repeated 20 times. The average of results is shown in Table 3.

When $k = 8$, the label-spreading learner based on k-NN kernel has the best learning effect on transaction cost, and the f1-score is 0.47. However, the f1-score is only 63.5% of the learner based on RBF kernel, which means that the learning effect of label-spreading learner based on k-NN kernel is generally inferior to the learner based on RBF kernel. Although the calculation speed of the learner based on k-NN kernel is about 271.5% faster, from the perspective of guaranteeing investors income, the label-spreading learner should be based on RBF kernel when predicting the transaction cost of stock orders.

Table 3. Label-spreading classification report

k	Precision	Recall	F1-score	Time cost
5	0.36	0.54	0.43	0.216
6	0.36	0.54	0.43	0.217
7	0.36	0.54	0.43	0.217
8	0.38	0.62	0.47	0.218
9	0.36	0.54	0.43	0.219

Parameters of Label-Spreading Learning. The setting of parameters affects the learning effect and convergence speed of learner. In the label-spreading learner based on the RBF kernel, the parameters that need manually set are the clamping factor α and the RBF parameter γ. The effect of parameter changes on label-spreading learner is analyzed by changing one parameter and fixing the values of other parameters. The learning effect is measured by the weighted average f1-score of L_{TE}, and the convergence speed of learner is evaluated by the running time of code.

First, we fixed the RBF kernel parameters, let $\gamma = 1$, and analyze the influence of the change of the parameter α on the learner. Because $\alpha \in [0, 1]$ and usually take less than 0.1, the exponential function with base 10 is used to select the clamping factor α, i.e. let $\alpha = 10^a, a \in [-5, 0)$and a set of α is obtained in steps of 0.25, i.e. $\boldsymbol{A} = \{\alpha_i = 10^{-5+\frac{i}{4}}|, i = 0, \ldots, 20\}$. To avoid accidental errors, the experiment was repeated 10 times for each α_i. The average of results is shown in Fig. 1.

As shown in Fig. 1, The clamping factor has less effect on the prediction of the learner. When the clamping factor is in a reasonable range, i.e. $\alpha \in \left[10^{-5}, 10^{-1}\right]$, the prediction of label has no significant change, and the prediction performance of the learner is greatly reduced when α is close to 1. For the calculation speed of the learner, when $\alpha \in [10^{-5}, 10^{-1}]$, the time for training is roughly the same, and it increases dramatically when α is close to 1.

Then, let the $\alpha = 0.1$ to analyze the effect of changes in γ on the learner. Similarly, since the value of γ often in $[0, 1]$, so let $\gamma = 10^g,\ g \in [-2, 2]$. In a step of 0.2, get a set of RBF parameter $\boldsymbol{\Gamma} = \{\gamma_i = 10^{-2+\frac{i}{5}}|, i = 0, \ldots, 20\}$. To avoid accidental errors, the experiment was repeated 10 times for each γ_i. The average of results is shown in Fig. 1.

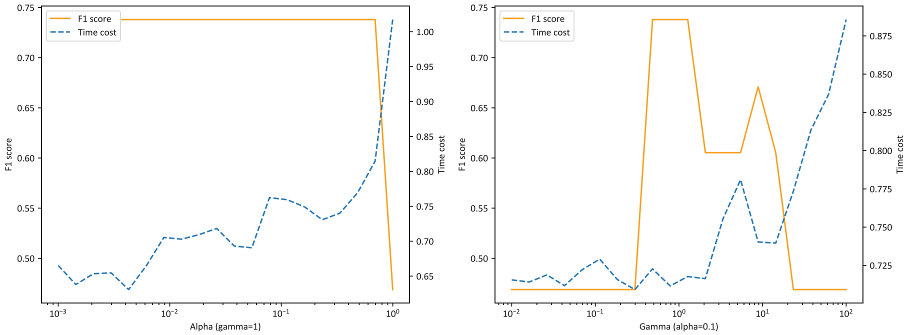

Fig. 1. Analysis of parameters of label-spreading learner.

Contrast with Supervised Learning. In theory, the information provided by unlabeled data can improve the learning effect [10]. In order to analyze the improvement of learning effect by unlabeled data in label-spreading, a typical supervised learning learner, support vector classifier (SVC), is selected for comparison, which is based on RBF kernel.

In data set$L$, the number of samples of class 0 is 61, and the number of samples of class 1 32. There is a large deviation in the number of samples of the two classes, which will cause SVC to make the decision boundary offset during classification and resulting in incorrect classification results. So that the class weight should be set to reduce the deviation. According to Eq. 11, class 0 weight is 0.34, and a class 1 weight is 0.56.

$$w_i = 1 - \frac{l_i}{l},\ i \in \{0, 1\}. \tag{11}$$

In order to improve the learning effect of SVC, we use grid search with cross-validation to search for optimal parameters. Because the correlation between the transaction orders is not obvious, it is assumed in the cross-validation that each order is independently distributed. It can be known from the SVC principle that the penalty factor C and the RBF parameter γ are crucial to the learning effect. So, let $C = 10^c, c \in [0,3], step = 0.15$; $\gamma = 10^g, g \in \left[10^{-2}, 10^2\right], step = 0.25$. Perform a 10-fold cross-validation on the parameter grid consisting of C and γ to search for the optima parameter (C^*, γ^*). The cross-validation score is represented by the weighted average f1-score and show in Fig. 2.

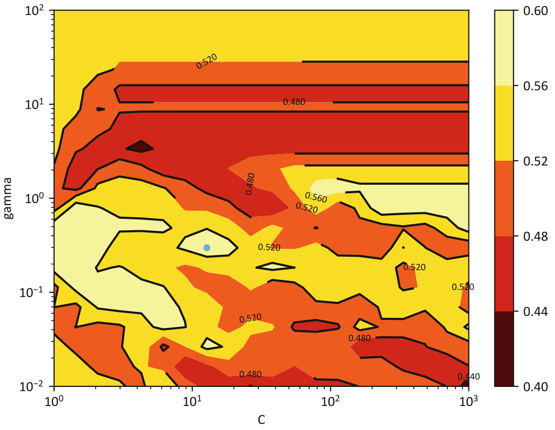

Fig. 2. Weighted f1-score on the grid of parameter space, blue dots are the parameters with the best score. (Color figure online)

According to the optimal parameter(C^*, γ^*), set SVC and train L_{TR}, then predict labels of L_{TE} to evaluate the learning effect of SVC. As Table 4 show, there is a large error in using SVC to predict transaction costs, and the weighted average f1-score is only 0.39. Moreover, the f1-score of class 1 is 0, which means in cannot predict the decline in transaction costs.

Table 4. SVC classification report

Label	Precision	Recall	F1-score
0	0.55	0.75	0.63
1	0.00	0.00	0.00
Weight average	0.34	0.46	0.39

Comparing the classification results of label-spreading and SVC in Table 5, it is obviously that the learning effect of label-spreading is significantly better than

SVC. Especially the label-spreading learner based on RBF kernel, the weighted average f1-Score is 89.7% more than SVC. It can be seen that a large number of easy-to-obtain simulated order data can improve the learning effect of predicting transaction cost.

Table 5. Comparison of learners

Learner	Precision	Recall	F1-score	Time cost
Label-spreading (RBF)	0.83	0.77	0.74	0.803
Label-spreading (knn)	0.38	0.62	0.47	0.219
SVC	0.34	0.46	0.39	2.272

5 Conclusions

This paper uses label-spreading algorithm to predict the transaction cost of stock orders. It can accurately classify the transaction costs of stock orders when the actual transaction data is scarce and difficult to label. The label-spreading learner integrate a small number of labeled real orders and a large number of unlabeled simulated orders, and train a learner simultaneously, thus the transaction cost classifier based on label-spreading has sufficient generalization performance. It can make full use of unlabeled data to improve the classifier and has a good learning effect on the transaction cost of stock orders, whose performance is far superior to the supervised classifier SVC.

The label-spreading model can construct a graph structure by k-NN kernel or RBF kernel. Compared with the RBF kernel, the graph built by k-NN kernel has a faster calculation speed, but the prediction effect is not good. In order to pursue the optimal prediction effect and ensure the return of investors, the label-spreading learner based on the RBF kernel should be used in analyzing and predicting the transaction cost of stock orders.

References

1. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of the Eighteenth International Conference on Machine Learning (2001)
2. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Conference on Computational Learning Theory (1998)
3. Bo, P., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the Association for Computational Linguistics (2004)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. **39**(1), 1–38 (1977)

5. Goldberg, A., Zhu, X.B.T.: Graph-based semi-supervised learning for sentiment categorization. In: Workshop on Graph Based Methods for Natural Language Processing (2006)
6. Charles River Advisors Ltd. (2014). https://www.crd.com/
7. Niu, Z.Y., Ji, D.H., Tan, C.L.: Word sense disambiguation using label propagation based semi-supervised learning. In: Proceedings of the ACL (2005)
8. Sun, B.W.: Research on the complexity of stock market volatility. In: Complexity Problems-Proceedings of Ccast (2001)
9. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995). https://doi.org/10.1007/978-1-4757-2440-0
10. Zhang, C.G., Yan, Z.: Semi-Supervised Learning (2013)
11. Zhou, D., Bousquet, O., Lal, T.N., Weston, J.: Learning with local and global consistency. In: International Conference on Neural Information Processing Systems (2004)
12. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Olkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems (2003)
13. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data. Technical report, vol. 3175, no. 2004, pp. 237–244 (2002)
14. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC, pp. 912–919 (2003)