International Journal of Database Theory and Application
Vol.10, No.1 (2017), pp.79-88
http://dx.doi.org/10.14257/ijdta.2017.10.1.08

Semi-supervised Text Classification Using SVM with Exponential Kernel

Liyun Zhong*

School of Mathematics and Computer Science, Gannan Normal University, Ganzhou 341000, P.R. China;
wth2003nc@163.com

Abstract

Kernel-based learning methods (kernel methods for short) in general and support vector machine (SVM) in particular have been successfully applied to the task of text classification. This is mainly due to their relatively high classification accuracy on several application domains as well as their ability to handle high dimensional and sparse data which is the prohibitive characteristics of textual data representation. A significant challenge in text classification is to reduce the need for labeled training data while maintaining an acceptable performance. This paper presents a semi-supervised technique using the exponential kernel for text classification. Specifically, the semantic similarities between terms are first determined with both labeled and unlabeled training data by means of a diffusion process on a graph defined by lexicon and co-occurrence information, and the exponential kernel is then constructed based on the learned semantic similarity. Finally, the SVM classifier trains a model for each class during the training phase and this model is then applied to all test examples in the test phase. The main feature of this approach is that it takes advantage of the exponential kernel to reveal the semantic similarities between terms in an unsupervised manner, which provides a kernel framework for semi-supervised learning. The proposed approach is demonstrated on several benchmark data sets for text classification and the experimental results show that it can significantly improve the classification performance.

Keywords*: semi-supervised learning, text classification, exponential kernel, support vector machine (SVM)*

1. Introduction

The widespread and increasing availability of massive textual data stimulates the development of text categorization field, which aims to automatically classify unlabeled documents into predefined categories according to some criteria of interest [1-2]. Categories are usually defined according to a variety of topics (*e.g.* SPORT vs. POLITICS) and a set of hand tagged examples is provided for training. Pioneered by [3], kernel methods [4] such as support vector machine (SVM) [5] have been heavily used for text categorization tasks, typically showing good results [6-17]. Basically, kernel methods work by mapping the data from the input space into a high-dimensional (possibly infinite) feature space, which is usually chosen to be a reproducing kernel Hilbert space (RKHS), and then building linear algorithms in the feature space to implement nonlinear counterparts in the input space. The mapping, rather than being given in an explicit form, is determined implicitly by specifying a kernel function, which computes the inner product between each pair of data points in the feature space. There are several reasons that make kernel methods applicable to text categorization. Firstly, instead of manual construction of feature space for the learning task, kernel functions provide an alternative

* Corresponding author.

ISSN: 2005-4270 IJDTA
Copyright ⓒ 2017 SERSC

way to design useful features in the feature space automatically, therefore, ensuring necessary representational power. Secondly, kernel methods offer a flexible and efficient way to define application-specific kernels for introducing background knowledge and modeling explicitly linguistic insights. This property allows to notably improve the performance of the general learning methods and their simple adaptation to the specific application. Finally, kernel methods can be naturally applied to the non-vectorial types of data, thus taking into account the structure of the data and greatly reducing the need for careful feature engineering in these structures.

From the point of view of modularization, kernel methods consist of two main components, namely the kernel and actual learning algorithm. The kernel can be considered as an interface between the input data and the learning algorithm, and is the key component to ensure the good performance of kernel methods [4, 18]. Actually, for real applications, kernel is the only task-specific component of kernel methods. In the domain of text categorization, the widely used kernel is the "Bag of Words" (BOW) kernel [4], which encodes the input documents as vectors whose dimensions correspond to the words or terms occurring in the corpus. Despite its ease of use, this kernel suffers from well-known limitations, mostly due to its inability to exploit semantic similarity between terms: documents sharing terms that are different but semantically related will be considered as unrelated. To address this problem, a number of attempts have been made to incorporate semantic knowledge into the BOW kernel, resulting in the so-called semantic kernels [4]. For example: the semantic kernels that use the external semantic knowledge like WordNet and Wikipeida were proposed to improve the kernel-based text categorization systems [6-8]. In the absence of external semantic knowledge, corpus-based statistical approaches are applied to capture the semantic relations between terms, resulting in the corpus-based semantic kernels [9-14]. Besides these kernels which are only suitable for vectorial types of data, the kernels suitable for non-vectorial types of data were also proposed for text categorization, such as string kernel [15], word-sequence kernel [16] and tree kernel [17]. Finally, it is advisable to combine different kernels in order to identify a good target kernel for text categorization [2].

In this paper, we only consider the exponential kernel [10, 12, 19-21], which is one of the corpus-based semantic kernels, for text categorization. Conceptually, exponential kernel can be obtained through a matrix exponentiation transformation on the given kernel matrix, and virtually exploits higher order co-occurrences to infer semantic similarity between terms. Geometrically, this kernel models semantic similarities by means of a diffusion process on a graph defined by lexicon and co-occurrence information. More importantly, it should be noted that the diffusion is an unsupervised process, which naturally provides a kernel framework for semi-supervised learning. A significant challenge in text classification is to reduce the need for labeled training data while maintaining an acceptable performance. To address this challenge, we present a semi-supervised technique for text classification. Specifically, the semantic similarities between terms are first determined by the diffusion process using both labeled and unlabeled training data, and the exponential kernel is then constructed based on the learned semantic similarity. Finally, the SVM classifier trains a model for each class during the training phase and this model is then applied to all test examples in the test phase. The proposed approach is demonstrated with two benchmarks for text classification.

The rest of this paper is organized as follows. Section 2 briefly introduces SVM. Section 3 then details the semi-supervised text classification procedure using the exponential kernel. Experimental results are reported in Section 4, followed by some conclusions in Section 5.

 Copyright ⓒ 2017 SERSC

2. Support Vector Machine

In general, kernel methods map data points from the input space to some feature space where even relatively simple algorithms such as linear methods can deliver very impressive performance [4-5]. The most attractive feature of kernel methods is that they can be applied in high dimensional feature spaces without suffering from the high cost of explicitly computing the feature map. This is possible with the *kernel trick*, *i.e.*, using a valid kernel function k on any set X (input space). A function $k(\boldsymbol{x}, \boldsymbol{x}')$ is a valid kernel if and only if for any finite set it produces symmetric and positive semi-definite Gram matrices. For such $k: \mathrm{X} \times \mathrm{X} \to \mathrm{R}$ (R denotes the set of real numbers), it is known that a mapping $\phi: \mathrm{X} \to \mathrm{F}$ (F denotes the feature space induced by a kernel function) into a reproducing kernel Hilbert space (RKHS), such that $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$ for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathrm{X}$. Popular kernel functions include linear kernel, polynomial kernel and Gaussian kernel. With a kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for any $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathrm{X}$, the Gram matrix or kernel matrix is given by $[\mathbf{K}]_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Since the Gram matrix and kernel function are essentially equivalent, we can refer to one or the other as "kernel" without risk of confusion.

We here consider the SVM, the most well-known kernel method in practice. In a binary classification problem, we are given l pairs of training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, where $\boldsymbol{x}_i \in \mathrm{X}$ and $y_i \in \{+1, -1\}$. The standard SVM tries to find a hyperplane $\boldsymbol{w}^{\mathrm{T}} \phi(\boldsymbol{x}) + b = 0$, which is determined by a weight vector $\boldsymbol{w}$ and a bias b. This hyperplane can be obtained by solving the following optimization problem:

$$\begin{aligned} &\min \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l} \xi_i \\ &\text{s.t.} \ \ y_i(\boldsymbol{w}^{\mathrm{T}} \phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0, \ \ i = 1, \cdots, l \end{aligned} \tag{1}$$

where $\boldsymbol{\xi} = (\xi_1, \cdots, \xi_l)^{\mathrm{T}}$ is the vector of slack variables and C is the regularization parameter used to impose a trade-off between the training error and generalization. This problem can be solved using the Lagrange method. Suppose α_i be the Lagrange multiplier corresponding to the ith inequality, the dual problem of (1) is shown to be

$$\begin{aligned} &\max \ \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &\text{s.t.} \ \ \sum_{i=1}^{l} \alpha_i y_i = 0, \\ &\qquad 0 \leq \alpha_i \leq C, \ \ i = 1, \cdots, l \end{aligned} \tag{2}$$

After the solution is obtained, the resultant decision function can be formulated as

$$f(\boldsymbol{x}) = \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b \right) \tag{3}$$

where the samples $\boldsymbol{x}_i$ with $\alpha_i > 0$ are called support vectors.

Copyright ⓒ 2017 SERSC

For multiclass classification problems, there are several approaches available to extend binary SVM to multiclass SVM [22-23]. These approaches roughly fall into two categories. The first denoted as all-in-one or single machine is to directly consider all data in one optimization formulation. The second involves considering a decomposition of a multiclass problem into several binary subproblems and then combining their solutions. There are two widely used strategies to decompose a multiclass problem: one-versus-rest (1-v-r) and one-versus-one (1-v-1). Given a problem with m classes, the 1-v-r strategy constructs m binary SVMs, in which each of them is trained to separate one class from the other classes, while the 1-v-1 strategy constructs $m(m-1)/2$ binary SVMs, in which each of them is trained to separate one class from another class. When a test sample is provided, it is applied to all the binary SVMs and their outputs are combined based on some voting techniques, such as "MaxWins" voting scheme which counts how often each class is output by the binary SVMs and the test sample is then assigned to the most voted class. Although both approaches present usually no significant difference in classification accuracy when the parameters of SVM are properly tuned, the decomposition one is often recommended for practical use because of lower computational overhead and conceptual simplicity.

3. Proposed Semi-supervised Text Classification Approach

3.1. Exponential Kernel

Let $\boldsymbol{S} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_l)$ be a set of documents. Consider that we are also given a dictionary V consisting of n words. The BOW model (also called vector space model, VSM) [3, 4] of the document $\boldsymbol{x}$ is defined as follows:

$$\phi : \boldsymbol{x} \to \phi(\boldsymbol{x}) = (tf(t_1, \boldsymbol{x}), \cdots, tf(t_n, \boldsymbol{x}))^{\mathrm{T}} \in \mathrm{R}^n \tag{4}$$

where $tf(t_i, \boldsymbol{x})$, $1 \le i \le n$, is the frequency of the occurrence of the word t_i in the document $\boldsymbol{x}$. If we consider the feature space defined by the VSM, the BOW kernel is given by the inner product between the feature vectors:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = < \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) > = \sum_{t \in \mathrm{V}} tf(t, \boldsymbol{x}_i) tf(t, \boldsymbol{x}_j) \tag{5}$$

BOW model is probably one of the simplest constructions used in text processing. In this model, the feature vectors are typically sparse with a small number of non-zero entries for those words occurring in the documents. Two documents that use semantically related but distinct words will therefore show no similarity. Ideally, semantically similar documents should be mapped to nearby positions in the feature space. In order to address this problem, a transformation of the feature vector of the type $\bar{\phi}(\boldsymbol{x}) = \mathbf{S}\phi(\boldsymbol{x})$ is required, where $\mathbf{S}$ is a semantic matrix indexed by pairs of words with the entries $[\mathbf{S}]_{i,j} = [\mathbf{S}]_{j,i}$, $1 \le i, j \le n$, indicating the strength of their semantic similarity. Using this transformation, the semantic kernels take the form of

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \bar{\phi}(\boldsymbol{x}_i)^{\mathrm{T}} \bar{\phi}(\boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \mathbf{S}^{\mathrm{T}} \mathbf{S} \phi(\boldsymbol{x}_j) \tag{6}$$

The semantic kernels correspond to representing a context as a less sparse vector $\mathbf{S}\phi(\boldsymbol{x})$, which has non-zero entries for all terms that are semantically similar to those presented in the document $\boldsymbol{x}$. Different choices of matrix $\mathbf{S}$ lead to different variants of semantic kernels, such as latent semantic kernel [9], domain kernel [11] and class weighting kernel [14].

 Copyright ⓒ 2017 SERSC

In practice, the problem of how to infer semantic similarities between terms from a corpus remains an open issue. Kandola *et al.* [10] proposed a semantic kernel named exponential kernel given by

$$\mathbf{K}(\lambda) = \mathbf{K}_0 \exp(\lambda \mathbf{K}_0) \tag{7}$$

where $\mathbf{K}_0$ is the kernel matrix of the BOW kernel, $\lambda \in [0, +\infty)$ is a decay factor. Let $\mathbf{D}$ be the feature example (term-by-document) matrix in the BOW representation, then $\mathbf{K}_0 = \mathbf{D}^{\mathrm{T}}\mathbf{D}$. Let $\mathbf{G} = \mathbf{D}\mathbf{D}^{\mathrm{T}}$, it is easy to prove that $\mathbf{K}(\lambda)$ corresponds to a semantic matrix $\exp(\lambda \mathbf{G}/2)$ [10], *i.e.*

$$\mathbf{S} = \exp\left(\lambda \mathbf{G}/2\right) = \left(\sum_{d=0}^{\infty} \frac{\lambda^d}{d!} \mathbf{G}^d \right)^{1/2} = \left(\mathbf{I} + \lambda \mathbf{G} + \frac{\lambda^2}{2!} \mathbf{G}^2 + \cdots + \frac{\lambda^2}{d!} \mathbf{G}^d + \cdots \right)^{1/2} \tag{8}$$

where $\mathbf{I}$ denotes the identity matrix. In fact, noting that $\mathbf{S}$ is a symmetric positive semi-definite matrix since $\mathbf{G}$ is symmetric [24], we have

$$\begin{aligned} \mathbf{K}(\lambda) &= \mathbf{D}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{S}\mathbf{D} = \mathbf{D}^{\mathrm{T}}\mathbf{S}^2\mathbf{D} = \mathbf{D}^{\mathrm{T}} \exp\left(\lambda \mathbf{G}\right)\mathbf{D} \\ &= \sum_{d=0}^{\infty} \frac{\lambda^d}{d!} \mathbf{D}^{\mathrm{T}}\mathbf{G}^d\mathbf{D} = \mathbf{K}_0 \left(\sum_{d=0}^{\infty} \frac{\lambda^d}{d!} \mathbf{K}_0^d \right) \\ &= \mathbf{K}_0 \exp(\lambda \mathbf{K}_0) \end{aligned} \tag{9}$$

Geometrically, exponential kernel models semantic similarities by means of a diffusion process on a graph defined by lexicon and co-occurrence information [10, 12, 19-21]. Specifically, such a graph has nodes indexed by all the terms in the corpus, and the edges are given by the co-occurrence between terms in documents of the corpus. A diffusion process on the graph can capture higher order co-occurrences between indirectly connected terms. Conceptually, if term t_1 co-occurs with term t_2 in some documents, we say t_1 and t_2 share a first-order correlation between them. If t_1 co-occurs with t_2 in some documents, and t_2 with t_3 in some others, then t_1 and t_3 are said to share a second-order correlation through t_2. Higher orders of correlation can be similarly defined. Noting that $\left[\mathbf{G}^d\right]_{i,j}$ is the number of dth -order co-occurrence paths between terms t_i and t_j in the graph [1], and the semantic matrix $\mathbf{S}$ combines all the order co-occurrence paths with exponentially decaying weights, we can easily find that the semantic similarity between two terms is measured by the number of the co-occurrence paths between them, and the semantic matrix $\mathbf{S}$ essentially exploits the higher order correlation between terms. Intuition shows that the higher the co-occurrence order is, the less similar the semantics becomes. The parameter λ is used to control the decaying speed for increasing orders. To summarize, exponential kernel takes all possible paths connecting two nodes into account, and propagates the similarity between two remote terms (or documents) in an elegant way. In addition, it is obvious that the exponential kernel is reduced to the standard BOW kernel when $\lambda = 0$. In other words, the BOW kernel is just a special case of the exponential kernel.

[1] The identity matrix $\mathbf{I}$ (*i.e*, $\mathbf{G}^0$) can be regarded as the indication of the zero-order correlation between terms, meaning only the similarity between a term and itself equals 1 and 0 for other cases.

Copyright ⓒ 2017 SERSC

3.2. Semi-supervised Text Classification Procedure Using Exponential Kernel

As mentioned before, the elements of the semantic matrix $\mathbf{S}$ give the strength of the semantic similarity between terms. Exponential kernel essentially exploits the higher order correlations to refine the similarity measure by performing a diffusion process on a graph defined by lexicon and co-occurrence information. It is obvious that the diffusion is an unsupervised process, which naturally provides a kernel framework for semi-supervised learning. In semi-supervised learning we are given a labeled data set $L=\{(\boldsymbol{x}_1, y_1),(\boldsymbol{x}_2, y_2),\cdots,(\boldsymbol{x}_l, y_l)\},\ y_i \in\{1,2,\cdots,c\},\ i\in\{1,2,\cdots,l\}$ and a an unlabelled data set $U=\{\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2},\cdots,\boldsymbol{x}_{l+u}\}$. We here propose a 4-step kernel method framework for semi-supervised text classification:

1) **Preprocessing input documents.** This step converts the input documents into formatted information. The details of this step will be described in Section 4.1. After this procedure, we are given the formatted L and U.
2) **Learning semantic matrix.** This step determines the semantic similarities between terms with both L and U by means of a diffusion process.
3) **Constructing exponential kernel.** This step constructs the exponential kernel based on the learned semantic matrix using (6).
4) **Using common kernel algorithms**, such as SVM.

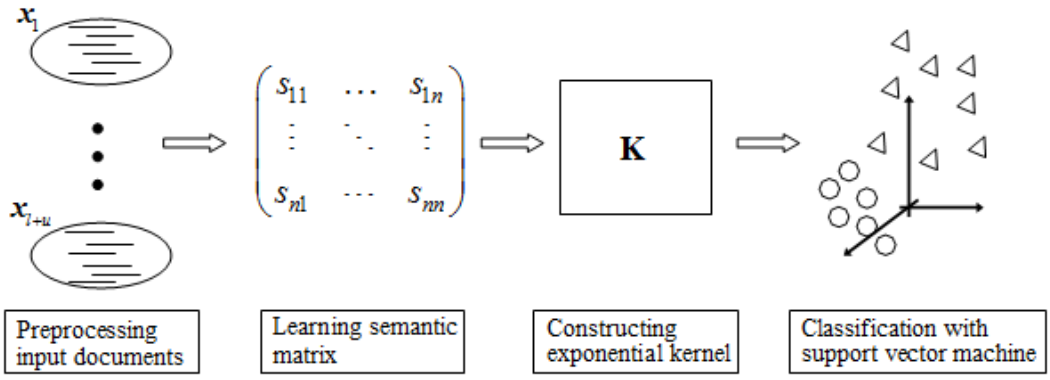

Figure 1. Semi-supervised Text Classification System Architecture Based on Exponential Kernel

Figure 1 demonstrates the architecture of the proposed semi-supervised text classification system based on the exponential kernel.

4. Experimental Results

This experiment evaluates the performance of the proposed approach on several textual data sets. Table 1 shows the details of the selected data sets which are variants of the popular 20Newsgroup data set [2]. It presents, for each data set, the number of samples, the number of features and the number of classes. The 20Newsgroup data set is a collection of approximately 20000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups and commonly used in the text mining domain. Some of the newsgroups are very closely related to each other (*e.g.* comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (*e.g.* misc.forsale / soc.religion.christian). We used four basic subgroups "SCIENCE", "COMP", "POLITICS" and "RELIGION" from the 20Newsgroup data set. The documents are

[2] http://www.ai.mit.edu/people/jrennie/20Newsgroups/.

 Copyright ⓒ 2017 SERSC

evenly distributed to the classes (each class including 500 documents). All data sets were preprocessed using the Text Mining Infrastructure (TMI) [25]. The preprocessing includes sentence boundary determination, stop word removal and stemming. We used the stemmer and stop word list embedded in the TMI. Rare terms which occur in less than three documents were filtered and Information Gain based feature selection method was used to select the most informative 2000 terms.

Table 1. Statistics of Selected Four Data Sets

Data set	#samples	#features	#classes
20NewsGroup-SCIENCE	2000	2225	4
20News-COMP	2500	2478	5
20News-POLITICS	1500	2478	3
20News-RELIGION	1500	2125	3

After the proper preprocessing, we used the LIBLINEAR package [26] to train and test the SVM model. We consider two types of kernels, *i.e.*, BOW kernel and exponential kernel for comparison. These kernels are embedded in the SVM classifier individually. The parameters of the SVM were optimized by five-fold cross-validation on the training set. For the BOW kernel, there is only one parameter C that needs to be optimized. We performed grid-search in one dimension (*i.e.*, a line-search) to choose this parameter from the set $\{2^{-2}, 2^{0}, \cdots, 2^{10}\}$. For the exponential kernel, there are two parameters C and λ that need to be to be optimized. We perform grid-search over two dimensions, *i.e.*, $C = \{2^{-2}, 2^{0}, \cdots, 2^{10}\}$ and $\lambda = \{2^{-1}, 2^{-2}, \cdots, 2^{-10}\}$.

For each data set considered in Table 1, we partitioned it into three groups: 30% and 20% of the data set are used for training and prediction, respectively. The training set and test set are taken as the labeled data L, and the rest (50% of the data set) is taken as the unlabeled data U (we assume that the labels of the data are unknown). Stratified sampling is used to preserve the ratio of different classes in these three groups. Table 2 summarizes the average classification accuracy with standard deviations over 10 trials. The bold font indicates the best performance. For more reliable results rather than those which would be expected by chance, two-tailed t-test with the significant level 0.05 is performed to determine whether there is a significant difference between the proposed approach and other approaches. The win-tie-loss (W-T-L) summarizations based on t-test are attached at the bottom of Table 2. A win or a loss means that the proposed approach is better or worse than other approach on a data set. A tie means that both approaches have the same performance. From this table, we find that the exponential kernel produces significantly better classification performances than the BOW kernel baseline. This implies that the semantic similarities obtained by means of a diffusion process on a graph defined by lexicon and co-occurrence information can improve the classification performance. More importantly, for all data sets we see that the proposed approach achieves significant performance improvement over the exponential kernel. Take the *20NewsGroup-SCIENCE* and *20News-COMP* data sets for example: the proposed approach achieves the classification accuracy of 94.78% and 81.24% whereas the exponential kernel achieves those of 93.52% and 78.39%, respectively. In other words, the proposed approach achieves the classification accuracy with relative improvements of 1.35% ((94.78-93.52)/93.52) and 3.61% ((81.24-78.39)/78.39) over the exponential kernel, respectively. Since whether or not the unlabeled data U is taken into consideration is the only difference between the proposed approach and the exponential kernel, these results imply that the unlabeled data has a conspicuous impact on the kernel

Copyright ⓒ 2017 SERSC

construction for text classification and demonstrate the effectiveness of the proposed approach with application to text classification.

Table 2. Classification Accuracy of Different Approaches on Four Data Sets

Data set	Classification accuracy (%)		
	BOW kernel	Exponential kernel	Proposed approach
20NewsGroup-SCIENCE	87.93±1.32	93.52±1.71	**94.78±0.36**
20News-COMP	76.64±1.59	78.39±0.28	**81.24±0.59**
20News-POLITICS	93.04±1.06	93.46±1.43	**95.09±0.47**
20News-RELIGION	84.68±0.78	86.73±1.62	**88.16±0.92**
W-T-L	4-0-0	4-0-0	-

5. Conclusion

We have presented a novel exponential kernel based semi-supervised text classification approach which incorporates the unlabeled data into the diffusion process of mining higher order correlations between terms. The main feature of this approach is that it takes advantage of the exponential kernel to reveal the semantic similarities between terms in an unsupervised manner, which provides a kernel framework for semi-supervised learning. Experimental evaluation shows the superior effectiveness of the proposed approach compared with other baseline models. Since in text classification one of the significant issues is the insufficient usage of abundant useful but unlabeled data, our approach provides an alternative to reduce the need for labeled training data while maintaining an acceptable performance. Future work will focus on the theoretical verification of the superior performance of the proposed approach, as well as making comparisons with other newly proposed methods for text classification. We also plan to apply the generic diffusion kernel [27], which upgrades the exponential kernel to a general and flexible form, for semi-supervised text classification.

Acknowledgments

We would like to thank all the referees for their constructive and insightful comments on this paper. This work is supported in part by the National Natural Science Foundation of China (No. 61562003) and the Natural Science Foundation of Jiangxi Province of China (No. 20161BAB202070).

References

[1] F. Sebastiani, “Machine learning in automated text categorization”, ACM Computing Surveys, vol. 34, no. 1, **(2002)**, pp. 1-47.
[2] T. Wang, H. Xie, L. Zhong and S. Hu, “A multiple kernel learning approach to text categorization”, Journal of Computational and Theoretical Nanoscience, vol. 12, no. 9, **(2015)**, pp. 2121-2126.
[3] T. Joachims, “Text categorization with support vector machines: learning with many relevant features”, Proceedings of the 10th European Conference on Machine Learning, Chemnitz, Germany, **(1998)**, pp. 137-142.
[4] J. Shawe-Taylor and N. Cristianini, “Kernel methods for pattern analysis”, Cambridge University Press, New York, **(2004)**.
[5] T. Wang, H. Huang and S. Tian, “Feature selection for SVM via optimization of kernel polarization with Gaussian ARD kernels”, Expert Systems with Applications, vol. 37, no. 9, **(2010)**, pp. 6663-6668.
[6] G. Siolas and F. d’Alché-Buc, “Support vector machines based on a semantic kernel for text categorization”, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, Como, Italy, **(2000)**, pp. 205-209.

 Copyright ⓒ 2017 SERSC

[7] S. Bloehdorn, R. Basili, M. Cammisa and A. Moschitti, "Semantic kernels for text categorization based on topological measures of feature similarity", Proceedings of the 6th IEEE International Conference on Data Mining, Hong Kong, China, **(2006)**, pp. 808-812.
[8] P. Wang and C. Domeniconi, "Building semantic kernels for text categorization using Wikipedia", Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, USA, **(2008)**, pp. 713-721.
[9] N. Cristianini, J. Shawe-Taylor and H. Lodhi, "Latent semantic kernels", Journal of Intelligent Information Systems, vol. 18, nos. 2-3, **(2002)**, pp. 127-152.
[10] J. Kandola, J. Shawe-Taylor and N. Cristianini, "Learning semantic similarity", Advances in Neural Information Processing Systems, vol. 15, **(2003)**, pp. 657-664.
[11] A. M. Gliozzo and C. Strapparava, "Domain kernels for text categorization", Proceedings of the 9th Conference on Computational Natural Language Learning, Ann Arbor, USA, **(2005)**, pp. 56-63.
[12] J. Chen, J. Zhong, Y. Xie and C. Cai, "Text categorization using SVM with exponential kernel", Applied Mechanics and Materials, vols. 519-520, **(2014)**, pp. 807-810.
[13] B. Altınel, M. C. Caniz and B. Diri, "A corpus-based semantic kernel for text categorization by using meaning values of terms", Engineering Applications of Artificial Intelligence, vol. 43, **(2015)**, pp. 54-66.
[14] B. Altınel, B. Diri and M. C. Ganiz, "A novel semantic smoothing kernel for text categorization with class-based weighting", Knowledge-Based Systems, vol. 89, **(2015)**, 265-277.
[15] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini and C. Watkins, "Text categorization using string kernels", Journal of Machine Learning Research, vol. 2, **(2002)**, pp. 419-444.
[16] N. Cancedda, E. Gaussier, C. Goutte and J-M. Renders, "Word-sequences kernels", Journal of Machine Learning Research, vol. 3, **(2003)**, pp. 1059-108.
[17] T. Goncalves and P. Quaresma, "Text categorization using tree kernels and linguistic information", Proceedings of the 7th International Conference on Machine Learning and Applications, San Diego, USA, **(2008)**, pp. 763-768.
[18] T. Wang, D. Zhao and S. Tian, "An overview of kernel alignment and its applications", Artificial Intelligence Review, vol. 43, no. 2, **(2015)**, pp. 179-192.
[19] T. Wang, J. Rao and D. Zhao, "Using exponential kernel for word sense disambiguation", Proceedings of the 23rd International Conference on Artificial Neural Networks, Sofia, Bulgaria, **(2013)**, pp. 545-552.
[20] T. Wang, J. Rao and Q. Hu, "Supervised word sense disambiguation using semantic diffusion kernel", Engineering Applications of Artificial Intelligence, vol. 27, **(2014)**, pp. 167-174.
[21] T. Wang and W. Li, "Learning class-informed semantic similarity", Proceedings of the 23rd International Conference on Neural Information Processing, Kyoto, Japan, Part III, LNCS 9949, **(2016)**, pp. 442–449.
[22] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines", IEEE Transactions on Neural Networks, vol. 13, no. 2, **(2002)**, pp. 415-425.
[23] T. Wang, D. Zhao and Y. Feng, "Two-stage multiple kernel learning with multiclass kernel polarization", Knowledge-Based Systems, vol. 48, **(2013)**, pp. 10-16.
[24] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures", Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, **(2002)**, pp. 315-322.
[25] L. E. Holzman, T. A. Fisher, L. M. Galitsky, A. Kontostathis and W. M. Pottenger, "A software infrastructure for research in textual data mining", International Journal on Artificial Intelligence Tools, vol. 14, no. 4, **(2004)**, pp. 829-849.
[26] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang and C. J. Lin, "IBLINEAR: A library for large linear classification", Journal of Machine Learning Research, vol. 9, **(2008)**, pp. 1871-1874.
[27] L. Jia and S. Liao, "A generic diffusion kernel for semi-supervised learning", Proceedings of the 5th International Symposium on Neural Networks, Beijing, China, Part I, LNCS 5263, **(2008)**, pp. 723-732.

Copyright ⓒ 2017 SERSC

Copyright ⓒ 2017 SERSC