Neural Comput & Applic (2018) 30:661–670
https://doi.org/10.1007/s00521-016-2684-y

ORIGINAL ARTICLE

Investor sentiment identification based on the universum SVM

Wen Long[1,2,3] · **Ye-ran Tang**[1,2,3] · **Ying-jie Tian**[1,2,3]

Received: 22 August 2016 / Accepted: 25 October 2016 / Published online: 21 November 2016

Abstract Universum refers to additional samples which contain priori knowledge for classification but belonging to none of the class. It has been proved that universum positioned "in between" the two classes obtain better results. Since opinions on stock market defined as investor sentiment involve quite a number of neutral views, these neutral views can be used as universum samples to better identify investor sentiment. With universum samples, this paper uses support vector machine (SVM) to classify posts on stock forum. We define bullish views as positive samples, define bearish views as negative samples, and also further discuss the situation of a 3-class problem with neutral views. Compared with standard SVM, the empirical studies with universum samples in this paper show better performance for both 2- and 3-class classifications.

Keywords Universum SVM · Investor sentiment · Text mining · Classification

✉ Ying-jie Tian
tyj@ucas.ac.cn

[1] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, People's Republic of China

[2] School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, People's Republic of China

[3] Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, People's Republic of China

1 Introduction

Investor sentiment which refers to the opinion of stock market has been a research hot spot of behavioral finance in recent years. The stock pricing model based on hypothesis of rational economic man infers that the price of a stock is determined by the discounted future dividends. Since the market stay rational, investor sentiment has little influence on stock market. However, researches on behavioral finance show several irrational phenomena may be provided by investor sentiment in stock market, and these irrational deviations are systemic. Instead of institutional sentiment, the vast majority of literatures related to investor sentiment tend to regard individual investors as sentiment traders [1–4].

To study investor sentiment and its effects on stock markets, it is necessary to identify individual's view on future trend of stocks. Previous literatures mainly contain 3 types of indicators to measure investor sentiment: the investor sentiment index from surveys, the indirect indicator from historical trading, and the investor sentiment index from internet information. The studies with surveys mainly use a variety of consumer sentiment survey index and consumer confidence index as a proxy variable of investor sentiment [5–8]. It is a direct indicator to obtain investors' opinions, but it takes a large amount of time and costs for survey. The indirect indicators based on historical information of stock market include historical prices, trading volume of stocks, etc. [9–15]. This method saves time and costs, but it is an indirect indicator of investor sentiment and cannot reflect investors' opinion directly. Besides, this method is limited by the variable selection and synthesis methods. Further, because the investor sentiment is obtained from historical data, it has hysteresis characteristic and can hardly reflect new information. The

high penetration of internet and the rapid development of data mining technology provide a way to identify investor sentiment directly and conveniently. Thus, applying text mining technology to obtain individual sentiment on stock forum has attracted much attention. Antweiler and Frank [16] used Naive Bayes algorithm and standard SVM to classify 1.5 million comments on Yahoo Finance based on manually annotation training set containing 1000 samples. Das and Chen [17] used several methods such as word count method, Bayesian classifier, etc. to analyze 300–500 comments as training set on Yahoo Finance from July 2001 to August 2001. Kim and Kim [18] used Naive Bayes algorithm with 4000 comments as training set to classify more than 32 million comments on Yahoo Finance from January 2005 to December 2005. Wu et al. [19] used SVM on 30,000 manually labeled reviews for 3-class sentiment classification. In general, standard SVM is widely used to classify investor sentiment from stock forum in financial studies. However, quite a number of neutral views involved in investor sentiment make it difficult for classification. In previous literatures, some of studies directly used 2-class classification method, which would reluctantly classify neutral views into positive or negative samples and reduce reliability of investor sentiment identification. Some other studies used 3-class classification method to separate views into positive, negative or neutral points, but the accuracy they calculate is the total accuracy of the 3 categories including neutral samples. Actually, the importance of 3 categories is different for the purpose of investor sentiment identification. We care more about accuracy of positive and negative samples, and it would be better if the neutral samples are just used as auxiliary part for classification. To solve this problem, this paper introduces universum support vector machine algorithm (U-SVM) to use these neutral points as universum samples to identify investor sentiment, since universum are additional samples belonging to none of the class and it has been proved that universum positioned "in between" the two classes to help obtain better results for classification.

Universum attracts wide attention as it contains priori knowledge for classification. It obtains additional information for a certain problem to be solved, and it belongs to none of the class. Take a handwritten digits recognition task for example, to distinguish digit "5" and "8", the other 8 digits excluding these 2 numbers can be set as universum samples. Compared with semi-supervised classification, both universum learning method and semi-supervised method contain unlabeled samples. However, unlike semi-supervised classification, universum sample does not belong to any class, whereas the unlabeled input in semi-supervised learning belongs to a certain class, although which class is not known in advance. Vapnik [20, 21] proposed the idea of universum and introduced it as an algorithm for SVM. Weston et al. [22] conducted the first experiments on training SVM with universum and showed the accuracy improvements with universum samples. They called the algorithm they proposed U-SVM. Sinz et al. [23] analyzed U-SVM algorithms and suggested that a good universum set was positioned "in between" the two classes. Cherkassky and Dai [24] and Cherkassky et al. [25] studied effectiveness of the U-SVM for high-dimensional data and found it depended on the distribution of universum samples relative to standard SVM decision boundary. Dhar and Cherkassky [26] extended such conclusions from Cherkassky et al. [25] with different misclassification costs. Previous studies on U-SVM algorithm has evaluated effectiveness and characters of appropriate universum samples. Besides using universum samples in standard SVM, several other machine learning methods with the universum have been proposed [27–37], which provide further evidence of the effectiveness of the universum. As for application, U-SVM method is widely used for the classification problem when training dataset contains additional samples belonging to none of the class that we are interested in. Gao et al. [38] used U-SVM to recognize translation initiation sites for protein sequences extraction. Chen and Zhang [39] conducted experiments with U-SVM on handwritten digits and human faces. Jiao et al. [40] classified tongue images using U-SVM. Hao and Zhang [41] applied U-SVM method to neuroimaging-based Alzheimer's disease classification studies. All of these applications have suggested great performance of universum samples. Because it has been proved that neutral samples located between the two classes (positive and negative) are more likely to obtain better results, whether it contains labeled neutral samples in dataset becomes a problem when using U-SVM algorithm. Alternatively, some researchers use random averaging samples which are generated by a pair of random positive and negative training samples and regard their average as universum if there is no labeled neutral sample in dataset.

The idea of universum is to use prior knowledge in additional samples. Several methods add new samples into original training set to improve classification performance. Semi-supervised method requires unlabeled samples to have the same distribution with original samples. In noise injection method, the added samples need subject to a different distribution. However, U-SVM does not need such assumptions about the distribution. Moreover, because we care more about accuracy of positive and negative samples than neutral samples, compared with standard 3-class SVM, U-SVM we use considers neutral sample as an auxiliary part for classification and is more suitable for our purpose. Since investor sentiment involves quite a number of neutral views which may influence classification and U-SVM algorithm can make good use of

these neutral samples as universum to improve classification, this paper uses support vector machine with universum samples to classify the posts on stock forum. We define bullish views as positive samples, bearish views as negative samples, and neutral views as universum samples. We compare the classification accuracy of U-SVM with those of standard SVM. Besides, we further discuss the situation of a 3-class problem to identify neutral views for out-of-sample prediction in financial studies.

2 Background

2.1 Support vector machine

It is widely acknowledged that support vector machines (SVMs) introduced by Vapnik et al. in 1990s [20, 42, 43] are powerful classification methods, and they are widely used in variety of fields [44–48]. Their mathematical representations, geometrical explanations, generalization abilities, and empirical performance make SVMs useful in a large amount of classification applications [49]. The goal of SVM is to learn an appropriate decision function to classify new samples after training with a labeled dataset.

Consider a classification problem in n-dimensional space with l training samples. The training samples can be defined as:

$$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}, \tag{1}$$

where $x_i \in R^n, i = 1, \ldots, l$, and for a binary classification problem, $y_i \in \{1, -1\}$. The goal is to identify a new sample x belonging to which class (1 or -1) after training with dataset T. To solve this problem, a decision function $f(x)$ is needed to separate the R^n space into 2 regions:

$$f(x) = \mathrm{sgn}(g(x)), \tag{2}$$

where $g(x)$ is a real function to obtain the value of y for each x. Particularly, for a linear classification problem, $g(x)$ can be a linear function:

$$g(x) = w \cdot x + b, \tag{3}$$

and the corresponding hyperplane is $w \cdot x + b = 0$.

For nonlinear separation, an appropriate map Φ is needed to transform an n-dimensional vector x into another m-dimensional vector in space R^m. Thus, the maximal soft-margin algorithm of the SVM deduces the following primal optimization problem:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & y_i(w \cdot \Phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, \ldots, l, \end{aligned} \tag{4}$$

where C is a penalty parameter and ξ_i represent the slack variables. In this paper, we use a nonlinear kernel function, and set $\Phi(x_i) \cdot \Phi(x_j) = K(x_i, x_j)$. A convex quadratic programming problem can be constructed as:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{j=1}^{l}\alpha_j \\ \text{s.t.} \quad & \sum_{i=1}^{l} y_i \alpha_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l, \end{aligned} \tag{5}$$

where α_i are Lagrangian multipliers. After obtaining the solution $\alpha^* = (\alpha_1^*, \ldots, \alpha_l^*)^T$, the optimal separating hyperplane can be given by:

$$\begin{aligned} g(x) &= \sum_{i=1}^{l} y_i \alpha_i^* K(x_i, x) + b^*, \\ b^* &= y_i - \sum_{i=1}^{l} y_i \alpha_i^* K(x_i, x_j). \end{aligned} \tag{6}$$

Then, a new sample is classified as 1 or -1 according to the decision function of formula (2).

2.2 Support vector machine with universum

A dataset of universum is a collection of additional samples known to belong to none of the class, and it contains priori knowledge for classification. The structural risk minimization principle in standard SVMs is to choose an appropriate decision function after finding a set of candidate decision function F, and it contains no prior knowledge for the learning task. Support vector machine with universum constructs a data-dependent structure on the set of admissible functions using universum samples. Compared with defining a distribution explicitly, obtaining a set of universum samples is more convenient for a learning task.

According to Cherkassky and Dai [24], Fig. 1 is an illustration of SVM with universum. Since universum belongs to none of the class, when using maximal margin algorithm, universum samples would better fall inside the margin borders. Take Fig. 1 for example, if margin width of the 2 hyperplanes are same, Hyperplane II is better than

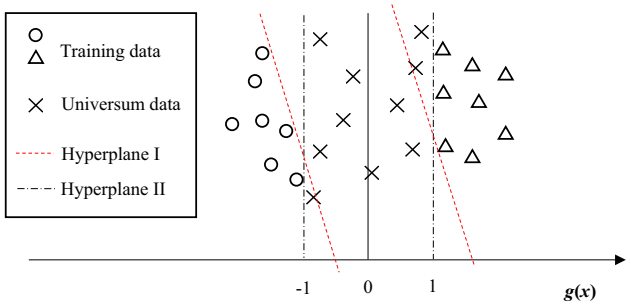


Fig. 1 Illustration of SVM with universum

Hyperplane I as it contains a larger number of universum samples fall inside the margin borders. Thus, training SVM with universum should both use maximal soft-margin algorithm and maximize the amount of universum samples distributed near the hyperplane.

Consider a training set given additional universum samples:

$$T = \{(x_1, y_1), \ldots, (x_l, y_l)\} \cup \{x_1^*, \ldots, x_u^*\}, \tag{7}$$

where $x_j^* \in R^n, j = 1, \ldots, u$ represent universum samples. Since universum samples reflect prior knowledge of the classification task by approximating them equivalent to hyperplane $g(x) = 0$, the primal optimization problem of maximal soft-margin algorithm of universum SVM (U-SVM) is set as:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C_t \sum_{i=1}^{l} \xi_i + C_u \sum_{s=1}^{u} \left(\psi_s + \psi_s^*\right) \\ \text{s.t.} \quad & y_i(w \cdot \Phi(x_i) + b) \geq 1 - \xi_i, \\ & -\varepsilon - \psi_s^* \leq w \cdot \Phi\left(x_s^*\right) + b \leq \varepsilon + \psi_s, \\ & \xi_i \geq 0, \quad i = 1, \ldots, l, \\ & \psi_s, \psi_s^* \geq 0, \quad s = 1, \ldots, u, \end{aligned} \tag{8}$$

where C_t is a penalty parameter and ξ_i is a slack variable for training samples (positive and negative). C_u is a penalty parameter, ψ_s, ψ_s^* are slack variables, and ε denotes ε-insensitive loss for universum samples. The U-SVM algorithm in formula (8) maximizes the margin between the separating hyperplanes, and it also maximizes the number of universum samples distributed near the hyperplane. Specially, if $C_u = 0$, formula (8) can be regarded as a standard SVM. The dual problem of U-SVM can be constructed as:

$$\begin{aligned} \min_{\alpha,\mu,v} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K\left(x_i, x_j\right) \\ & + \frac{1}{2}\sum_{s=1}^{u}\sum_{t=1}^{u} (\mu_s - v_s)(\mu_t - v_t) K\left(x_s^*, x_t^*\right) \\ & + \sum_{i=1}^{l}\sum_{s=1}^{u} \alpha_i y_i (\mu_s - v_s) K\left(x_i, x_s^*\right) \\ & - \sum_{i=1}^{l} \alpha_i + \varepsilon \sum_{s=1}^{u} (\mu_s + v_s) \text{s.t.} \\ & \sum_{i=1}^{l} y_i \alpha_i + \sum_{s=1}^{u} (\mu_s - v_s) = 0, \\ & 0 \leq \alpha_i \leq C_t, \quad i = 1, \ldots, l, \\ & 0 \leq \mu_s, \ v_s \leq C_u, \quad s = 1, \ldots, u, \end{aligned} \tag{9}$$

where α_i, μ_i, v_i are Lagrangian multipliers. Here, we also choose an appropriate kernel function $K\left(x_i, x_j\right)$. We get $\alpha^* = \left(\alpha_1^*, \ldots, \alpha_l^*\right)^{\mathrm{T}}$, $\mu^* = \left(\mu_1^*, \ldots, \mu_u^*\right)^{\mathrm{T}}$, $v^* = \left(v_1^*, \ldots, v_u^*\right)^{\mathrm{T}}$ by solving formula (9). Then, the optimal separating hyperplane with priori knowledge of universum can be given by:

$$\begin{aligned} g(x) &= \sum_{i=1}^{l} y_i \alpha_i^* K(x_i, x) - \sum_{s=1}^{u} \left(v_s^* - \mu_s^*\right) K\left(x_s^*, x\right) + b^*, \\ b^* &= y_i - \sum_{i=1}^{l} y_i \alpha_i^* K\left(x_i, x_j\right) + \sum_{s=1}^{u} \left(v_s^* - \mu_s^*\right) K\left(x_s^*, x_j\right). \end{aligned} \tag{10}$$

3 Data and preprocessing

3.1 Dataset of investor sentiment

The data which reflect individual investor sentiment are collected from online posts in a large-scale stock forum called Eastmoney stock forum, which is one of the largest stock forum in China. Eastmoney is a financial portal founded in 2004, and its stock forum's visits and posts have reached a certain scale since the year of 2011. It has become the largest and the most influential financial portal whose effective visit time is accounted for 43.8% of total effective visit time in financial portals. In July 2016, the ranking of Eastmoney in financial portals is 2 according to China Websites Ranking organized by Internet Society of China. Considering the activity degree of online discussion in different sectors, energy sector is selected as a representation in this paper. We get a random sample of 5990 online posts of 24 stocks which are the constituent stocks of CSI 300 energy index as samples.

The 5990 unstructured reviews are manually classified as bullish views, bearish views and neutral views by financial researchers. To ensure the reliability of labels, we accept majority opinions of 5 financial researchers on a certain post. We define bullish views as positive samples, bearish views as negative samples, and neutral views as neutral samples. As a result, the 5990 samples contain 1010 positive samples, 1212 negative samples and 3768 neutral samples.

3.2 Text data preprocessing

For classification, unstructured text reviews should be changed into digital data for computer processing. We call such a process reprocessing. The standard preprocessing methods contain data cleaning, text representation, and feature extraction.

3.2.1 Data cleaning

Since investors often discuss whatever they want on stock forum, the text data there contain a lot of punctuation, noise, etc., and cannot be directly used for analysis. Therefore, we use data cleaning technology to eliminate punctuations and gibberish. We also use some preprocessing technologies for Chinese text specially, such as words segmentation.

3.2.2 Text representation

Text representation is a technology to change text information into digital data. We use N-gram method based on vector space model. Vector space model [50] is one of the text representation methods based on and extending the bag of words model [51], and it is commonly used in preprocessing of text classification. Bag of words model assumes that each word is independent, and it represents a text as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. The N-gram method based on vector space model we use is considering that the relation of adjacent words may probably have effects on classification. Parameter N here represents number of words related to a certain word. That is, the emergence of the N-th word is associated with $N-1$ words in front of it. In this paper, we set $N = 1, 2, 3$. With N-gram method based on vector space model, each text is expressed as a vector in the language space, and each feature gets the weight according to its importance in the text.

When weighting for each feature, Salton et al. [52] suggested that the importance of the word can be reflected by Boolean, frequency, or TF–IDF method. Boolean method simply divides a word into two parts by distinguishing whether it appears in the text. Frequency method considers the frequency of the word and allows computing a continuous weight. TF–IDF is the method that multiplies term (word) frequency and inverse document frequency together, and it assumes the word that often appears in a certain text and seldom appears in whole document is important for classification. In this paper, we use all of these 3 methods for word weighting.

3.2.3 Feature extraction

Feature extraction method is used to remove the features that contribute little for analysis, which may reduce computing complexity and avoid over fitting. If the total number of the features in samples is large and some features are even rarely involved in texts, it can be assumed that these infrequent features have little contribution to classification, and can be ignored. In that case, the feature extraction in this paper is to extract the features whose occurrence time is no less than a certain number. The dimensions of samples in different minimum occurrence times after data cleaning and text representation are shown in Table 1. N-gram from 1 to 3 represents number of words related in a certain feature for vector space model, and Min occurrence from 1 to 5 represents minimum occurrence times of the extracted features. It can be seen that, when minimum occurrence time is small, the feature dimension is high, which may raise computing complexity. When minimum occurrence time is large, the feature dimension is low, which may lose important information for classification. Considering both computing complexity and integrality of information, we set minimum occurrence time as 3 in this paper.

Table 1 Feature dimensions in different preprocessing methods

N-gram	1	2	3
Min occurrence = 1	16,820	85,932	160,581
Min occurrence = 2	6497	11,318	13,164
Min occurrence = 3	4128	5489	5848
Min occurrence = 4	3085	3772	4027
Min occurrence = 5	2470	2943	3159

Table 2 Ranges of parameters

Parameter	Value
C for formula (5)	$[10^{-5}, 10^{5}]$
C_t for formula (9)	$[10^{-5}, 10^{5}]$
C_u for formula (9)	$[10^{-5}, 10^{5}]$
ε for formula (9)	[0, 1]

4 Experiment

We use 5990 preprocessed samples as training set. It contains 1010 positive samples, 1212 negative samples and 3768 neutral samples. The neutral samples in this paper are used as universum. The ranges of parameters for grid search are shown in Table 2.

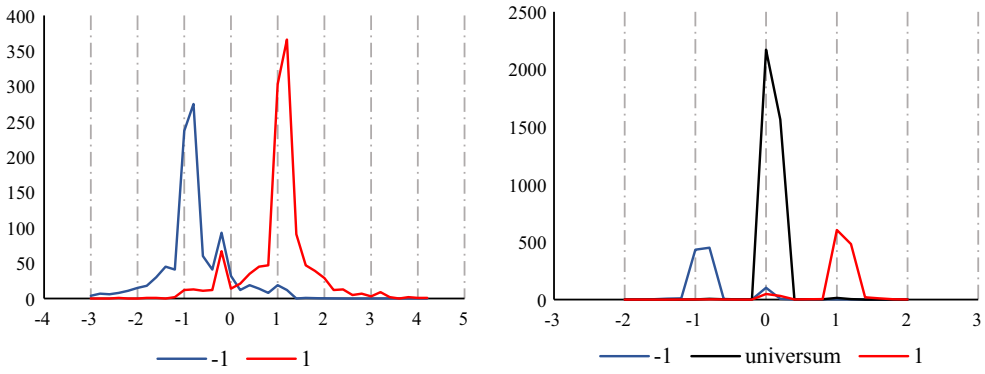
With fivefold cross-validation, we compare accuracy of SVM and U-SVM in different preprocessing methods. Accuracy we use to evaluate the performance of the classifier is as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{11}$$

where TP represents true-positive prediction, TN is true-negative prediction, FP refers to false-positive prediction, and FN denotes false-negative prediction. Thus, numerator

Table 3 Accuracy of SVM and U-SVM

N-gram	1		2		3	
	SVM (%)	U-SVM (%)	SVM (%)	U-SVM (%)	SVM (%)	U-SVM (%)
TF-IDF	70.93	***72.23***	70.88	***72.28***	70.93	***72.55***
Frequency	70.03	**70.48**	69.76	**70.25**	69.80	**70.12**
Boolean	69.85	**70.43**	69.71	**70.25**	69.67	**70.16**



(a) SVM **(b)** U-SVM

Fig. 2 Histogram of projections onto normal direction of hyperplane. **a** SVM, **b** U-SVM

of formula (11) represents the number of correct predictions, and denominator refers to total number of predicted records. The results are shown in Table 3. TF–IDF, Frequency, and Boolean in Table 3 refer to different text representation methods mentioned in part 3.

In Table 3, the bold refers to accuracy of U-SVM performs better than SVM, and the bold italic refers to accuracy of TF-IDF is better than Frequency and Boolean methods.

It can be seen that U-SVM performs better than SVM in all the cases. Particularly, among 3 text representation methods, TF–IDF performs observably better than other 2 methods. Thus, we use TF–IDF text representation method for further discussion. To visually analyze the results, we use the method proposed by Cherkassky and Dai [24] to generate the histogram of projections of samples onto the normal direction vector of the hyperplane for both SVM and U-SVM. We first, respectively, calculate $g(x)$ in formula (6) for training samples of standard SVM and calculate $g(x)$ in formula (10) for both training samples and universum samples of U-SVM, which provides projections onto the normal direction vector of the hyperplane. Then, we generate the histogram by dividing its range into around 15 different bins. The interval of each bins is 0.2. The histograms for samples of 1-gram with TF-IDF as an example are shown in Fig. 2.

Figure 2 illustrates the effect of universum on classification. Samples using standard SVM without universum are not separated well, and the samples near the part where 2 lines intersect will be misclassified. However, when using U-SVM, positive samples and negative samples are separated more clearly. It suggests better performance when using U-SVM method, which testifies the theoretical results that a data-dependent structure on the set of admissible functions with universum samples can improve learning performance. Besides, universum samples are intensive distributed near the hyperplane, which provide us convenience for expanding the problem to 3 classes to identify neutral views for out-of-sample prediction in further discussion.

For detailed description, we sample training sets of size 50, 100, 200, 500, and 1000 from original dataset, and compare accuracy for different training subset sizes with different sample sizes of universum. The sample sizes of universum are 0, 200, 500, and 1000. Particularly, when size of universum is 0, it is a standard SVM. The results for samples of 1-gram with TF-IDF are shown in Table 4. The percentages in brackets represent differences of accuracy compared with one row above. We also test other datasets for different text representation methods and obtain the similar results.

The results in Table 4 suggest a better performance of U-SVM compared with SVM in all these 5 different sizes of training samples. Further, an obvious advantage for U-SVM is obtained when training sizes are small. When training sizes are 50 and 100 in Table 4, compared with standard SVM, accuracy of U-SVM in 3 different sizes of universum is at least more than 3.92 and 7.02%, respectively. It is mainly because the information provided by universum samples makes up for the lack of training data, which proves that universum belonging to none of the class contain priori knowledge for classification. Besides, the

Table 4 Accuracy for different training subset sizes

Training sizes	50	100	200	500	1000
SVM	48.66 %	52.17 %	61.97 %	64.29 %	68.24 %
U-SVM 200	52.58 %	59.19 %	62.11 %	65.45 %	69.15 %
	(+3.92 %)	**(+7.02 %)**	(+0.14 %)	(+1.16 %)	(+0.91 %)
U-SVM 500	54.05 %	60.98 %	62.51 %	65.91 %	69.80 %
	(+1.47 %)	(+1.79 %)	(+0.40 %)	(+0.46 %)	(+0.65 %)
U-SVM 1000	55.99 %	61.55 %	62.56 %	66.32 %	69.39 %
	(+1.94 %)	(+0.57 %)	(+0.05 %)	(+0.41 %)	**(−0.41 %)**

Bold values indicate results that are discussed in the main text

accuracy of different sample sizes for universum suggests that advantage provided by universum may not be the sustainable growth with the increasing number of universum samples. Take 1000 training size for example, compared with 500 universum samples, the accuracy of 1000 universum samples is reduced instead (−0.41%). A possible explanation is that since universum follow a certain distribution, when number of universum samples arrives at a certain level, there is almost no more information which new universum samples could provide. Thus, the sample size of universum should consider both accuracy and redundancy.

5 Further discussion

We have already observed a better performance of U-SVM compared with standard SVM in binary classification problem to divide investor sentiment into bullish (positive) and bearish (negative). However, for out-of-sample prediction, there are a large amount of neutral views that cannot be identified as either positive samples or negative samples. Above experiments in this paper define neutral samples as universum which are only used in training process and will not be classified in prediction, but in this part, in order to well identify investor sentiment in financial application, we not only need to identify positive and negative samples, but also need to separate neutral samples. Thus, in this part, we discuss the situation of a 3-class problem to identify neutral views for out-of-sample prediction. Firstly, we discuss the empirical separating hyperplane construction of U-SVM for 3-class classification and then analyze its effectiveness.

5.1 Empirical separating hyperplane of U-SVM for 3-class classification

In formula (10), $g(x)$ is a real function to obtain the value of each input x for classification. In binary classification, separating hyperplane is often set as $g(x) = 0$. Thus, if $g(x) > 0$, we classify samples as positive samples, and if $g(x) < 0$, we classify samples as negative samples. One way to separate neutral samples for prediction is expanding separating hyperplane, which means, introducing the parameter a and setting separating hyperplanes as $g(x) = \pm a$. Samples will be classified as positive samples if $g(x) > a$, as negative samples if $g(x) < -a$, or as neutral samples otherwise.

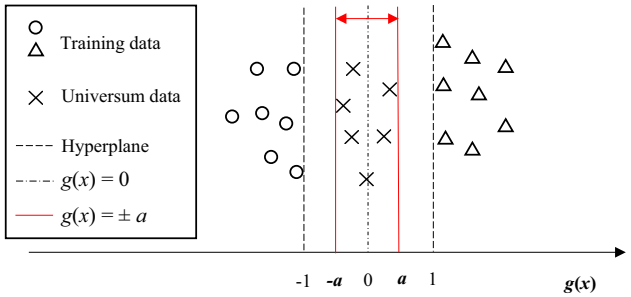


Fig. 3 Illustration of expanding empirical separating hyperplane

Idea of expanding separating hyperplane is illustrated in Fig. 3. The 2 solid lines in Fig. 3 represent expanding separating hyperplanes $g(x) = \pm a$, and universum data can be separated with appropriate parameter a via $-a < g(x) < a$. The value of parameter a is set as 1 commonly (as dashed line called Hyperplane in Fig. 3), because in training process, one of the goals is to maximize the number of universum samples fall inside the margin borders $-1 < g(x) < 1$. However, since the projections onto normal direction of hyperplane $g(x) = 0$ for universum samples are intensively distributed near the hyperplane (Fig. 2b) and soft-margin algorithm makes some of positive and negative samples beyond their borders, $-1 < g(x) < 1$ may not be the best choice to identify neutral samples. Instead, it is possible to classify neutral views well via different value of parameter a in different dataset. Thus, we use empirical method to determine an appropriate parameter a for a certain classification problem. Accuracy for such a 3-class classification of different parameter a is shown in Table 5. The results verify our inference that $-1 < g(x) < 1$ is not the best decision function to identify neutral samples. Instead, the best accuracy is obtained when $a = 0.2$ in all these 3 cases compared with other value of parameter a. Thus,

Table 5 Accuracy in different separating hyperplanes

a	0.1 (%)	0.2 (%)	0.3 (%)	0.4 (%)	0.5 (%)
1-gram	67.25	70.43	70.03	69.18	68.20
2-gram	67.23	70.45	70.13	69.33	68.11
3-gram	66.58	70.17	69.88	69.07	67.68

Table 6 Performance of U-SVM and SVM for 3-class classification

U-SVM				SVM			
Estimation	Original label			Estimation	Original label		
	−1	0	1		−1	0	1
−1	233	**106**	**33**	−1	308	**178**	**102**
0	702	3474	667	0	583	3363	587
1	**75**	**188**	512	1	**119**	**227**	523
Total	1010	3768	1212	Total	1010	3768	1212
Accuracy	70.43 %			Accuracy	70.02 %		
MAE	0.314			MAE	0.337		

Bold values indicate better results of U-SVM than SVM method

$g(x) = 0.2$ are the empirical separating hyperplanes of U-SVM for 3-class classification in this dataset.

5.2 Performance of U-SVM for 3-class classification

To analyze effectiveness of U-SVM for 3-class classification using above empirical separating hyperplanes, we compared its performance with standard 3-class SVM. The results are shown in Table 6. Accuracy of U-SVM is 70.43%, and accuracy of standard SVM is 70.02%. When comparing accuracy, U-SVM is not obviously better than standard SVM. However, when recognizing investor sentiment in financial studies, effect of misclassifying investor sentiment to neutral view or to its opposite view is different. One is considering mean absolute error. If we label positive samples as 1, negative samples as −1, and neutral samples as 0, the absolute error for misclassifying positive samples to negative samples is 2, while misclassifying it to neutral samples is 1, and vice versa. Here, we introduce mean absolute error (MAE) to measure the performance of classifiers:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{12}$$

where y_i represents actual label, $\hat{y}_i$ denotes predicted label, and n is total number of predicted records. Another explanation is that after investor sentiment identification, most of the studies would compose investor sentiment index for the subsequent financial studies. The formula for sentiment index composition is often as follows:

$$M_t = \ln\left[\frac{1 + M_t^{\mathrm{BUY}}}{1 + M_t^{\mathrm{SELL}}}\right], \tag{13}$$

where M_t^{BUY} represents total number of bullish posts (positive samples) in time interval t, and M_t^{SELL} represents total number of bearish posts (negative samples). The neutral posts play a relatively weak role for sentiment index composing in formula (13). Thus, compared with misclassifying a certain investor sentiment to its opposite view, misclassifying it to neutral view is acceptable instead.

We compare the estimated value and actual value of both U-SVM and standard SVM, and find a better performance of U-SVM. As shown in Table 6, Estimation means results of classification based on machine learning and Original label represents above manual-annotated label. The number of negative samples which are misclassified as positive samples in U-SVM is 75, and that of standard 3-class SVM is 119. Similarly, 33 positive samples are misclassified as negative samples in U-SVM, which is obviously less than 102 in standard 3-class SVM. Thus, compared with standard 3-class SVM, most of misclassified data are classified as neutral samples in U-SVM. Since misclassified to its opposite is more harmful than misclassified to a neutral view, U-SVM performs better than standard 3-class SVM. Besides, 294 neutral samples are misclassified (106 misclassified as negative, and 188 as positive) in U-SVM, and 405 neutral samples are misclassified (178 misclassified as negative, and 227 as positive) in standard 3-class SVM. It suggests less neutral samples are misclassified in U-SVM, which will also cause less damaging influences according to formula (13) in sentiment index composing compared with standard SVM. We also calculate mean absolute error in formula (12) for both U-SVM and standard 3-class SVM, and find a significant lower MAE of U-SVM than that of SVM. It is mainly because the data-dependent structure with universum samples contains priori information and improves learning performance, so that positive and negative samples can be separated more clearly, and universum samples can be intensive distributed near the hyperplane, which reduce misclassified neutral samples in U-SVM. Besides, an appropriate value of parameter a for empirical separating hyperplane to identify neutral samples is important. All of these results suggest a better performance of U-SVM in 3-class classification.

6 Conclusions

This paper studies the performance of universum SVM for investor sentiment identification. Our empirical studies suggest that for 2-class classification problem, SVM with

universum has a better performance than standard SVM in several text representation methods of the dataset. Results for different training subset sizes suggest that universum have especially good performance for small training dataset. Besides, effectiveness of universum may not be sustainable growth with the increasing number of universum samples. When size of universum samples arrives at a certain level, almost no more accuracy increasing can be provided by additional universum. Since for financial studies of investor sentiment, a large amount of neutral views need to be classified, we further discuss the situation of a 3-class problem to identify neutral views in out-of-sample prediction. We propose that $g(x) = 0.2$ are the empirical separating hyperplanes of U-SVM for the 3-class classification task in this paper, and compare its effectiveness with a standard 3-class SVM. Accuracy of U-SVM is not obviously better than that of standard SVM, but the results of U-SVM are more acceptable, because compared with standard 3-class SVM, most of misclassified data are classified as neutral samples, and less neutral samples are misclassified in U-SVM. We also calculate mean absolute error for both U-SVM and standard SVM, and find a significant lower MAE for U-SVM.

Overall, this paper shows better performance of U-SVM for both 2- and 3-class classifications and tries to make multiple appropriate explanations. However, due to the constraints of authors' resources and capacity, the original dataset from stock forum for empirical study is still quite single, even if we use several different text representation methods. In our further work, the original dataset is expected to be more abundant, and we can further discuss the empirical separating hyperplanes of U-SVM for 3-class classification in different datasets. Besides, the method for 3-class classification with U-SVM in this paper is empirical, and we may improve its theoretical part in future.

Acknowledgments This work has been partially supported by grants from National Natural Science Foundation of China (Nos. 61472390, 71101146, 11271361, 71331005, and 11226089), Major International (Regional) Joint Research Project (No. 71110107026) and the Beijing Natural Science Foundation (No. 1162005).

References

1. Long JBD, Waldmann RJ (1990) Noise trader risk in financial markets. J Bradford De Longs working papers 98(4):703–738
2. Lee CMC, Shleifer A, Thaler RH (1991) Investor sentiment and the closed-end fund puzzle. J Financ 46(1):75–109
3. Nagel S (2005) Short sales, institutional investors and the cross-section of stock returns. J Financ Econ 78(2):277–309
4. Barberis N, Xiong W (2010) Realization utility. J Financ Econ 104(2):251–271
5. Otoo MW (1999) Consumer sentiment and the stock market. Working Paper, Board of Governors of the Federal Reserve System, Washington, DC, pp 1–16
6. Charoenrook A (2006) Does sentiment matter? Working Paper, Ahlbrandt University
7. Lemmon M, Portniaguina E (2006) Consumer confidence and asset prices: some empirical evidence. Rev Financ Stud 19(4):1499–1529
8. Schmeling M (2009) Investor sentiment and stock returns: some international evidence. J Empir Financ 16(3):394–408
9. Wheatley SM, Neal R (1998) Do measures of investor sentiment predict returns? J Financ Quant Anal 33:523–547
10. Baker M, Wurgler J (2006) Investor sentiment and the cross-section of stock returns. Soc Sci Electron Publ 61(4):1645–1680
11. Baker M, Wurgler J (2007) Investor sentiment in the stock market. Soc Sci Electron Publ 21(2):129–151
12. Baker M, Wurgler J, Yuan Y (2012) Global, local, and contagious investor sentiment. J Financ Econ 104(2):272–287
13. Stambaugh RF, Yu J, Yuan Y (2012) The short of it: investor sentiment and anomalies. J Financ Econ 104(2):288–302
14. Stambaugh RF, Yu J, Yuan Y (2015) Arbitrage asymmetry and the idiosyncratic volatility puzzle. J Financ 70(5):1903–1948
15. Berger D, Turtle HJ (2015) Sentiment bubbles. J Financ Mark 23:59–74
16. Werner Antweiler, Frank Murray Z (2004) Is all that talk just noise? The information content of internet stock message boards. J Financ 59(3):1259–1294
17. Das SR, Chen MY (2007) Yahoo! for Amazon: sentiment extraction from small talk on the web. Manage Sci 53:1375–1388
18. Kim SH, Kim D (2014) Investor sentiment from internet message postings and the predictability of stock returns. J Econ Behav Organ 107(PB):708–729
19. Wu DD, Zheng L, Olson DL (2014) A decision support approach for online stock forum sentiment analysis. IEEE Trans Syst Man Cybern Syst 44(8):1077–1087
20. Vapnik VN (1998) Statistical learning theory. Wiley, New York
21. Vapnik V (2006) Estimation of dependences based on empirical data, 2nd edn. Springer, Berlin
22. Weston J, Collobert R, Sinz F, Bottou L, Vapnik V (2006) Inference with the universum. In: International conference, vol 2006, pp 1009–1016
23. Sinz FH, Chapelle O, Agarwal A, Schölkopf B (2007) An analysis of inference with the universum. Adv Neural Inf Process Syst 20(2008):1369–1376
24. Cherkassky V, Dai W (2009) Empirical study of the universum SVM learning for high-dimensional data. In: International conference on artificial neural networks—ICANN 2009, vol 5768, pp 932–941
25. Cherkassky V, Dhar S, Dai W (2011) Practical conditions for effectiveness of the universum learning. IEEE Trans Neural Netw 22(8):1241–1255
26. Dhar S, Cherkassky V (2015) Development and evaluation of cost-sensitive universum-SVM. IEEE Trans Cybern 45(4):806–818
27. Zhang D, Wang J, Wang F, Zhang C (2008) Semi-supervised classification with universum. In: Siam international conference on data mining, SDM 2008, April 24–26, 2008, Atlanta, Georgia, USA, vol 2, pp 340–344
28. Chen S, Zhang C (2009) Selecting informative universum sample for semi-supervised learning. In: International joint conference on artificial intelligence, vol 18, pp 111–122
29. Shen C, Wang P, Shen F, Wang H (2011) Uboost: boosting with the universum. IEEE Trans Pattern Anal Mach Intell 34(4):825–832
30. Qi Z, Tian Y, Yong S (2012) Twin support vector machine with universum data. Neural Netw 36C(3):112–119
31. Qi Z, Tian Y, Shi Y (2014) A nonparallel support vector machine for a classification problem with universum learning. J Comput Appl Math 263(263):288–298

32. Lu S, Tong L (2015) Weighted twin support vector machine with universum. Adv Comput Sci Int J 3(2):17–23
33. Xu Y, Chen M, Li G (2015) Least squares twin support vector machine with universum data for classification. Int J Syst Sci 47(15):3637–3645
34. Liu CL, Hsaio WH, Lee CH, Chang TH (2015) Semi-supervised text classification with universum learning. IEEE Trans Cybern 46(2):1
35. Xu Y, Chen M, Yang Z, Li G (2016) ν-twin support vector machine with universum data for classification. Appl Intell 44(4):956–968
36. Pan S, Wu J, Zhu X, Long G, Zhang C (2016) Boosting for graph classification with universum. Knowl Inf Syst 1–25. doi:10.1007/s10115-016-0934-z
37. Zhu C (2016) Double-fold localized multiple matrix learning machine with universum. Form Pattern Anal Appl 1–28. doi:10.1007/s10044-016-0548-9
38. Gao T, Tian Y, Shao X, Deng N (2008) Accurate prediction of translation initiation sites by universum SVM. J Chem Eng Jpn 42(8):570–575
39. Chen S, Zhang C (2009) Image classification via SVM using in-between universum samples. In: 16th IEEE international conference on image processing (ICIP), pp 1421–1424
40. Jiao Y, Zhang X, Zhuo L, Chen M (2010) Tongue image classification based on Universum SVM. In: IEEE international conference on biomedical engineering and informatics, vol 2, pp 657–660
41. Hao X, Zhang D (2013) Ensemble universum SVM learning for multimodal classification of Alzheimer's disease. Mach Learn Med Imaging 8184(2013):227–234
42. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
43. Vapnik VN (1996) The nature of statistical learning theory. Springer, New York
44. Trafalis TB, Ince H (2000) Support vector machine for regression and applications to financial forecasting. IEEE-Inns-Enns international joint conference on neural networks, vol 6, pp 6348–6348
45. Schölkopf B, Tsuda K, Vert J (2004) Support vector machine applications in computational biology. Kernel methods in computational biology. MIT Press, Cambridge
46. Goh KS, Chang EY, Li B (2005) Using one-class and two-class svms for multiclass image annotation. IEEE Trans Knowl Data Eng 17(10):1333–1346
47. Isa D, Lee LH, Kallimani VP, Rajkumar R (2008) Text document preprocessing with the Bayes formula for classification using the support vector machine. IEEE Trans Knowl Data Eng 20(9):1264–1272
48. Borgwardt KM (2011) Kernel methods in bioinformatics. Handbook of statistical bioinformatics. Springer, Berlin
49. Deng N, Tian Y, Zhang C (2012) Support vector machines. Optimization based theory, algorithms, and extensions. CRC Press, New York
50. Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18(10):613–620
51. Harris ZS (1954) Distributional structure. Synthese Language Library 10:146–162
52. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manage 24(88):513–523