Jurnal Teknologi

Full Paper

SENTIMENT CLASSIFICATION WITH CONCEPT DRIFT AND IMBALANCED CLASS DISTRIBUTIONS

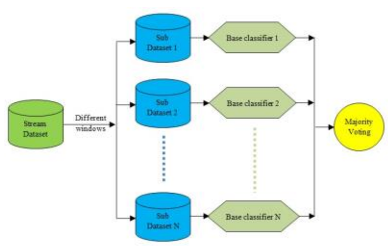
Abbas Jalilvand[a,b*], Naomie Salim[a]

[a]Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia
[b]Department of Computer Engineering, Hashtgerd Branch, Islamic Azad University, Alborz, Iran

Article history
Received
5 September 2016
Received in revised form
14 November 2016
Accepted
8 November 2016

*Corresponding author
jabbas2@live.utm.my

Graphical abstract



Abstract

Document-level sentiment classification aims to automate the task of classifying a textual review, which is given on a single topic, as expressing a positive or negative sentiment. In general, people express their opinions towards an entity based on their characteristics which may change over time. User's opinions are changed due to evolution of target entities over time. However, the existing sentiment classification approaches did not considered the evolution of User's opinions. They assumed that instances are independent, identically distributed and generated from a stationary distribution, while generated from a stream distribution. They used the static classification model that builds a classifier using a training set without considering the time that reviews are posted. However, time may be very useful as an important feature for classification task. In this paper, a stream sentiment classification framework is proposed to deal with concept drift and imbalanced data distribution using ensemble learning and instance selection methods. The experimental results show the effectiveness of the proposed method in compared with static sentiment classification.

Keywords: Sentiment classification, Concept drift, Imbalanced data, Ensemble learning, Instance selection

1.0 INTRODUCTION

Nowadays, the web is the most important place for expressing sentiments, evaluations, and reviews. Lots of people are tending to give their opinions in forums, blogs, discussion boards and social networks. However, with the rapid growth of e-commerce activity, the number of reviews and opinions has increased exponentially and this source of information is becoming unworkable. Nevertheless, the high volume of reviews makes it difficult for individuals and organizations to read and understand all of them. To solve this problem, a hot research area has recently emerged, which is called opinion mining and sentiment analysis. Sentiment classification is the most active field in opinion mining that aims to determine whether an opinionated text expresses a positive, negative or neutral opinion. Sentiment classification is applied at word-level, sentence-level, document-level and feature/aspect-level using different methods ranging from unsupervised to supervised approaches [1-3].

Supervised sentiment classification is aim to automatically classify an opinion text into the positive ('thumbs up') or negative ('thumbs down') class by employing some machine learning techniques (e.g. Support Vector Machine (SVM), Naïve Bayes (NB), and K Nearest Neighbors (KNN))[3]. They usually employs a static supervised learning strategy, in which a classification model is first built using a training set to classify a testing set without considering the time that reviews are posted.

However, time may be very useful as an important feature for classification task. In general, people express their sentiments about a target entity (e.g. product or a service etc.) based on their characteristics which are changed over time. User's opinions are changed due to evolution of target entities over time. For example, in the phone product, some features changed (add or remove) at the specific time and some terms (words) associated to the features may be appeared or disappeared in the phone reviews. However, the existing sentiment classification approaches not considered the evolution of review document. They assumed that instances are independent, identically distributed and generated from a stationary distribution (Figure 1), while generated from a stream distribution.

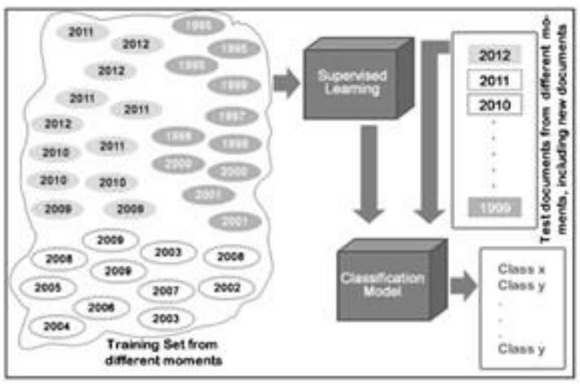


Figure 1 Static learning model

For example, in the phone product, before 2010, a phone with (2MB RAM, 1MP video and 200$ price) was widely accepted as a good one; While, after 2010, a phone with (2MB RAM, 1MP video and 200$ price) is not considered to be of high configuration. The phenomenon of concept changing over time is termed as concept drift in machine learning. In contrast to static concept learning, ordering of the training data is important in concept drift learning. In fact, each target function inferred at time t can only utilize the data given before t (Figure 2).

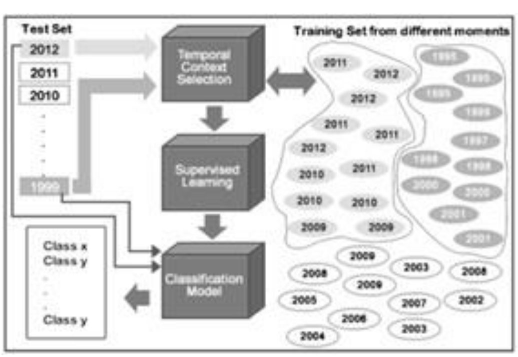


Figure 2 Stream learning model

In concept drifts learning, classifier needs to be updated to track the changing in data. It is not questionable that the ability to automatically adapt the classifier over time plays an important role in the real-world application of sentiment classification. Besides the concept drifts problem, imbalanced data is another problem needed to be addressed. In classification task, distribution classes of dataset may be unequal that is called imbalanced data problem, that learning algorithms are biased towards the majority classes [5]. One important issue, not yet convincingly addressed, is the handling of concept drifts and imbalanced data problems in the sentiment classification domain. Thus, a general framework for dealing with both skewed class distribution and concept drifts is in great demand. In summary, the main contribution of this paper is a new methodology for stream sentiment classification, which can track changing user's opinions. A series of experiments was conducted to evaluate the performance of three different classifiers (SVM, NB and KNN) to classify the sentiment of stream reviews. The remainder of the paper is organized as follows: Sub section 1.1 provides a review of related work on sentiment classification, concept drift and imbalanced data handling methods. Section 2 provides the research design. Several experiments are presented in Section 3. Finally, the conclusions are discussed in Section 4.

1.1 Literature Review

In this paper, the sentiment classification is considered to be stream classification. Thus, two problems concept drifts and imbalanced data based on instance selection method are addressed. Concept drifts, imbalanced data and instance selection are three hot fields that researchers have done a great deal of research to address them.

1.1.1 Sentiment Classification

Sentiment analysis aims to analyze opinions that are presented by people [3]. Supervised sentiment classification attempts to determine whether a text is positive or negative using machine learning approach. Sentiment classification has several important tasks, including preprocessing, data reduction, classification, and etc. Many approaches ranging from unsupervised, semi-supervised and supervised are used for sentiment classification. Traditional topical text classification approaches were applied by many researchers for the supervised sentiment classification. They considered an opinionated document as a bag of words (BOW) and used machine learning techniques. Pioneering work on document-level sentiment classification compared NB, Maximum Entropy (ME), and SVM to classify movie reviews into two classes: positive and negative and achieved the highest classification accuracy (82.9 percent) using SVM [1]. After the success of the supervised approach in sentiment classification, researchers have tried to improve their performance. Thus, some work has been done on feature selection methods to decrease the feature dimension [2, 6] in sentiment classification. Feature selection is an important part of classification task by reducing the irrelevant and redundant features to improve the accuracy and training speed of classifiers. Previous work, however, mostly focuses on stationary classification model while ignoring the stream of opinion data. Thus, in this paper, stream classification model is adopted for sentiment classification.

1.1.2 Concept Drift

Generally, there are two main approaches for coping with the concept drift in the data streams, most of them stem from the same approach in that the algorithm's ability to adapt to concept drift is achieved by learning from a single window of most recent examples. They can be divided into two main groups: trigger based and evolving. Trigger-based methods work based on a change detector and an on-line classifier. The classifier is updated if the change is detected [7]. Evolving methods don't use any direction to detect changing to update the classifier. Adaptive ensembles are one those methods. In this paper we are particularly interested in block-based ensembles, where component classifiers are constructed from sequential-coming blocks (also called data chunks) of training data. When a new block is available, a new classifier is built from it and existing classifiers are evaluated. The worst classifier is replaced with the new classifier in the ensemble. The dynamic weighted majority (DWM) algorithm [8] is presented which uses an online learner such as NB, or incremental tree inducer to train an ensemble with the final voting decision obtained by dynamic weighted majority voting. The voting weight of each classifier is set to 1 when created, and is reduced when that classifier misclassifies an instance. Once the classifier's weight falls below a threshold, it is removed from the ensemble. The Learn++.NSE algorithm, on the other hand, uses a weighted sum of the current and past normalized pseudo errors of each classifier to compute the voting weight [9].

1.1.3 Imbalanced Data

Handling class imbalance has become an important research problem in recent years because more people have realized that imbalance in class distribution causes suboptimal classification performance [10]. Proposed solutions to this problem include preprocessing data, transforming algorithms, or post-processing models. Among the solutions, balancing training set distribution is the most popular approach, specifically, many sampling algorithms either under-sample majority examples or over-sample minority examples. Instance selection is the most popular approach for under-sampling. Instance selection methods can be divided in two groups: wrapper methods and filter method. Wrapper methods search the space of Instance subsets to find an optimal subset based on the accuracy obtained by a classifier. Filter methods remove irrelevant and redundant instances based on a selection function.

Most of the wrapper methods use the KNN algorithm. One of the earliest methods is the Condensed Nearest Neighbor (CNN) [11] that starts with S including one instance belonging to each class. Then, each instance in T is classified using S as training set, instances is misclassified is added to S and this step is repeated until all instances be classified correctly using S. Since noisy instances are commonly misclassified by their neighbors, this method retains them. Some extensions of CNN are proposed to enhance its performance: Selective Nearest Neighbor rule (SNN) [12]. Edited Nearest Neighbor (ENN) [13] is another instance selection method which aims to remove noisy instances in a training set. Instances which the majority class of its k nearest neighbors is different are discarded (ENN uses k=3). The Reduced Nearest Neighbor rule (RNN) [14] is another algorithm that starts with S = T and removes each instance from S if such a removal does not cause any other instances in T to be misclassified by the instances remaining in S. There are also various methods based on active learning that deal with the selection of relevant instances [15].

2.0 METHODOLOGY

Most existing concept drift learning algorithms, work based on learning from a window of most recent examples to adapt the concept drift [7]. Figure 3 illustrates this approach. Obviously, the approach automatically excludes older examples that are no longer relevant. Determining the appropriate window size play an important role that is not easy to do.

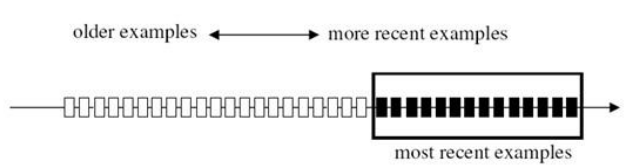


Figure 3 A typical approach to concept drift learning

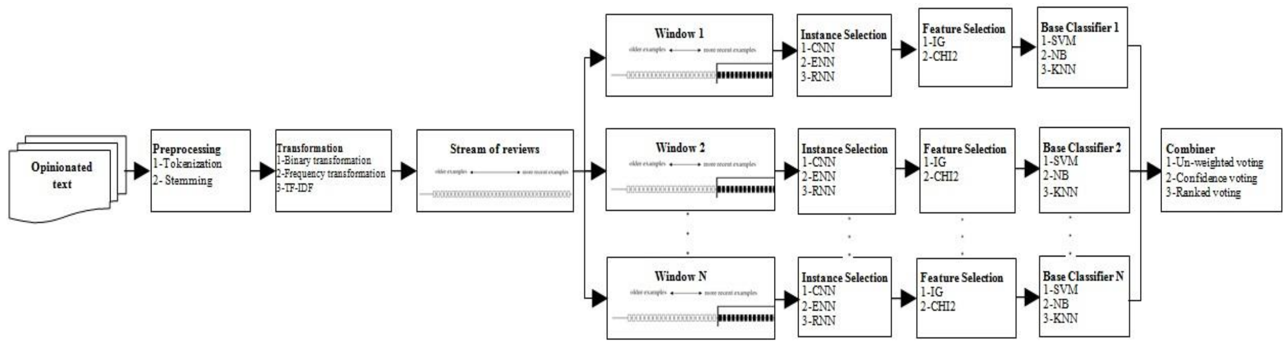


Figure 4 The stream sentiment classification framework

Some researchers have developed an adaptive window adjustment heuristically to solve the problem and it is considered effective in slow drift rate condition [4]. Among these methods, the most popular evolving technique for handling concept drift is classifiers ensemble [8, 9]. Besides the challenge to adapt learning model, imbalanced data is a critical problem that needs to be concerned. Many approaches have been proposed to deal with to the imbalanced data problem both at the data and algorithmic levels. Most approaches for learning from such data are based on under-sampling the majority class or over-sampling the minority class [5]. The training dataset is balanced in Oversampling by instance generation and in under-sampling methods by instance selection. These methods might improve the prediction accuracy of minority classes, but they are not sufficient for stream data with concept drifts due to lack of adaptability.

Some researcher have used ensemble learning to deal with imbalanced data distribution. They used over/under-sampling methods to generate diversity for ensemble classification that lead to be highly accurate [4]. Previous studies assume sentiment datasets are balanced, while in the real world they are imbalanced, especially when considering in stream learning form. Therefore, ensemble learning is effective for deal with both imbalanced data distribution and concept drifts problem [5] that is used to propose a framework for sentiment classification in this study. This framework addresses two research challenges in stream sentiment classification: Concept Drift and Imbalanced data. In ensemble learning, model construction plays a vital role. There are four fundamental approaches to build diverse base classifiers: 1) using different combination schemes, 2) using different classifier models, 3) using different feature subsets, and 4) using different training sets that is the most popular [16]. In this study, an integrated model based on different training sets and instance selection method is proposed. The proposed framework is shown in Figure 4. We have proposed a simple strategy that can effectively classify imbalanced stream review. In stream data mining, the incoming stream data arrives in sequential chunks, $C_1, C_2, \dots, C_t$ where C_t is the most up-to-date chunk. The next chunk C_{t+1} is considered to be the testing set that aims to predict using a classifier that is trained based on previous chunks as training set.

$$\underbrace{d(1,1),\dots,d(1,m)}_{chunk_1},\underbrace{d(2,1),\dots,d(2,m)}_{chunk_2},\dots,\underbrace{d(t,1),\dots,d(t,m)}_{chunk_t},\underbrace{d(t+1,1),\dots,d(t+1,m)}_{chunk_{t+1}}$$

Different windows of most recent training dataset are selected to build base classifiers. Since, the selected training set for each base classifier is imbalanced; instance selection approach is used to balance the class distribution and a new balanced training set is formed. To improve the performance of classification task, feature selection is considered as next step. Information gain and Chi-square are the most popular filtering techniques [17] that can be applied to select appropriate features. Then *N* classifiers are trained based on *N* balanced training sets. Finally the outputs of base classifiers are combined to predict the testing set.

3.0 RESULTS AND DISCUSSION

In this section, we empirically evaluated the performance of concept drift on sentiment classification domain. Based on our research design, two different experiments have been conducted. In experiment 1, we compared the stream sentiment classification against previously used static sentiment classification method to show the effectiveness of stream classification in this domain. In Experiment 2, we evaluated the performance of proposed framework for stream sentiment classification. Experiments conducted using RNN algorithm for instance selection, IG algorithm for feature selection and three algorithms (SVM, NB and KNN) for classification tasks. We used the SVM classifier based on default parameter values with a usual nonlinear kernel from LIBSVM software package [18]. OpenPR-NB [19] is used as the Naïve Bayes classifier in our experiments. We set K=3 for KNN classifier using Euclidean measure. In static learning using finite training sets, cross-validation and variants (leave-one-out, bootstrap) are the standard methods to evaluate learning systems. Cross-validation is appropriate for datasets, generated by stationary distributions, and assuming that instances are independent. In data streams, the distribution generating examples and the decision models evolve over time, cross-validation are not applicable.

3.1 Data Source

Four publicly available datasets were used in this research. The multi-domain sentiment (MDS) used by Blitzer *et al*. crawled from Amazon.com containing four different types of product reviews (Book, DVD, Electronics and Kitchen) [20]. This dataset contains 1000 positive and 1000 negative examples for each domain. Pre-processing was performed on both of the datasets. Punctuation, non-alphabet characters and some unsuitable stop words were removed. We adopted term present/absent model (unigram) to represent features and extracted all words occurring at least three times [1]. Summary statistics of the datasets before and after preprocessing is shown in Table 1. The review number distribution of Book datasets (one of the datasets as an example) based on seasonally period is shown in Figure 5.

Table 1 Dataset in the number of words

Dataset	Book	DVD	Electronic	Kitchen
Corpus size(before pre-processing)	354203	343317	223609	188402
Corpus size(after pre-processing)	176853	173257	98796	83027
The number of features	8457	4657	4216	3738

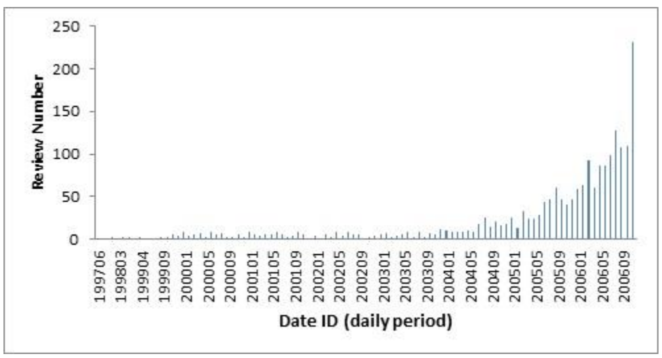


Figure 5 Review Number Distribution of Book Dataset

3.2 Experiment 1

In this section, some experiments are conducted to show the effectiveness of stream sentiment classification. Thus, stream sentiment classification is compared to static sentiment classification method. To do this, several chunks from different dataset are selected as testing set, and are classified using two stream and static models. The results can show the effectiveness of the proposed stream sentiment

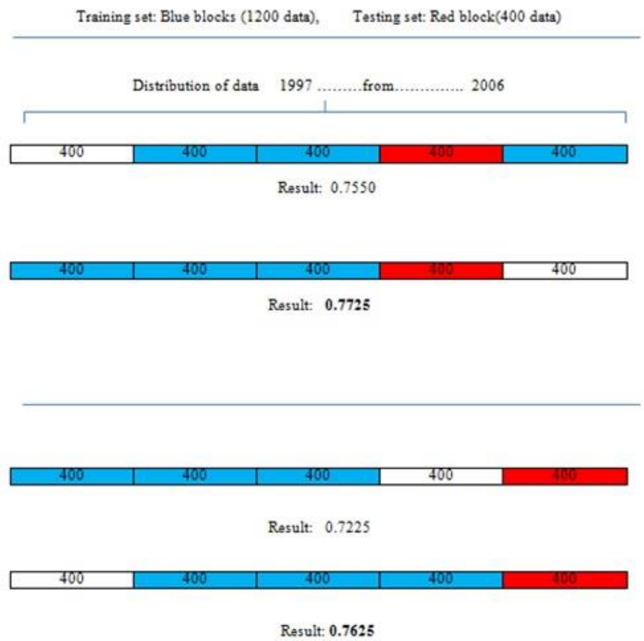


Figure 6 Stream against static sentiment classification accuracy

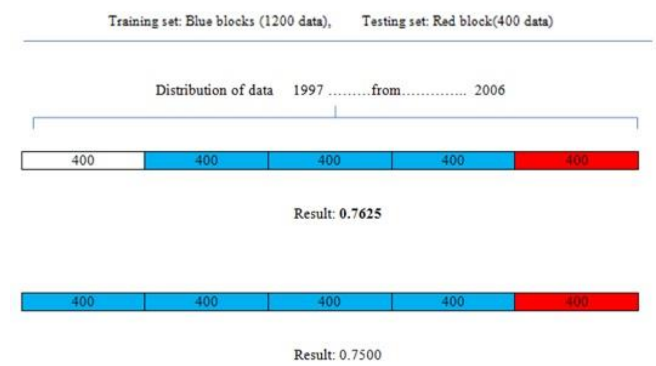


Figure 7 Stream sentiment classification accuracy with different windows

classification. In order to assess the impact of stream sentiment classification, the reviews in datasets are sorted based on posted time. One chunk is randomly selected as testing set and others chunks are used as training set. To build a static classifier, some chunks in any order is considered as training set, while, chunks that are before of the testing chunk are used to construct the stream classifier. Also, to assess the impact of appropriate window to handle concept drift, different windows in same condition are evaluated.

As it can be seen from Figure 6, in the first case, the accuracy of testing set (red chunk, contains 400 instances) is 0.7550 when the data used as training set are without any order (the blue chunks). While, the classification accuracy of this testing set is increased to 0.7726 when the training set is used is placed before it. In the second case, the importance of the ordering of training set is shown as well. Therefore, the sentiment classification in stream model is more effective. Figure 7 shows that the classification accuracy of testing set using (w= 1200 data) has considerably better outcome than (w=1600 data). Therefore, the window of most recent examples is highly important to adapt the concept drift.

3.3 Experiment 2

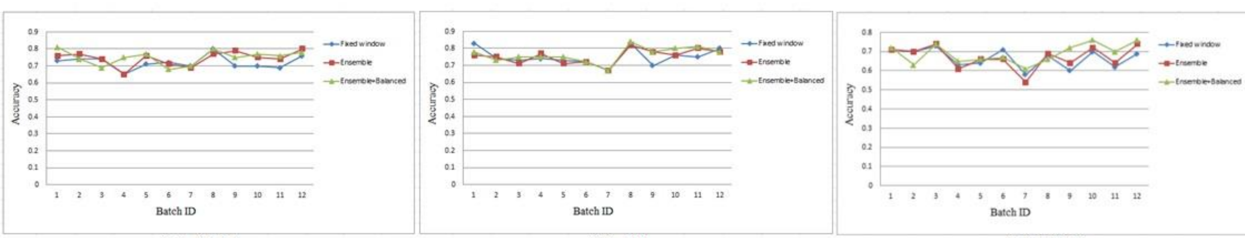
To the best of our knowledge, this study is the first attempt to use stream classification for sentiment analysis. Since, we have shown the effectiveness of the stream classification versus static classification in experiment 1. The stream sentiment classification performance with concept drift learning and imbalanced class distribution is reported in experiment 2 on four datasets. Each dataset has 2000 reviews that are divided into 20 chunks (each chunk contains 100 instances). We investigate three different window management approaches:

- Fixed Window: The classifier is built on the instances from a fixed size of window. Here, we assign the window size to 8 previous chunks that are considered to be the training dataset.
- Ensemble: Ensemble the base classifiers are built based on different sizes of window.
- Ensemble + Balanced: Ensemble the base classifiers are built based on different sizes of window and are balanced using instance selection method.

As we set the window size to 8 chunks, the ninth chunk is considered to be the first testing set and next 11 chunks are evaluated as other testing sets incoming over time. Figures 8-11 show the averaged accuracy of the 12 chunks for the four datasets using three different classification algorithms (SVM, NB and KNN). In each step, the incoming chunk is tested using three methods and the averaged accuracy is plotted. Also, Tables (2-5) summarize the experimental results (precision, recall, F1 and accuracy) for all chunks in average.

Table 2 Book

Method	SVM				NB				KNN			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
Fixed window	0.6611	0.7381	0.6893	0.7200	**0.7116**	0.7596	0.7291	0.7500	**0.5295**	0.7002	0.6000	0.6658
Ensemble	0.6888	**0.7652**	0.7162	0.7442	0.6984	0.7635	0.7255	0.7525	0.5505	0.7023	0.6149	0.6708
Ensemble + Balanced	**0.7475**	0.7370	**0.7390**	**0.7500**	0.7111	**0.7751**	**0.7385**	**0.7633**	0.5526	**0.7288**	**0.6266**	**0.6900**

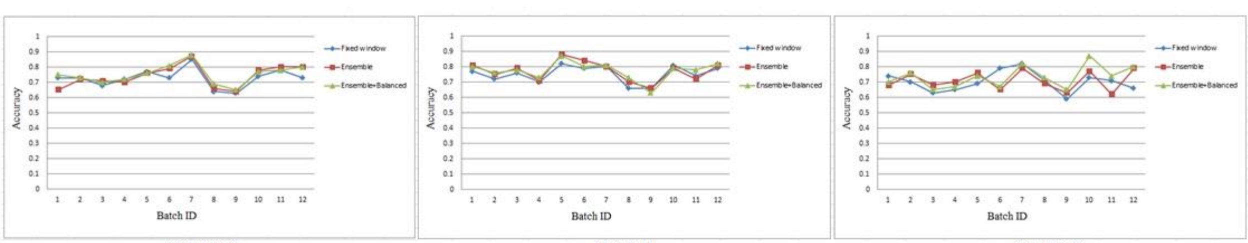


(a): SVM (b): NB (c): KNN

Figure 8 Stream sentiment classification accuracy for Book dataset

Table 3 Dvd

Method	SVM				NB				KNN			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
Fixed window	0.7410	0.7449	0.7287	0.7275	**0.8173**	0.7310	0.7674	0.7517	0.6873	0.7207	0.6981	0.7017
Ensemble	0.7231	0.7608	0.7316	0.7400	0.7999	0.7655	0.7792	0.7717	0.6711	0.7477	0.6988	0.7092
Ensemble + Balanced	0.7475	0.7641	0.7533	0.7533	0.8069	0.7689	0.7836	0.7750	0.7116	0.7543	0.7279	0.7333



(a): SVM (b): NB (c): KNN

Figure 9 Stream sentiment classification accuracy for Dvd dataset

Table 4 Electronic

Method	SVM				NB				KNN			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
Fixed window	0.7508	0.7865	0.7549	0.7625	0.8005	0.7789	0.7869	0.7900	0.7618	0.7451	0.7464	0.7467
Ensemble	0.7369	**0.8168**	0.7640	0.7800	**0.8143**	0.7906	0.8001	0.7992	0.7946	0.7661	0.7762	0.7725
Ensemble + Balanced	**0.8112**	0.7783	**0.7930**	**0.7950**	0.8120	**0.8028**	**0.8053**	**0.8067**	**0.8005**	**0.7724**	**0.7822**	**0.7792**

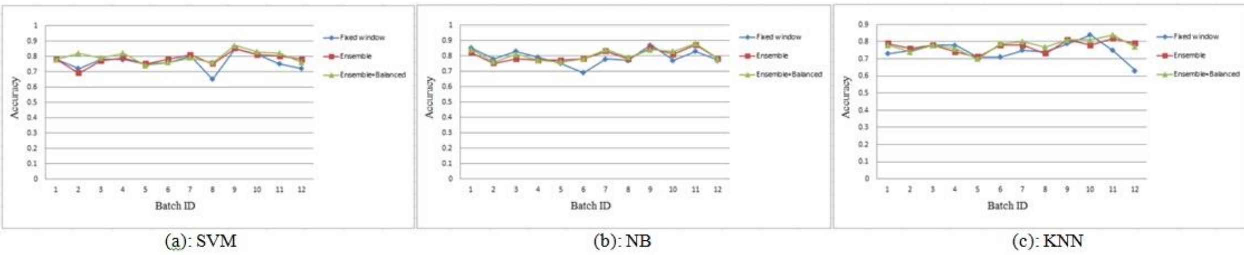


Figure 10 Stream sentiment classification accuracy for Electronic dataset

Table 5 Kitchen

Method	SVM				NB				KNN			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
Fixed window	0.7041	0.8084	0.7414	0.7583	0.8047	0.7941	0.7955	0.7983	**0.7793**	0.7412	0.7516	0.7450
Ensemble	0.6937	**0.8194**	0.7432	0.7633	**0.8240**	0.8142	0.8170	0.8192	0.7584	0.7628	0.7556	0.7575
Ensemble + Balanced	**0.7316**	0.8180	**0.7669**	**0.7800**	0.8223	**0.8174**	**0.8179**	**0.8208**	0.7616	**0.7733**	**0.7633**	**0.7675**

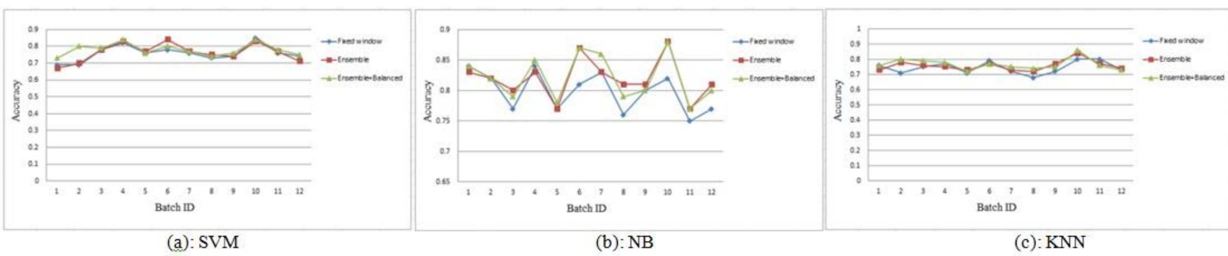


Figure 11 Stream sentiment classification accuracy for Kitchen dataset

The highest average results of different methods are boldfaced. As can be seen from the experimental results the Ensemble + Balanced method has better performance using NB, while the lowest performance was observed with the fixed window method using KNN. The highest average accuracy of the Book dataset is 76.33%. The highest average accuracy of the Dvd dataset is 77.5%. The highest average accuracy of the Electronic dataset is 80.67%. The highest average accuracy of the Kitchen dataset is 82.08%. It is interesting that the NB classifier has better performance in all datasets. We could see from the results (Table 6) that NB is a more appropriate learner for this domain than SVM (accuracy with it is about 2.2% better) and KNN (accuracy with it is about 5% better).

Table 6 All dataset on average

Method	SVM		NB		KNN	
	F1	Accuracy	F1	Accuracy	F1	Accuracy
Fixed window	0.7286	0.7421	0.7697	0.7725	0.6990	0.7148
Ensemble	0.7388	0.7569	0.7805	0.7857	0.7114	0.7275
Ensemble + Balanced	0.7631	0.7696	**0.7863**	**0.7915**	0.7250	0.7425

4.0 CONCLUSION

In this paper, a framework for stream sentiment classification is proposed. In this framework, two problems, concept drifting and imbalanced data distribution, are addressed using ensemble learning and instance selection methods. Our research has demonstrated that sentiment classification is a stream data mining problem, and the proposed framework can adapt the classifiers and is effective for improving the classification accuracy on sentiment datasets. Empirical results showed that proposed method has better performance in compare with two other window size management algorithms. Also, NB is a more appropriate learner than SVM and KNN learners for this stream sentiment classification. For the future work, we plan to apply different methods to determine the appropriate window to deal with concept drift in sentiment classification as well as applying different instance selection methods to handle the imbalanced data distribution.

Acknowledgement

This work is supported by the Research Management Centre (RMC) at the Universiti Teknologi Malaysia (UTM) under Exploratory Research Grant Scheme (Vote No. 13H46) and Hashtgerd Branch Islamic Azad University (HIAU).

References

[1] Pang, B., Lee, L. and Vaithyanathan, S. 2002. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. *Proceedings Of The ACL-02 Conference On Empirical Methods In Natural Language Processing.* Association for Computational Linguistics. 10: 79-86.

[2] Abbasi, A. 2010. Intelligent Feature Selection For Opinion Classification. *IEEE Intelligent Systems*. 25: 75-79.

[3] Wang, G., Sun, J., Ma, J., Xu, K., & Gu, J. 2014. Sentiment Classification: The Contribution Of Ensemble Learning. *Decision Support Systems*. 57: 77-93.

[4] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. 2014. A Survey On Concept Drift Adaptation. *ACM Computing Surveys (CSUR)*. 46(4): 44.

[5] Gao, J., Ding, B., Fan, W., Han, J., & Yu, P. S. 2008. Classifying Data Streams With Skewed Class Distributions And Concept Drifts. *IEEE Internet Computing.* 12(6): 37-49.

[6] Jalilvand, A. and Salim, N. 2016. Feature Unionization: A Novel Approach For Dimension Reduction. *Applied Soft Computing.* doi:10.1016/j.asoc.2016.08.031.

[7] Gama, J. 2010. *Knowledge Discovery from Data Streams*. CRC Publishers.

[8] J. Z. Kolter and M. A. Maloof, 2007. Dynamic Weighted Majority: An Ensemble Method For Drifting Concepts. *Journal of Machine Learning Research*. 8: 2755-2790.

[9] M. D. Muhlbaier and R. Polikar, 2007. Multiple Classifiers Based Incremental Learning Algorithm for Learning in Nonstationary Environments. *IEEE International Conference on Machine Learning and Cybernetics (ICMLC 2007)*. 6: 3618-3623.

[10] N. V. Chawla, N. Japkowicz, and A. Kotcz, 2004. Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explorations*. 6(1): 1-6.

[11] Hart, P. E. 1968. The Condensed Nearest Neighbor Rule. *IEEE Trans Inf Theory.* 14: 515-516.

[12] Ritter, G. L., Woodruff, H. B., Lowry, S. R., Isenhour, T. L. 1975. An Algorithm For A Selective Nearest Neighbor Decision Rule. *IEEE Trans Inf Theory.* 21(6): 665-669.

[13] Wilson, D. L. 1972. Asymptotic Properties Of Nearest Neighbor Rules Using Edited Data. *IEEE Trans Syst Man Cybern*. 2: 408-421.

[14] G. W. Gates. 1972. The Reduced Nearest Neighbor Rule. *IEEE Trans. Information Theory*. 18(3): 431-433.

[15] Stojanović, M. B., Božić, M. M., Stanković, M. M., & Stajić, Z. P. 2014. A Methodology For Training Set Instance Selection Using Mutual Information In Time Series Prediction. *Neurocomputing*. 141: 236-245.

[16] Tsai, C. F., Hsu, Y. F. and Yen, D. C. 2014. A Comparative Study Of Classifier Ensembles For Bankruptcy Prediction. *Applied Soft Computing*. 24: 977-984.

[17] F. Sebastiani. 2002. Machine Learning In Automated Text Categorization. *ACM Computing Surveys (CSUR)*. 34: 1-47.

[18] Chang, C. C. and Lin, C. J. 2011. LIBSVM: A Library For Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*. 2(3): 27.

[19] Nan, J. 2016. Accessed from http://www.openpr.org.cn/.

[20] Blitzer, J., Dredze, M., & F. Pereira. 2007. Biographies, Bollywood, Boom-Boxes And Blenders: Domain Adaptation For Sentiment Classification. *Proceedings Of The 45th Annual Meeting Of The Association Of Computational Linguistics*. 440-447.