*International Journal of Innovative Computing, Information and Control*
Volume **13**, Number **2**, April **2017**

 ISSN 1349-4198
pp. 695–701

# SEMI-SUPERVISED WORD SENSE DISAMBIGUATION USING VON NEUMANN KERNEL

Wensheng Zhu

Modern Education Technology Center
Gannan Normal University
Economic & Technological Development Zone, Ganzhou 341000, P. R. China
cjtgnnu@163.com

Received September 2016; revised January 2017

Abstract. *Kernel methods such as support vector machine (SVM) and kernel principal component analysis (KPCA) have been successfully applied to word sense disambiguation (WSD), which aims at identifying which sense of a word is used in a sentence, when the word has multiple meanings. This paper proposes using the von Neumann kernel for semi-supervised WSD. 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 von Neumann 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 property of this method is that it takes advantage of the von Neumann 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 with several SENSEVAL benchmark examples.*
**Keywords:** Semi-supervised learning, Word sense disambiguation, Von Neumann kernel, Support vector machine

1. **Introduction.** In computational linguistics, word sense disambiguation is an open problem of natural language processing and ontology. WSD is defined as the task of automatically assigning the most appropriate meaning to a polysemous word in a given context [1]. The solution to this problem impacts other computer-related writing, such as discourse, improving relevance of search engines, anaphora resolution, coherence, and inference. During the past decade, many supervised machine learning algorithms have been used for the task of automatic WSD, among which, kernel methods [2], such as SVM, regularized least-squares classification (RLSC) and kernel principal component analysis (KPCA), have demonstrated excellent performance in terms of accuracy and robustness [3-13]. 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 only task-specific component of kernel methods. In the domain of WSD, the widely used kernel is the "Bag of Words" (BOW) kernel [13], which is based on the BOW representation of the context in which an ambiguous word occurs. In this representation, each word or term constitutes a dimension in a vector space, independent of other terms in the same context. Despite its ease of use, this kernel suffers from well-known limitations, mostly due to its inability to exploit semantic similarity between terms: contexts 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 [13]. For example, the

semantic kernels that use the external semantic knowledge provided by word thesauri or ontology were proposed to improve the kernel-based WSD system [6]. In the absence of external semantic knowledge, latent semantic indexing (LSI) technology was applied to capturing the semantic relations between terms [8].

Recently, Wang et al. [10,11] proposed applying the semantic diffusion kernel [14] to improving the WSD system. Semantic diffusion 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. The diffusion is an unsupervised process, which naturally provides a kernel framework for semi-supervised learning. A significant challenge in WSD 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 WSD based on the von Neumann kernel. The main property of this technique is that the semantic similarities between terms are first determined by the diffusion process using both labeled and unlabeled training data, and the von Neumann kernel is then constructed based on the learned semantic similarity. Experiments on several SENSEVAL benchmark data sets demonstrate the proposed approach is sound and effective.

2. **An Overview of Von Neumann Kernel.** In machine learning-based WSD systems, the features extracted from the contexts are usually in the BOW representation which reduces a text to a histogram of word frequencies [10,11]. Let $t_0$ denote the word to be disambiguated and $\boldsymbol{x} = (t_{-r}, \ldots, t_{-1}, t_1, \ldots, t_s)$ be the context of $t_0$, where $t_{-r}, \ldots, t_{-1}$ are the words in the order they appear preceding $t_0$, and correspondingly $t_1, \ldots, t_s$ are the words that follow $t_0$ in the text. We define a context span parameter $\tau$ to control the length of the context. For a fixed $\tau$, we take always the largest context so that $r \leq \tau$ and $s \leq \tau$. Note that if there exist $\tau$ words preceding and following the word to be disambiguated, respectively, then $r = s = \tau$, otherwise $r < \tau$ or $s < \tau$. Consider that we are also given a vocabulary $V$ consisting of $n$ words, which can be extracted from all the contexts in the training corpus. The BOW model (also called vector space model, VSM) of the context $\boldsymbol{x}$ is given by

$$\phi : \boldsymbol{x} \to \phi(\boldsymbol{x}) = (tf(t_1, \boldsymbol{x}), \ldots, tf(t_n, \boldsymbol{x}))^{\mathrm{T}} \in \mathrm{R}^n \tag{1}$$

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

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = < \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) > = \sum_{t \in V} tf(t, \boldsymbol{x}_i) tf(t, \boldsymbol{x}_j) \tag{2}$$

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 contexts. Two contexts that use semantically related but distinct words will therefore show no similarity. Ideally, semantically similar contexts 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 $[\mathbf{S}]_{i,j} = [\mathbf{S}]_{j,i}$, $1 \leq i, j \leq n$, indicating the strength of their semantic similarity. Using this transformation, the semantic kernels take the form of

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \bar{\phi}(\boldsymbol{x}_i)^{\mathrm{T}} \bar{\phi}(\boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \mathbf{S}^{\mathrm{T}} \mathbf{S} \phi(\boldsymbol{x}_j) \tag{3}$$

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 context $\boldsymbol{x}$.

In practice, the problem of how to infer semantic similarities between terms from a corpus remains an open issue. Kandola et al. [14] proposed a semantic kernel named von Neumann kernel given by

$$\mathbf{K}(\lambda) = \mathbf{K}_0(\mathbf{I} - \lambda\mathbf{K}_0)^{-1} \tag{4}$$

where $\mathbf{K}_0$ is the kernel matrix of the BOW kernel, and $\lambda \in [0, ||\mathbf{K}_0||^{-1})$ is a decay factor. Let $\mathbf{D}$ be the feature example (term-by-document) matrix in the BOW representation, and then $\mathbf{K}_0 = \mathbf{D}^{\mathrm{T}}\mathbf{D}$. Let $\mathbf{G} = \mathbf{D}\mathbf{D}^{\mathrm{T}}$, and it is easy to prove that $\mathbf{K}(\lambda)$ corresponds to a semantic matrix $(\mathbf{I} - \lambda\mathbf{G})^{-1/2}$ [14], i.e.,

$$\mathbf{S} = (\mathbf{I} - \lambda\mathbf{G})^{-1/2} = \left(\sum_{d=0}^{\infty} \lambda^d \mathbf{G}^d\right)^{1/2} = \left(\mathbf{I} + \lambda\mathbf{G} + \lambda^2\mathbf{G}^2 + \cdots + \lambda^d\mathbf{G}^d + \cdots\right)^{1/2} \tag{5}$$

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 [15], we have

$$\begin{aligned} \mathbf{K}(\lambda) &= \mathbf{D}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{S}\mathbf{D} = \mathbf{D}^{\mathrm{T}}\mathbf{S}^2\mathbf{D} = \mathbf{D}^{\mathrm{T}}(\mathbf{I} - \lambda\mathbf{G})^{-1}\mathbf{D} \\ &= \sum_{d=0}^{\infty} \lambda^d \mathbf{D}^{\mathrm{T}}\mathbf{G}^d\mathbf{D} = \mathbf{K}_0\left(\sum_{d=0}^{\infty} \lambda^d \mathbf{K}_0^d\right) = \mathbf{K}_0(\mathbf{I} - \lambda\mathbf{K}_0)^{-1} \end{aligned} \tag{6}$$

It is obvious that when $\lambda = 0$, the von Neumann kernel is reduced to the standard BOW kernel. In other words, the BOW kernel is just a special case of the von Neumann kernel.

3. **Semi-Supervised WSD Procedure Using Von Neumann Kernel.** In the geometrical viewpoint, von Neumann kernel models semantic similarities by means of a diffusion process on a graph defined by lexicon and co-occurrence information [10,11,14]. 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 $d$th-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 polynomially 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 $\lambda$ is used to control the decaying speed for increasing orders. To summarize, von Neumann kernel takes all possible paths connecting two nodes into account, and propagates the similarity between two remote terms (or documents) in an elegant way.

As mentioned before, the elements of the semantic matrix $\mathbf{S}$ give the strength of the semantic similarity between terms. Von Neumann 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

[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.

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 $L = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_l, y_l)\}$, $y_i \in \{1, 2, \ldots, c\}$, $i \in \{1, 2, \ldots, l\}$ and an unlabelled data set $U = \{\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_{l+u}\}$. We here propose a 4-step kernel method framework for semi-supervised WSD.

**Step 1: Preprocessing input documents.** This step converts the input documents into formatted information. The details of this step will be described in the evaluation section. After this procedure, we are given the formatted $L$ and $U$.

**Step 2: Learning semantic matrix.** This step determines the semantic similarities between terms with both $L$ and $U$ by means of a diffusion process.

**Step 3: Constructing von Neumann kernel.** This step constructs the von Neumann kernel based on the learned semantic matrix using (3).

**Step 4: Using common kernel algorithms**, such as SVM and RLSC.

The main property of the proposed method is that the semantic similarities between terms are first determined by the diffusion process using both labeled and unlabeled training data (Step 2), and the von Neumann kernel is then constructed based on the learned semantic similarity (Step 3). For Step 4, although there are many kernel-based learning algorithms for selection, we here choose SVM as the learning machine since the last decade has witnessed an explosion of the use of SVM for WSD, both in the lexical sample and all-words exercises. In view of that WSD is a multiclass classification problem and SVM was originally proposed for solving binary classification problems, we need to extend binary SVM to multiclass SVM.

There are several approaches available to extend binary SVM to multiclass SVM [16,17]. 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.

4. **Experimental Setup.** This experiment evaluates the performance of the proposed method with several SENSEVAL[2] benchmark examples. SENSEVAL is the international organization devoted to the evaluation of WSD systems. We select the data sets for four words, namely *interest*, *line*, *hard* and *serve*, which have been used in numerous comparative studies of WSD. Table 1 provides the statistics of these data sets. It presents, for each data set, the number of instances, the number of senses, the minimum and maximum sense tag frequencies and percentages. It is easy to find that for all the data sets, the distribution of senses is severely skewed. For example, for the *hard* data set, the distribution of senses is skewed with almost 80% of the instances used in the most popular sense.

[2]http://www.senseval.org/

TABLE 1. Statistics of four selected data sets

| dataset | #instances | #senses | #min | #max |
|---|---|---|---|---|
| interest | 2368 | 6 | 11 (0.46%) | 1252 (52.87%) |
| line | 4147 | 6 | 349 (8.42%) | 2218 (53.48%) |
| hard | 4333 | 3 | 376 (8.68%) | 3455 (79.74%) |
| serve | 4378 | 4 | 439 (10.03%) | 1814 (41.43%) |

For each data set considered in Table 1, we partition 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. For the training data, we first remove the words that are in a list of stop words (for example: "is", "are", "a" and "the"). Words that contain no alphabetic characters, such as punctuation symbols and numbers, are also discarded. We then extract the surrounding words, which can be in the current sentence or immediately adjacent sentences, within the $\pm 5$-word window (i.e., $r = s = \tau = 5$) context of an ambiguous word. The extracted words are finally converted to their lemma forms in lower case. Each lemma is considered as one feature and its value is set to be the "term frequency". For the test set and unlabeled data, the similar preprocessing is carried out but the features are the same as those extracted from the training set (we directly eliminate those lemmas found in the test set or unlabeled data but not in the training set).

After the proper preprocessing, we use the LIBLINEAR package [18] to train and test the SVM model. We consider two types of kernels, i.e., BOW kernel and von Neumann kernel for comparison. These kernels are embedded in the SVM classifier individually. The parameters of the SVM are 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 perform grid-search in one dimension (i.e., a line-search) to choose this parameter from the set $\{2^{-2}, 2^{0}, \ldots, 2^{10}\}$. For the von Neumann kernel, there are two parameters $C$ and $\lambda$ that need to be optimized. We perform grid-search over two dimensions, i.e., $C = \{2^{-2}, 2^{0}, \ldots, 2^{10}\}$ and $\lambda = \{2^{-1}, 2^{-2}, \ldots, 2^{-10}\}$.

5. **Results and Discussion.** Since all the considered data sets are characterized by the skewed class distribution, we use $F_1$-measure to measure the classification performance. The average classification results with standard deviations in terms of the micro- and macro-$F_1$ over 10 trials are summarized in Tables 2 and 3, respectively. The bold font indicates the best performance. From these tables, we find that the von Neumann 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

TABLE 2. Micro-averaged $F_1$ values of different methods on four data sets

| Data set | Micro-$F_1$ (%) | | |
|---|---|---|---|
| | BOW kernel | Von Neumann kernel | Proposed method |
| interest | 85.29±1.02 | 87.12±0.98 | **88.19±0.27** |
| line | 82.14±0.57 | 83.27±0.53 | **84.36±0.94** |
| hard | 82.83±0.91 | 83.98±0.86 | **85.00±0.62** |
| serve | 85.36±1.28 | 86.10±1.24 | **87.46±0.33** |

TABLE 3. Macro-averaged $F_1$ values of different methods on four data sets

| Data set | Macro-$F_1$ (%) | | |
|---|---|---|---|
| | BOW kernel | Von Neumann kernel | Proposed method |
| interest | 63.23±1.08 | 70.58±0.62 | **73.36±0.31** |
| line | 74.12±0.73 | 75.46±1.23 | **77.42±0.92** |
| hard | 32.16±1.52 | 33.51±0.76 | **34.93±0.27** |
| serve | 54.07±0.89 | 55.96±0.48 | **58.31±0.64** |

performance. More importantly, for all data sets we see that the proposed approach achieves significant performance improvement over the von Neumann kernel. Take the *interest* data set for example: the proposed approach achieves the micro- and macro-$F_1$ values of 88.19% and 73.76% whereas the von Neumann kernel achieves those of 87.12% and 70.58%, respectively. In other words, the proposed approach achieves the micro- and macro-$F_1$ values with relative improvements of 1.23% ((88.19-87.12)/87.12) and 4.50% ((73.76-70.58)/70.58) over the von Neumann kernel, respectively. It should be noted that the performance differences are statistically significant ($p > 0.05$) in light of the pairs t-tests on all four data sets. Since whether or not the unlabeled data $U$ is taken into consideration is the only difference between the proposed approach and the von Neumann kernel, these results imply that the unlabeled data has a conspicuous impact on the kernel construction for WSD and demonstrate the effectiveness of the proposed approach with application to WSD.

Finally, it is also worth noting that, due to the severely skewed class distribution of the data sets, for all methods the micro-averaged $F_1$ values are consistently higher than the macro-averaged $F_1$ values. Conceptually, the micro-averaged $F_1$ will not be affected by the small classes since it gives an equal weight to all instances. On the contrary, the macro-averaged $F_1$ is an average over all the classes so the small classes will drastically affect the value.

6. **Conclusions.** We have presented a novel von Neumann kernel based semi-supervised WSD 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 von Neumann 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 WSD 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 automatic WSD.

**Acknowledgment.** This work is partially supported by the Science and Technology Program Foundation of Jiangxi Education Committee of China (No. GJJ151019) and the Natural Science Foundation of Jiangxi Province of China (No. 20161BAB202070). The author also gratefully acknowledges the helpful comments and suggestions of the reviewers, which have improved the presentation.

## REFERENCES


[1] R. Navigli, Word sense disambiguation: A survey, *ACM Computing Surveys*, vol.41, no.2, pp.1-69, 2009.

[2] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, UK, 2004.

[3] Y. K. Lee and H. T. Ng, An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, USA, pp.41-48, 2002.

[4] Y. K. Lee, H. T. Ng and T. K. Chia, Supervised word sense disambiguation with support vector machines and multiple knowledge sources, *Proc. of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, pp.137-140, 2004.

[5] M. Popescu, Regularized least-squares classification for word sense disambiguation, *Proc. of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, pp.209-212, 2004.

[6] A. Gliozzo, C. Giuliano and C. Strapparava, Domain kernels for word sense disambiguation, *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics*, University of Michigan, USA, pp.403-410, 2005.

[7] M. Joshi, T. Pedersen, R. Maclin and S. Pakhomov, Kernel methods for word sense disambiguation and acronym expansion, *Proc. of the 21st National Conference on Artificial Intelligence*, Boston, USA, 2006.

[8] C. Giuliano, A. Gliozzo and C. Strapparava, Kernel methods for minimally supervised WSD, *Computational Linguistics*, vol.35, no.4, pp.513-528, 2009.

[9] T. Pahikkala, S. Pyysalo, J. Boberg, J. Järvinen and T. Salakoski, Matrix representations, linear transformations, and kernels for disambiguation in natural language, *Machine Learning*, vol.74, no.2, pp.133-158, 2009.

[10] T. Wang, J. Rao and D. Zhao, Using exponential kernel for word sense disambiguation, *Proc. of the 23rd International Conference on Artificial Neural Networks*, Sofia, Bulgaria, pp.545-552, 2013.

[11] T. Wang, J. Rao and Q. Hu, Supervised word sense disambiguation using semantic diffusion kernel, *Engineering Applications of Artificial Intelligence*, vol.27, pp.167-174, 2014.

[12] T. Wang, J. Zhong, J. Chen and Q. Hu, Composite kernels for automatic word sense disambiguation, *Journal of Computational and Theoretical Nanoscience*, vol.12, no.4, pp.619-623, 2015.

[13] X. Li, S. Qing, H. Zhang, T. Wang and H. Yang, Kernel methods for word sense disambiguation, *Artificial Intelligence Review*, vol.46, no.1, pp.41-58, 2016.

[14] J. Kandola, J. Shawe-Taylor and N. Cristianini, Learning semantic similarity, *Advances in Neural Information Processing Systems*, vol.15, pp.657-664, 2003.

[15] R. I. Kondor and J. Lafferty, Diffusion kernels on graphs and other discrete structures, *Proc. of the 19th International Conference on Machine Learning*, Sydney, Australia, pp.315-322, 2002.

[16] C. W. Hsu and C. J. Lin, A comparison of methods for multiclass support vector machines, *IEEE Trans. Neural Networks*, vol.13, no.2, pp.415-425, 2002.

[17] T. Wang, D. Zhao and Y. Feng, Two-stage multiple kernel learning with multiclass kernel polarization, *Knowledge-Based Systems*, vol.48, pp.10-16, 2013.

[18] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang and C. J. Lin, LIBLINEAR: A library for large linear classification, *Journal of Machine Learning Research*, vol.9, pp.1871-1874, 2008.