Semantic Classification with Distributional Kernels

Diarmuid Ó Séaghdha
Computer Laboratory
University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
do242@cl.cam.ac.uk

Ann Copestake
Computer Laboratory
University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
aac10@cl.cam.ac.uk

Abstract

Distributional measures of lexical similarity and kernel methods for classification are well-known tools in Natural Language Processing. We bring these two methods together by introducing *distributional kernels* that compare co-occurrence probability distributions. We demonstrate the effectiveness of these kernels by presenting state-of-the-art results on datasets for three semantic classification: compound noun interpretation, identification of semantic relations between nominals and semantic classification of verbs. Finally, we consider explanations for the impressive performance of distributional kernels and sketch some promising generalisations.

1 Introduction

This paper draws a connection between two well-known topics in statistical Natural Language Processing: distributional measures of lexical similarity and kernel methods for classification. Distributional similarity measures quantify the similarity between pairs of words through their observed co-occurrences with other words in corpus data. The kernel functions used in support vector machine classifiers also allow an interpretation as similarity measures; however, not all similarity measures can be used as kernels. In particular, kernel functions must satisfy the mathematical property of *positive semi-definiteness*. In Section 2 we consider kernel functions suitable for comparing co-occurrence probability distributions and show that these kernels are closely related to measures known from the distributional similarity literature. We apply these *distributional kernels* to three semantic classification tasks: compound noun interpretation, identification of semantic relations between nominals and semantic classification of verbs. In all cases, the distributional kernels outperform the linear and Gaussian kernels standardly used for SVM classification and furthermore achieve state-of-the-art results. In Section 4 we provide a concrete explanation for the superior performance of distributional kernels, and in Section 5 we outline some promising directions for future research.

2 Theory

2.1 Distributional Similarity Measures

Distributional approaches to lexical similarity assume that words appearing in similar contexts are likely to have similar or related meanings. To measure distributional similarity, we use a representation of words based on observation of their relations with other words. Specifically, a target word w is represented in terms of a set C of admissible co-occurrence types $c = (r, w')$, where the word w' belongs to a co-occurrence vocabulary V_c and r is a relation that holds between w and w'. Co-occurrence relations may be syntactic (e.g., verb-argument, conjunct-conjunct) or may simply be one of proximity in text. Counts $f(w, c)$ of a target word w's co-occurrences can be estimated from language corpora, and these counts can be weighted in a variety of ways to reflect prior knowledge or to reduce statistical noise. A simple weighting method is to represent each word w as a vector of co-occurrence probabilities $(P(c_1|w), \dots, P(c_{|C|}|w))$. This vector defines the parameters of a categorical or multinomial probability distribution, giving a useful probabilistic interpretation of the distributional model. As the vector for each target word must sum to 1, the marginal distributions of target words have little effect on the resulting similarity estimates. Many

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 649–656
Manchester, August 2008

similarity measures and weighting functions have been proposed for distributional vectors; comparative studies include Lee (1999), Curran (2003) and Weeds and Weir (2005).

2.2 Kernel Methods for Computing Similarity and Distance

In this section we describe two classes of functions, positive semi-definite and negative semi-definite kernels, and state some relationships between these classes. The mathematical treatment follows Berg et al. (1984). A good general introduction to kernels and support vector machines is the book by Cristianini and Shawe-Taylor (2000).

Let $\mathcal{X}$ be a set of items and let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a symmetric real-valued function on pairs of items in $\mathcal{X}$. Then k is a *positive semi-definite (psd) kernel* if for all finite n-element sets $X \subseteq \mathcal{X}$, the $n \times n$ *Gram matrix* K defined by $K_{ij} = k(x_i, x_j)$ satisfies the property

$$v'Kv \geq 0,\ \forall v \in \mathbb{R}^n \tag{1}$$

This is equivalent to requiring that k define an inner product in a Hilbert space $\mathcal{F}$ which may be the same as $\mathcal{X}$ or may differ in dimensionality or in type ($\mathcal{F}$ is by definition a vector space, but $\mathcal{X}$ need not be). An intuitive interpretation of psd kernels is that they provide a similarity measure on members of $\mathcal{X}$ based on an embedding ϕ from input space $\mathcal{X}$ into feature space $\mathcal{F}$. It can be shown that a function is psd if and only if all Gram matrices K have no negative eigenvalues.

Kernel functions have received significant attention in recent years through their applications in machine learning, most notably support vector machines (SVMs, Cortes and Vapnik (1995)). SVM classifiers learn a decision boundary between two data classes that maximises the minimum distance or *margin* from the training points in each class to the boundary. The notion of distance used and the feature space in which the boundary is set are determined by the choice of kernel function. So long as the kernel satisfies (1), the SVM optimisation algorithm is guaranteed to converge to a global optimum that affords the geometric interpretation of margin maximisation. Besides these desirable optimisation properties, kernel methods have the advantage that the choice of kernel can be based on prior knowledge about the problem and on the nature of the data.

A *negative semi-definite (nsd) kernel* is a symmetric function $\tilde{k} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that for all finite n-element sets $X \subseteq \mathcal{X}$ and for all vectors $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$ with $\sum_i v_i = 0$

$$v'\tilde{K}v \leq 0 \tag{2}$$

Whereas positive semi-definite kernels correspond to inner products in a Hilbert space $\mathcal{F}$, negative semi-definite kernels correspond to squared distances. In particular, if $\tilde{k}(x, y) = 0$ only when $x = y$ then $\sqrt{\tilde{k}}$ is a metric. If a function k is psd, then $-k$ is always nsd, but the converse does not hold.[1] However, Berg et al. (1984) describe two simple methods for inducing a positive semi-definite function k from negative semi-definite $\tilde{k}$:

$$\begin{aligned} k(x,y) = \tilde{k}(x,x_0) + \tilde{k}(y,x_0) - \tilde{k}(x,y) \\ - \tilde{k}(x_0,x_0),\ \forall x_0 \in \mathcal{X} \end{aligned} \tag{3a}$$

$$k(x,y) = \exp(-\alpha\tilde{k}(x,y)),\ \forall \alpha > 0 \tag{3b}$$

The point x_0 in (3a) can be viewed as providing an origin in $\mathcal{F}$ that is the image of some point in the input space $\mathcal{X}$; the choice of x_0 does not have an effect on SVM classification. A familiar example of these transformations arises if we take $\tilde{k}$ to be the squared Euclidean L_2 distance $\|x - y\|^2 = \sum_i (x_i - y_i)^2$. Applying (3a) and setting x_0 to be the zero vector, we obtain a quantity that is twice the linear kernel $k(x, y) = \sum_i x_i y_i$. Applying (3b) we derive the Gaussian kernel $k(x, y) = \exp(-\alpha\|x - y\|^2)$. In the next section we consider kernels obtained by plugging alternative squared metrics into equations (3a) and (3b).

2.3 Distributional Kernels

Given the effectiveness of distributional similarity measures for numerous tasks in NLP and the interpretation of kernels as similarity functions, it seems natural to consider the use of kernels tailored for co-occurrence distributions when performing semantic classification. As shown in Section 2.2 the standardly used linear and Gaussian kernels derive from the L_2 distance, yet Lee (1999) has shown that this distance measure is relatively poor at comparing co-occurrence distributions. Information theory provides a number of alternative distance functions on probability measures, of which the L_1 distance (also called *variational distance*), Kullback-Leibler divergence and Jensen-Shannon divergence are well-known in NLP and

[1] Negated nsd functions are sometimes called *conditionally psd*; they constitute a superset of the psd functions.

Distance	Definition	Derived linear kernel
$(L_2 \text{ distance})^2$	$\sum_c (P(c\vert w_1) - P(c\vert w_2))^2$	$\sum_c P(c\vert w_1)P(c\vert w_2)$
L_1 distance	$\sum_c \lvert P(c\vert w_1) - P(c\vert w_2)\rvert$	$\sum_c \min(P(c\vert w_1), P(c\vert w_2))$
Jensen-Shannon divergence	$\sum_c P(c\vert w_1)\log_2(\frac{2P(c\vert w_1)}{P(c\vert w_1)+P(c\vert w_2)}) + P(c\vert w_2)\log_2(\frac{2P(c\vert w_2)}{P(c\vert w_1)+P(c\vert w_2)})$	$-\sum_c P(c\vert w_1)\log_2(\frac{P(c\vert w_1)}{P(c\vert w_1)+P(c\vert w_2)}) + P(c\vert w_2)\log_2(\frac{P(c\vert w_2)}{P(c\vert w_1)+P(c\vert w_2)})$
Hellinger distance	$\sum_c (\sqrt{P(c\vert w_1)} - \sqrt{P(c\vert w_2)})^2$	$\sum_c \sqrt{P(c\vert w_1)P(c\vert w_2)}$

Table 1: Squared metric distances on co-occurrence distributions and corresponding linear kernels

were shown by Lee to give better similarity estimates than the L_2 distance.

In Section 2.2 we have seen how to derive psd kernels (similarities) from nsd kernels (distances). It seems likely that distance measures that are known to work well for comparing co-occurrence distributions will also give us suitable psd similarity measures. Negative semi-definite kernels are by definition symmetric, which rules the Kullback-Leibler divergence and Lee's (1999) α-skew divergence out of consideration. The nsd condition (2) is met if the distance function is a squared metric in a Hilbert space. In this paper we use a parametric family of squared Hilbertian metrics on probability distributions that has been discussed by Hein and Bousquet (2005). This family contains many familiar distances including the L_1 distance, Jensen-Shannon divergence (*JSD*) and the Hellinger distance used in statistics, though not the squared L_2 distance. Positive semi-definite *distributional kernels* can be derived from these distances through equations (3a) and (3b). We interpret the distributional kernels produced by (3a) and (3b) as analogues of the linear and Gaussian kernels respectively, given by a different norm or concept of distance in the feature space $\mathcal{F}$. Hence the *linear* distributional kernels produced by (3a) correspond to inner products in the input space $\mathcal{X}$, and the *rbf* distributional kernels produced by (3b) are radial basis functions corresponding to inner products in a high-dimensional Hilbert space of Gaussian-like functions. In this paper we use the unmodified term "linear kernel" in the standard sense of the linear kernel derived from the L_2 distance and make explicit the related distance when referring to other linear kernels, e.g., the "JSD linear kernel". Likewise, we use the standard term "Gaussian" to refer to the L_2 rbf kernel, and denote other rbf kernels as, for example, the "JSD rbf kernel".

Table 1 lists relevant squared metric distances and their derived linear kernels. The linear kernel derived from the L_1 distance is the same as the *difference-weighted token-based similarity* measure of Weeds and Weir (2005). The JSD linear kernel can be rewritten as (2 - *JSD*), where JSD is the value of the Jensen-Shannon divergence. This formulation is used as a similarity measure by Lin (1999). Dagan et al. (1999) use a similarity measure $10^{-\alpha JSD}$, though they acknowledge that this transformation is heuristically motivated. The rbf kernel $\exp(-\alpha JSD)$ provides a theoretically sound alternative when the psd property is required. It follows from the above discussion that these previously known distributional similarity measures are valid kernel functions and can be used directly for SVM classification.

Finally, we consider the status of other popular distributional measures. The familiar cosine similarity measure is provably a valid psd kernel, as it is the L_2 linear kernel calculated between L_2-normalised vectors. Distributional vectors are by definition L_1-normalised (they sum to 1), but there is evidence that L_2 normalisation is optimal when using L_2 kernels for tasks such as text categorisation (Leopold and Kindermann, 2002). Indeed, in the experiments described below L_2-normalised feature vectors are used with the L_2 kernels, and the L_2 linear kernel function then becomes identical to the cosine similarity. Other similarity measures, such as that of Lin (1998), can be shown to be non-psd by calculating similarity matrices from real or artificial data and showing that their nonzero eigenvalues are not all positive, as is required by psd functions.

3 Practice

3.1 General Methodology

All experiments were performed using the LIBSVM Support Vector Machine library (Chang and Lin, 2001), modified to implement one-against-all classification. The members of the distributional kernel family all performed similarly but the Jensen-Shannon divergence kernels gave the most consistently impressive results, and we restrict discussion to these kernels due to considerations of space and clarity. In each experiment we compare the standard linear and Gaussian kernels with the linear and JSD rbf kernels. As a preprocessing step for the L_2 kernels, each feature vector was normalised to have unit L_2 norm. For the Jensen-Shannon kernels, the feature vectors were normalised to have unit L_1 norm, i.e., to define a probability distribution. For all datasets and all training-test splits the SVM cost parameter C was optimised in the range $(2^{-6}, 2^{-4}, \dots, 2^{12})$ through cross-validation on the training set. In addition, the width parameter α was optimised in the same way for the rbf kernels. The number of optimisation folds differed according to the size of the dataset and the number of training-test splits to be evaluated: we used 10 folds for the compound task, leave-one-out cross-validation for SemEval Task 4 and 25 folds for the verb classification task.

3.2 Compound Noun Interpretation

The task of interpreting the semantics of noun compounds is one which has recently received considerable attention (Lauer, 1995; Girju et al., 2005; Turney, 2006). For a given noun-noun compound, the problem is to identify the semantic relation between the compound's constituents – that a *kitchen knife* is a *knife used in a kitchen* but a *steel knife* is a *knife made of steel*.[2] The difficulty of the task is due to the fact that the knowledge required to interpret compounds is not made explicit in the contexts where they appear, and hence standard context-based methods for classifying semantic relations in text cannot be applied. Most previous work making use of lexical similarity has been based on WordNet measures (Kim and Baldwin, 2005; Girju et al., 2005). Ó Séaghdha and Copestake (2007) were to our knowledge the first to apply a distributional model. Here we build on their methodology by introducing a probabilistic feature weighting scheme and applying the new distributional kernels.

For our experiments we used the dataset of Ó Séaghdha and Copestake (2007), which consists of 1443 noun compounds annotated with six semantic relations: *BE*, *HAVE*, *IN*, *AGENT*, *INSTRUMENT* and *ABOUT*.[3] The classification baseline associated with always choosing the most frequent relation (*IN*) is 21.3%. For each compound (N_1, N_2) in the dataset, we associate the co-occurrence probability vector $(P(c_1|N_1), \dots, P(c_{|C|}|N_1))$ with N_1 and the vector $(P(c_1|N_2), \dots, P(c_{|C|}|N_2))$ with N_2. The probability vector for the compound is created by appending the two constituent vectors, each scaled by 0.5 to weight both constituents equally and ensure that the new vector sums to 1. These probability vectors are used to compute the Jensen-Shannon kernel values. The preprocessing step for the L_2 kernels is analogous, except that the co-occurrence *frequency* vector $(f(c_1, N_i), \dots, f(c_{|C|}, N_i))$ for each constituent N_i is normalised to have unit L_2 norm (instead of unit L_1 norm); the combined feature vector for each data item is also L_2-normalised.[4]

The co-occurrence relation we counted to estimate the probability vectors was the conjunction relation. This relation gives sparse but high-quality information, and was shown to be effective by Ó Séaghdha and Copestake. We extracted two feature sets from two very different corpora. The first is the 90 million word written component of the British National Corpus (Burnard, 1995). This corpus was parsed with the RASP parser (Briscoe et al., 2006) and all instances of the `conj` grammatical relation were counted. The co-occurrence vocabulary V_c was set to the 10,000 words most frequently entering into a `conj` relation across the corpus. The second corpus we used was the Web 1T 5-Gram Corpus (Brants and Franz, 2006), which contains frequency counts for n-grams up to length 5 extracted from Google's index of approximately 1 trillion words of Web text. As the nature of this corpus precludes parsing, we used a simple pattern-based technique to extract conjunctions. An n-gram was judged to contain a conjunction co-occurrence between N_i and N_j if it con-

[2] In the classification scheme considered here, *kitchen knife* would have the label *IN* and *steel knife* would be labelled *BE*.

[3] This dataset is available from `http://www.cl.cam.ac.uk/~do242/resources.html`.

[4] The importance of performing both normalisation steps was suggested to us by an anonymous reviewer's comments.

	BNC		5-Gram	
Kernel	Acc	F	Acc	F
Linear	57.9	55.8	55.0	52.5
Gaussian	58.0	56.2	53.5	50.8
JSD (linear)	**59.9**	57.8	60.2	58.1
JSD (rbf)	59.8	**57.9**	**61.0**	**58.8**

Table 2: Results for compound interpretation

	BNC		5-Gram	
Kernel	Acc	F	Acc	F
Linear	67.6	57.1	65.4	63.3
Gaussian	66.8	60.7	65.6	62.9
JSD (linear)	**71.4**	**68.8**	69.6	65.8
JSD (rbf)	69.9	66.7	**70.7**	**67.5**

Table 3: Results for SemEval Task 4

tained the pattern N_i *and* (¬N)* N_j (¬N)*. A noun dictionary automatically constructed from WordNet and an electronic version of Webster's 1913 Unabridged Dictionary determined the sets of admissible nouns {N} and non-nouns {¬N}.[5] The vocabulary V_c was again set to the 10,000 most frequent conjuncts, and the probability estimates $P(c|w)$ were based on the n-gram frequencies for each n-gram matching the extraction pattern. A third feature set extracted from the 5-Gram Corpus by using a larger set of joining terms was also studied but the results were not significantly different from the sparser conjunction feature sets and are not presented here.

Performance was measured by splitting the data into five folds and performing cross-validation. Results for the two feature sets and four kernels are presented in Table 2. The kernels derived from the Jensen-Shannon divergence clearly outperform the L_2 distance-based linear and Gaussian kernels in both accuracy and macro-averaged F-score. The best performing kernel-feature combination is the Jensen-Shannon rbf kernel with the 5-Gram features, which attains 61.0% accuracy and 58.8% F-score. This surpasses the best previous result of 57.1% accuracy, 55.3% F-score that was reported by Ó Séaghdha and Copestake (2007) for this dataset. That result was obtained by combining a distributional model with a relational similarity model based on string kernels; incorporating relational similarity into the system described here improves performance even further (Ó Séaghdha, 2008).

3.3 SemEval Task 4

Task 4 at the 2007 SemEval competition (Girju et al., 2007) focused on the identification of semantic relations among nominals in text. Identification of each of seven relations was designed as a binary classification task with 140 training sentences and around 70 test sentences.[6] To ensure that the task be a challenging one, the negative test examples were all "near misses" in that they were plausible candidates for the relation to hold but failed to meet one of the criteria for that relation. This was achieved by selecting both positive and negative examples from the results of the same targeted Google queries. The majority-class baseline for this task gives Accuracy = 57.0%, F-score = 30.8%, while the all-true baseline (label every test sentence positive) gives Accuracy = 48.5%, F-score = 64.8%.

We used the same feature sets and kernels as in Section 3.2. The results are presented in Table 3. Again, the JSD kernels outperform the standard L_2 kernels by a considerable margin. The best performing feature-kernel combination achieves 71.4% Accuracy and 68.8% F-score, higher than the best performance attained in the SemEval competition without using WordNet similarity measures (Accuracy = 67.0%, F-score = 65.1%; Nakov and Hearst (2007)). This is also higher than the performance of all but three of the 14 SemEval entries which did use WordNet. Davidov and Rappoport (2008) have recently described a WordNet-free method that attains slightly lower accuracy (70.1%) and slightly higher F-score (70.6%) than our method. Taken together, Davidov and Rappoport's results and ours define the current state of the art on this task.

3.4 Verb Classification

To investigate the effectiveness of distributional kernels on a different kind of semantic classification task, we tested our methods on the verb class data of Sun et al. (2008). This dataset consists of 204 verbs assigned to 17 of Levin's (1993) verb classes. Each verb is represented by a set of features corresponding to the distribution of its instances across subcategorisation frames (SCFs).

[5] The electronic version of Webster's is available from `http://msowww.anu.edu.au/~ralph/OPTED/`.

[6] The relations are *Cause-Effect*, *Instrument-Agency*, *Product-Producer*, *Origin-Entity*, *Theme-Tool*, *Part-Whole* and *Content-Container*.

	FS3		FS5	
Kernel	Acc	F	Acc	F
Linear	67.1	65.5	67.6	65.9
Gaussian	60.8	58.6	62.7	60.2
JSD (linear)	**70.6**	**67.3**	69.6	66.4
JSD (rbf)	68.6	65.1	**70.1**	**67.2**
Sun et al. (SVM)	57.8	58.2	57.3	57.4
Sun et al. (GS)	59.3	57.1	64.2	62.5

Table 4: Results for leave-one-out verb classification and comparison with Sun et al.'s (2008) SVM and Gaussian fitting methods

These frames include information about syntactic constituents (*NP*, *NP_NP*, *NP_SCOMP*, ...) and some lexical information about subcategorised prepositions (*NP_with*, *out*, ...). The feature values are counts of SCFs extracted from a large corpus. As the feature vector for each verb naturally defines a probability distribution over SCFs, it seems intuitive to apply distributional kernels to the problem of predicting Levin classes for verbs.

Sun et al. use multiple feature sets of varying sparsity and noisiness. We report results on the two feature sets for which they reported best performance; for continuity we keep the names *FS3* and *FS5* for these feature sets. These were derived from the least filtered and hence least sparse subcategorisation lexicon (which they call VALEX 1) and differ in the granularity of prepositional SCFs. The SCF representation in FS5 is richer and hence potentially more discriminative, but it is also sparser. Using an SVM with a Gaussian kernel, Sun et al. achieved their best results on FS3. Perhaps surprisingly, their best results overall were attained with FS5 by a simple method based on fitting multivariate Gaussian distributions to each class in the training data and assigning the maximum likelihood class to test points.

Following Sun et al., we use a leave-one-out measure of verb classification performance. As the examples are distributed equally across the 17 classes, the random baseline accuracy is 5.9$\%$. Table 4 presents our results with L_2 and JSD kernels, as well as those of Sun et al. The best overall performance is attained by the JSD linear kernel, which scores higher than the L_2-derived kernels on both feature sets. The L_2 linear kernel also performs quite well and with consistency. The JSD rbf kernel was less consistent over cross-validation runs, seemingly due to uncertainty in selecting the optimal α parameter value; it clearly outperforms the L_2 linear kernel on one feature set (FS5) but on the other (FS3) it attains a slightly lower F-score while maintaining a higher accuracy. The Gaussian kernel seems particularly ill-suited to this dataset, performing significantly worse than the other kernels. The difference between Sun et al.'s results with the Gaussian kernel and ours with the same kernel may be due to the use of one-against-all classification here instead of one-against-one, or it may be due to differences in preprocessing or parameter optimisation.

4 The effect of marginal distributions

It is natural to ask why distributional kernels perform better than the standard linear and Gaussian kernels. One answer might be that just as information theory provides the "correct" notion of information for many purposes, it also provides the "correct" notion of distance between probability distributions. Hein and Bousquet (2005) show that their family of distributional kernels are invariant to bijective transformations of the event space C and suggest that this property is a valuable one for image histogram classification where data may be represented in a range of equivalent colour spaces. However, it is not clear that this confers an advantage when comparing lexical co-occurrence distributions; when transformations are performed on the space of co-occurrence types, they are generally not information-conserving, for example lemmatisation or stemming.

A more practical explanation is that the distributional kernels and distances are less sensitive than the (squared) L_2 distance and its derived kernels to the marginal frequencies of co-occurrence types. When a type c has high frequency we expect that it will have higher variance, i.e., the differences $|P(c|w_1) - P(c|w_2)|$ will tend to be greater even if c is not a more important signifier of similarity.[7] These differences contribute quadratically to the L_2 distance and hence also to the associated rbf kernel, i.e., the Gaussian kernel. It is also easy to see that types c for which $P(c|w_i)$ tends to be large will dominate the value of the linear kernel. This explanation is also plausibly a factor in the relatively poor performance of L_2 distance as a lexical dissimilarity measure, as demon-

[7] Chapelle et al. (1999) give a similar explanation for the performance of a related family of kernels on a histogram classification task.

strated by Lee (1999). In contrast, the differences $|P(c|w_1) - P(c|w_2)|$ are not squared in the L_1 distance formula, and the minimum function in the L_1 linear kernel dampens the effect of high-variance co-occurrence types. The Jensen-Shannon formula is more difficult to interpret, as the difference terms do not directly appear. While co-occurrence types with large $P(c|w_1)$ and $P(c|w_2)$ do contribute more to the distance and kernel values, it is the proportional size of the difference that appears in the log term rather than its magnitude. Finally, the Hellinger distance and kernels squash the variance associated with c through the square root function.

5 Discussion and Future Directions

Kernels on probability measures have been discussed in the machine learning literature (Kondor and Jebara, 2003; Cuturi et al., 2005; Hein and Bousquet, 2005), but they have previously been applied only to standard image and text classification benchmark tasks. We seem to be the first to use distributional kernels for semantic classification and to note their connection with familiar lexical similarity measures. Indeed, the only research we are aware of on kernels tailored for lexical similarity is the small body of work on WordNet kernels, e.g., Basili et al. (2006). In contrast, Support Vector Machines have been widely adopted for computational semantic tasks, from word sense disambiguation (Gliozzo et al., 2005) to semantic role labelling (Pradhan et al., 2004). The standard feature sets for semantic role labelling and many other tasks are collections of heterogeneous features that do not correspond to probability distributions. So long as the features are restricted to positive values, distributional kernels can be applied; it will be interesting (and informative) to see whether they retain their superiority in this setting.

One advantage of kernel methods is that kernels can be defined for non-vectorial data structures such as strings, trees, graphs and sets. A promising topic of future research is the design of distributional kernels for comparing structured objects, based on the feature space embedding associated with convolution kernels (Haussler, 1999). These kernels map structures in $\mathcal{X}$ into a space whose dimensions correspond to substructures of the elements of $\mathcal{X}$. Thus strings are mapped onto vectors of substring counts, and trees are mapped onto vectors of subtree counts. We adopt the perspective that this mapping represents structures $x_i \in \mathcal{X}$ as measures over substructures $\bar{x}_1, \ldots, \bar{x}_d$. Properly normalised, this gives a distributional probability vector $(P(\bar{x}_1), \ldots, P(\bar{x}_d))$ similar to those used for computing lexical similarity. This perspective motivates the use of distributional inner products instead of the dot products implicitly used in standard convolution kernels. Several authors have suggested applying distributional similarity measures to sentences and phrases for tasks such as question answering (Lin and Pantel, 2001; Weeds et al., 2005). Distributional kernels on strings and trees should provide a flexible implementation of these suggestions that is compatible with SVM classification and does not require manual feature engineering. Furthermore, there is a ready generalisation to kernels on sets of structures; if a set is represented as the normalised sum of its member embeddings in feature space $\mathcal{F}$, distributional methods can be applied directly.

6 Conclusion

In this paper we have introduced distributional kernels for classification with co-occurrence probability distributions. The suitability of distributional kernels for semantic classification is intuitive, given their relation to proven distributional methods for computing semantic similarity, and in practice they work very well. As these kernels give state-of-the-art results on the three datasets we have tested, we expect that they will prove useful for a wide range of semantic classification problems in future.

Acknowledgements

We are grateful to Andreas Vlachos and three anonymous reviewers for their useful comments, and to Anna Korhonen for providing the verb classification dataset. This work was supported in part by EPSRC Grant EP/C010035/1.

References

Basili, Roberto, Marco Cammisa, and Alessandro Moschitti. 2006. A semantic kernel to classify texts with very few training examples. *Informatica*, 30(2):163–172.

Berg, Christian, Jens P. R. Christensen, and Paul Ressel. 1984. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*. Springer, Berlin.

Brants, Thorsten and Alex Franz, 2006. *Web 1T 5-gram Corpus Version 1.1*. Linguistic Data Consortium.

Briscoe, Ted, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the ACL Interactive Presentation Sessions*.

Burnard, Lou, 1995. *Users' Guide for the British National Corpus*. British National Corpus Consortium, Oxford University Computing Service.

Chang, Chih-Chung and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chapelle, Olivier, Patrick Haffner, and Vladimir N. Vapnik. 1999. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064.

Cortes, Corinna and Vladimir Vapnik. 1995. Support vector networks. *Machine Learning*, 20(3):273–297.

Cristianini, Nello and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge.

Curran, James. 2003. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Cuturi, Marco, Kenji Fukumizu, and Jean-Philippe Vert. 2005. Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:1169–1198.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1–4):43–69.

Davidov, Dmitry and Ari Rappoport. 2008. Classification of semantic relationships between nominals using pattern clusters. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.

Girju, Roxana, Dan Moldovan, Marta Tatu, and Daniel Antohe. 2005. On the semantics of noun compounds. *Computer Speech and Language*, 19(4):479–496.

Girju, Roxana, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 Task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations*.

Gliozzo, Alfio, Claudio Giuliano, and Carlo Strapparava. 2005. Domain kernels for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Haussler, David. 1999. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, Computer Science Department, University of California at Santa Cruz.

Hein, Matthias and Olivier Bousquet. 2005. Hilbertian metrics and positive definite kernels on probability measures. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*.

Kim, Su Nam and Timothy Baldwin. 2005. Automatic interpretation of noun compounds using WordNet similarity. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing*.

Kondor, Risi and Tony Jebara. 2003. A kernel between sets of vectors. In *Proceedings of the 20th International Conference on Machine Learning*.

Lauer, Mark. 1995. *Designing Statistical Language Learners: Experiments on Compound Nouns*. Ph.D. thesis, Macquarie University.

Lee, Lillian. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

Leopold, Edda and Jörg Kindermann. 2002. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46(1–3):423–444.

Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago.

Lin, Dekang and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Lin, Dekang. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*.

Nakov, Preslav I. and Marti A. Hearst. 2007. UCB: System description for SemEval Task #4. In *Proceedings of the 4th International Workshop on Semantic Evaluations*.

Ó Séaghdha, Diarmuid and Ann Copestake. 2007. Co-occurrence contexts for noun compound interpretation. In *Proceedings of the ACL Workshop on a Broader Perspective on Multiword Expressions*.

Ó Séaghdha, Diarmuid. 2008. *Learning Compound Noun Semantics*. Ph.D. thesis, Computer Laboratory, University of Cambridge. In preparation.

Pradhan, Sameer, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2004. Support vector learning for semantic argument classification. *Machine Learning*, 60(1):11–39.

Sun, Lin, Anna Korhonen, and Yuval Krymolowski. 2008. Verb class discovery from rich syntactic data. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics*.

Turney, Peter D. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Weeds, Julie and David Weir. 2005. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–476.

Weeds, Julie, David Weir, and Bill Keller. 2005. The distributional similarity of sub-parses. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*.