온라인 제품 리뷰 스팸 판별을 위한 점증적 SVM

지청장[1,2], 장진홍[3], 강대기[2,*]

[1]웨이팡과기대학, [2]동서대학교, [3]서우광 실험초등학교

Incremental SVM for Online Product Review Spam Detection

Ji Chengzhang[1,2], Zhang Jinhong[3] and Dae-Ki Kang[2,*]

[1]Weifang University of Science & Technology, [2]Dongseo University,

[3]Shou Guang Experimental Primary School

E-mail : dkkang@dongseo.ac.kr[*]

요 약

제품 리뷰들은 잠재적인 고객의 구매 선택에 매우 중요하다. 제품 리뷰들은 또한 제조사들로 하여금 자신들의 제품의 문제점을 찾고 경쟁자들의 비즈니스 정보를 수집하는 데 사용된다. 그러나 어떤 사람들은 가짜 리뷰를 쓰고, 잠재적인 고객들과 제조사들로 하여금 잘못된 선택을 하게 만든다. 따라서 가짜 리뷰 판별은 전자 상거래 사이트에서 주된 문제들 중 하나이다. 서포트 벡터 머신즈(SVM)는 좋은 성능을 보이는 중요한 텍스트 분류 알고리즘이다. 본 논문에서는 온라인 리뷰 스팸을 판별하기 위해 가중치, Karush - Kuhn-Tucker(KKT) 조건의 확장, 그리고 컨벡스 헐(Convex Hull)에 근거한 점증적 알고리즘을 제시한다. 최종적으로 우리는 제시된 알고리즘의 성능을 이론적으로 분석한다.

ABSTRACT

Reviews are very important for potential consumer' making choices. They are also used by manufacturers to find problems of their products and to collect competitors' business information. But someone write fake reviews to mislead readers to make wrong choices. Therefore detecting fake reviews is an important problem for the E-commerce sites. Support Vector Machines (SVMs) are very important text classification algorithms with excellent performance. In this paper, we propose a new incremental algorithm based on weight and the extension of Karush - Kuhn-Tucker(KKT) conditions and Convex Hull for online Review Spam Detection. Finally, we analyze its performance in theory.

키워드

Review spam, Incremental SVMs, Online detection, Karush - Kuhn-Tucker conditions, Convex hull.

Ⅰ. Introduction

Most of the E-commerce sites allow the consumers to post reviews of products. That is helpful for the customers to make a decision to buy the product. Furthermore, vendors also utility those opinions to survey corporate reputation and brand perception. But some people post fake reviews to mislead readers on purpose. It is necessary to detect the fake reviews and to remove them.

Many learning algorithms [1] are proposed in the papers to resolve this problem. These algorithms actually classify the reviews into real review and fake reviews. This is a typical classification problem. Most of those learning algorithms are batch ones, and all the training data is obtained and trained at a time. But in many practical applications, it is expensive or time consuming to obtain training data, so learning materials usually is supported incrementally and learning process can't be finished at a time. Consequently continuous information acquisition and exchange is necessary to provide a classification system with true world representation. Incremental learning is an available method to resolve this problem and some kinds of incremental algorithms are proposed and used in online system. Among them, incremental Support Vector Machine (SVM) algorithm is an emerging method with good performance. Traditional incremental SVMs retrain the classifier with newly arrived samples and support vectors of the current trained SVMs. Firstly, using only the support vectors of the current classifier as historical information in the updating process will possibly result in degrading the accuracy because some non-support vectors could include important information. In [13], authors increase the historical data to improve it. However, because historical data is not efficiently managed, above problem is not well resolved and it wastes a lot of memory to preserve the historical data. We define the weight that indicates example′ importance for resolving this problem. Secondly, there are a lot of examples having no effect on generating SVMs in the training data. SVMs require a long time and a lot of memory for training when training set is big. Some approaches are proposed to improve it by removing unuseful examples from the incremental examples [12].

To resolve above problems, we propose a new incremental SVM algorithm based on importance weight and extension of KKT conditions and Convex Hull. We call it WKH-SVMs. Finally, we make some analysis of the performance in theory. This paper arranged in following sections. Section 2 gives some related background. Section 3 gives the details of proposed work. Section 4 makes some analysis for WKH-SVMs. Section 5 gives us a conclusion of this paper and discusses the future work.

II. Related Work

2.1 Support Vector Machines

In a high- or infinite-dimensional space, Support Vector Machine constructs a hyperplane that represents the largest margin between the two classes and it is often used for classification and regression. The larger the margin the lower the generalization error of the classifier.

In standard learning, we use the following notation to describe SVMs [7, 13]. Training set X includes labeled vectors $\{(x_1,y_1),\ldots,(x_n,y_n)\}$, those item (x_i,y_i) is an example i with features and class label y_i of that example. Classification function f(x) is denoted by using a hypothesis vector w and bias term b:

$$f(x)=sign(\langle w,x\rangle+b)$$

Finding an optimal hyperplane is equivalent to solving the following QP problem:

$$\tau(w,\xi)=\frac{1}{2}\parallel w\parallel^2+C\sum_{i=1}^{n}\xi_i$$

subject to:

$$y_i(\langle w,x_i\rangle+b)\geq 1-\xi_i,\xi_i\geq 0$$

where ξ_i is a slack variable to tolerate mis-classifications, and indicates the amount of error that the classifier makes on a given example x_i. The penalty factor C shows how much importance to give loss function.

2.2 Incremental learning

Now there is no standard definition of incremental learning. We adopt the definition that is proposed in [5]. It needs to meet the following criteria:

1. It can acquire additional information from new data.
2. It should not need to get the original examples used to train the existing classifier.
3. It should retain previously acquired knowledge.
4. It should allow new data to bring in the new classes.

In some real-world problems, we often require the online system to have the ability to incrementally learn from data, and incremental learning often is used as online learning. It allows a online learning system to adapt itself in a changing environment.

2.3 Incremental Support Vector Machines

Re-training SVMs on the entire data of previously obtained data for new data will cost a lot of memory and computation time. In [3], Syed proposed an incremental SVM Learning Algorithm for it. Because the amount of support vectors usually is much smaller than the amount of training examples and support vectors include the best important information, it is considerably reasonable to use support vectors to represent the previous data and to use an old hypothesis as the starting point. Due to the KKT conditions, it is not necessary to retrain on well classified data [7]. But some examples that meet KKT conditions could include some important information for final SVM. Therefore, if we only consider the examples violating KKT conditions, that some useful information will be lost and accuracy could be degraded.

III. Proposed Algorithm

Support Vector Machines have been successfully used for learning with large and high dimensional data sets. Unfortunately, we have to spend large amount of time to train Support Vector Machines when the training data is big. Some incremental algorithms use support vectors as compression of previous data. They combine current support vectors with incremental examples as training set to reduce the training data. However, discarding all previous data except their support vectors will degrade the accuracy and give only approximate results. To solve above problem, we make some work from following aspects.

3.1 Defining importance weight for training data to reduce training data

The training data include historical examples and incremental data. So to save the memory and time, we can 1) reduce preserved historical examples, 2) reduce incremental data, 3) reduce whole training set. But to improve the accuracy, we increase historical examples instead of only preserving the support vectors as historical data. So we can reduce the incremental data and whole training data. Here we first consider reducing the whole training data.

In [13], the historical data is increased to improve the accuracy and size of training data is limited by threshold p to control the scale. When the preserved training set is full, their algorithm will remove oldest example from training data. This is not a good method because the oldest examples maybe include more important information and the size of training data need to be increased to guarantee the good accuracy. So we think about a new removing strategy called weighted-based. When the training set is full, we will remove less importance examples so that the same amount of training data include more information or the same amount of information is included in less example data. Obviously, example which is between the class is very important for generating SVMs. More specifically, the nearer to respective hyperplanes $f(x)=\pm 1$ a vector is, the more important it is. When a vector is farther away from corresponding hyperplane, it is more likely to be well classified or misclassified vector with low importance.

The necessary and sufficient condition of KKT conditions is: $y_i f(x_i) \geq 1$. If $y_i f(x_i) = 1$, vector (x_i, y_i) is on the boundary of the margin. If $y_i f(x_i) > 1$, vector (x_i, y_i) is outside the margin. $|y_i f(x_i)|$ is the relative distance to the hyperplane $f(x) = 0$. So we make a definition to measure importance of vector.

Definition: given vector $(x_i, y_i) \in X$, SVMs is from set X, $f(x) = 0$ is classification hyperplane. The importance of vector (x_i, y_i) to SVMs is define as:

$$I_{(x_i, y_i)} = ||y_i f(x_i)| - 1| = ||f(x_i)| - 1| \quad (1)$$

The smaller value of importance is, the more important the vector is. In the incremental SVMs, if a vector (x_i, y_i) is most important to all the SVMs generated, it is usually most likely to be the support vector. We use the average of importance of vector as weight to denote its importance.

3.2 Using Convex Hull and extension of KKT conditions to reduce incremental data

We reduce incremental data by using the geometric properties of SVMs to further decrease the memory consuming and computation time. The computation of SVMs is equivalent to the problem of computing the nearest points between two classes. If an incremental example violates KKT conditions, it will change previous SVMs. So some ones utility those conditions to reduce incremental data [10, 13]. But some examples that meet KKT conditions are likely to become support vectors, and accuracy could be degraded because these data is not trained. Here, we extend the KKT conditions to increase this kind of examples. At the same time, because the support vectors are usually local extremums or near extremums [9, 11, 14], we can use extremes' examples and delete inside samples from the whole incremental set to reduce training data. The main problem is how to find the boundary of a data set. Convex hull is widely applied in SVM classification to reduce training data. In this paper, we use the extension of KKT conditions $y_i f(x_i) \geq m$ and convex hull to reduce the incremental data set. We use SMO algorithm [8, 13] as a core SVM solver, because this method can converge quickly from good initial hypothesis.

3.3 WKH-SVMs Algorithm

We define below notation to describe WKH-SVMs algorithm:

1. X_{new}: newly arrived data set
2. X_{train}: training data set
3. Convex(X): a function computing Convex Hull of data set X
4. X_{near}: data set of near examples between class
5. X_{hull}: data set of Convex Hull
6. p: threshold of size of Xtrain
7. m: threshold of extension of KKT conditions

```
Given: C, m, p:
Initialize $w \leftarrow 0, b \leftarrow 0, X_{train} \leftarrow \{\ \}$
While $X_{new}$ is ready do
  $X_{near} \leftarrow \{\ \}$
  For each $(x_i, y_i) \in X_{new}$ do
    Classify $x_i$ by $f(x_i) = sign(\langle w, x_i \rangle + b)$
    If $y_i f(x_i) < m$ then $X_{near} \leftarrow (x_i, y_i)$
  $X_{hull} \leftarrow Convex(X_{near})$
  If $X_{hull} \equiv null$ then next round
                    else next step
  While $size(X_{hull}) + size(X_{train}) > p$
    Remove biggest weight example
        from $X_{train}$
  Compute the weight of each $(x_i, y_i) \in X_{hull}$
  Update the weight of each $(x_i, y_i) \in X_{train}$
  $X_{train} \leftarrow X_{hull}$
  Find $w', \text{and}\, b'$ with SMO on $X_{train}$
    using $w\, \text{and}\, b$ as seed hypothesis.
  Set $(w, b) \leftarrow (w', b')$
done
```

Ⅳ. Performance Analysis

In this paper, our algorithm efficiently preserve training samples with important information for training process by adding importance weight to examples and reduce the size of incremental examples set to form smaller training data set whose examples are most likely to be support vectors.

Firstly, it use a threshold m to extract examples which is nearer to the margin between the class from incremental set. Threshold m is bigger than 1, so that we can get the non-SV with important information to improve the classification accuracy. If m<1, this algorithm will be similar to the algorithm in [13]. If m=1, this algorithm will be similar to traditional one. Secondly, those extracted examples further are reduced by computing those convex hull. Those vectors on the convex hull are put into training dataset as next round training examples.

Thirdly, threshold p limits the scale of training set, so that training process is finished in limited training time and it avoid large computation time in large scale training set.

Finally, importance weight is used to extract more important examples for SVMs in the training set, less important examples will be replaced by chosen incremental examples. So we can make threshold p smaller to save the memory and computation time. At the same time, almost all misleading examples are removed as less importance examples. This is one of the reasons why the classification accuracy is improved with our WKH-SVMs.

Ⅴ. Conclusion and Future Research

In this paper, we present some research on incremental SVMs and propose a new incremental SVMs algorithm based on importance weight and extension of KKT conditions and convex hull. Finally, we make some performance analysis in theory and find some advantages. Our future work is to improve our algorithm and to make some experiments to test our WKH-SVMs in detecting system.

References

[1] N. Jindal and B. Liu. Analyzing and detecting review spam. In Proceedings of the 7th IEEE International Conference on Data Mining (ICDM), pages 547 - 552, Los Alamitos, CA, IEEE Computer Society. 2007.

[2] Lim E-P,Nguyen V-A,Jindal N,et al. Detecting product review spammers using rating behaviors [C]MProceedings of the 19th ACM international conference on Information and knowledge management. Toronto, ON, Canada: ACM, 930-948, 2010.

[3] N. Syed, H. Liu, and K. Sung. Handling concept drifts in incremental learning with support vector machines. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99), San Diego, CA, USA, 1999

[4] Coppock H, Freund J. All or None versus Incremental Learning of Errorless Shock Escapes by the Rat. Science, 135: 318－319, 1962.

[5] R. Polikar, L. Udpa, S. S. Udpa, and V. Honavar, "Learn++: an incremental learning algorithm for supervised neural networks", IEEE Trans. Syst., Man, Cybern. C, vol. 31, pp.497 -508 2001

[6] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In NIPS, pages 409 – 415, 2000.

[7] B. Scholkopf and A. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2001.

[8] J. Platt. Sequenital minimal optimization: A fast algorithm for training support vector machines. In B. Scholkopf, C. Burges, and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning. MIT Press, 1998.

[9] X. J. Peng and Y. F. Wang, “Geometric algorithms to large margin classifier based on affine hulls,”IEEE Trans. Neural Netw. Learn. Syst.,vol. 23, no. 2, pp. 236 – 246, Feb. 2012.

[10] Franc V, Hlavac V, An iterative algorithm learning the maximal margin classifier. Pattern Recogn Lett 36:1985 – 1996, 2003.

[11] Mavroforakis ME, Theodoridis S, A geometric approach to support vector machine (SVM) classification. IEEE Trans Neural Netw 17(3):671 – 682, 2006

[12] K. Lau and Q. Wu, “Online training of support vector classifier,”Pattern Recognit., vol. 36, no. 8, pp. 1913 – 1920, 2003.

[13] Sculley D, Wachman GM, Relaxed online SVMs for spam filtering. In: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval. pp 415 – 422, 2007

[14] Bennett K. and Bredensteiner E. Duality and Geometry in SVMs. In P. Langley editor, Proc. of 17 th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, 65-72, 2000