Compactly Supported Basis Functions as Support Vector Kernels for Classification

Peter Wittek, and Chew Lim Tan, *Senior Member, IEEE*

Abstract—Wavelet kernels have been introduced for both support vector regression and classification. Most of these wavelet kernels do not use the inner product of the embedding space, but use wavelets in a similar fashion to radial basis function kernels. Wavelet analysis is typically carried out on data with a temporal or spatial relation between consecutive data points. We argue that it is possible to order the features of a general data set so that consecutive features are statistically related to each other, thus enabling us to interpret the vector representation of an object as a series of equally or randomly spaced observations of a hypothetical continuous signal. By approximating the signal with compactly supported basis functions and employing the inner product of the embedding L_2 space, we gain a new family of wavelet kernels. Empirical results show a clear advantage in favor of these kernels.

Index Terms—Wavelet kernels, feature engineering, feature correlation, semantic kernels

1 Introduction

A mesmerizing array of feature selection methods has been developed and thoroughly benchmarked, emphasizing the importance of a good feature set (see, for instance, [1] for a recent overview). Among other reasons, one motivation to remove features or to reduce their weights is to deal with the curse of dimensionality, which, in many cases, causes increased computational needs. However, with the advent of kernel methods [2], even infinite dimensional feature spaces are tractable. Our research question is: Can we keep all the features and use feature interactions to our advantage?

Recently, wavelet kernels have been investigated in certain applications including regression [3], voice classification [4], and biomarker discovery in protein structures [5], bringing in the reach framework of wavelet analysis from signal processing. As one attempt with a Haar wavelet in information retrieval highlights [6], there is one fundamental problem with existing wavelet kernels when it comes to classification: in order to make such wavelet kernels operational, a relation (traditionally temporal or spatial) is assumed between subsequent features that describe the data instances. We argue that it is possible to establish a relation between subsequent features (not necessarily temporal or spatial), and thus wavelet kernels can be applied in any domain. Moreover, we also show that our proposed wavelet kernel also expands on the features and weights related features to improve both precision and recall in a classification scenario.

The key contribution of this paper is threefold:

1) A novel feature ordering algorithm is described and benchmarked against an existing method for ordination. The ordination is a necessary step for the proposed kernel.
2) A new family of wavelet kernels for classification is introduced, which can be applied on any data, irrespective of whether there is temporal or spatial relation between the features.
3) On text collections, the proposed compactly supported wavelet kernel functions provide semantic smoothing while also preserving computational efficiency.

With regards to the above, the key results are

1) The kernel outperforms baseline methods, especially when the feature space is sparse.
2) Many parameter combinations can be ruled out, and the kernel is fairly robust to the rest of the combinations.

The rest of this paper is organized as follows. Section 2 discusses related work on support vector machines and wavelet kernels in particular. In Section 3, we identify the conditions of applicability and the properties we require from a kernel that enables the efficient incorporation of prior knowledge or knowledge of distributional properties of the feature set. The compactness of our representation is achieved by encoding the pairwise relatedness of features into a one-dimensional order of the features (Section 3.1). This order establishes a relation between consecutive features that is similar to a spatial or temporal relation in other application fields of wavelet analysis. Section 3.2 introduces nonorthonormal wavelet kernels with a compact support, able to utilize this order to improve classification performance. Section 4 presents experimental results on several benchmark databases, including general datasets with no prior knowledge and text collections where prior knowledge on feature dependence is widely available. Finally, Section 5 concludes the paper.

• *P. Wittek and C.L. Tan are with the School of Computing, National University of Singapore.*
E-mail: {wittek,tancl}@comp.nus.edu.sg .

2 Related Work

Categorization is the task of assigning unlabeled objects to predefined categories. A support vector machine (SVM) is a kind of supervised learning algorithm which learns a given independent and identically distributed training example set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x} \in \mathbf{R}^n$, $y \in \{-1, 1\}$, and n is the number of features. Kernel mapping maps the training examples from an input space into a so-called feature space in which the mapped training examples are linearly separable. SVMs require the solution of the following optimization problem:

$$\text{Minimize} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

subject to

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + a) \geq 1 - \xi_i, \quad \xi_i \geq 0,$$

where ξ_i is the allowed error term in soft-margin classification, and C is a penalty parameter to regulate the sum of the error terms, and $\phi(.)$ is an allowed mapping. With the above notation, we can write the decision function as $f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\phi(\mathbf{x}) + a)$.

In most practical application, the dual formulation of the optimization problem is solved, which can be written as follows:

$$\text{Maximize} \quad \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i y_i \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to

$$\sum_{i=1}^{l}\alpha_i y_i = 0, \quad \alpha_i \in [0, C], \quad i = 1, \ldots, l.$$

The decision function for a binary decision problem becomes $f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{l}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + a\right)$.

Since the mapping does not need to be calculated explicitly, the embedding space can be of infinite dimensions. A kernel can be thought of as an inner product of the embedding space:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)). \tag{1}$$

Moreover, any continuous symmetric function $K(\mathbf{x}_i, \mathbf{x}_j) \in L_2 \otimes L_2$ may be used as an admissible kernel, as long as satisfies a weak form of Mercer's condition [7]:

$$\int\int K(\mathbf{x}_i, \mathbf{x}_j)g(\mathbf{x}_i)g(\mathbf{x}_j) \geq 0 \quad \text{for all } g \in L_2(\mathbf{R}^n). \tag{2}$$

This latter approach gave rise to radial basis function kernels, and a recent work explored combining wavelets with SVMs proving that these kernels satisfy the above condition [3].

Wavelet analysis expresses or approximates a signal by a family of functions generated by dilations and translations of another function $\psi(x)$ called the mother wavelet [8]:

$$\psi_{\alpha,\beta}(x) = |\alpha|^{-1/2}\psi(\frac{x-\beta}{\alpha}) \quad \text{over the integers } \alpha, \beta,$$

where α and β denote the dilation and translation, respectively, and x is just a "dummy" variable unrelated to the data collection. The above functions are also referred to as child wavelets. The child wavelets form a basis for functions in L_2. If $\mathbf{x}_i = (x_{1i}, x_{2i}, \ldots, x_{ni}), \mathbf{x}_j = (x_{1j}, x_{2j}, \ldots, x_{nj}) \in \mathbf{R}^n$, then the wavelet kernels are [3]:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \prod_{k=1}^{n}\psi(\frac{x_{ki}-\beta}{\alpha})\psi(\frac{x_{kj}-\beta}{\alpha}). \tag{3}$$

The authors focused on the kernel defined in Equation 3 and mainly on support vector regression, but results confirmed earlier findings that wavelets can be very efficient in classification problems [9], [10]. Compactly supported wavelets have also been benchmarked for support vector regression [11].

In an unrelated attempt, Hoenkamp has identified another wavelet kernel for information retrieval [6]. Singular value decomposition (SVD) based rank reduction produces the closest rank k matrix to a given matrix. The advantage of SVD is that it removes noise from the data by dimension reduction, by discarding the smallest singular values of the decomposition. Hoenkamp introduced the Haar transform as an alternative to SVD to achieve dimensionality reduction with lower computational complexity. The Haar wavelet's mother wavelet function $\psi(x)$ can be described as a compactly supported basis function:

$$\psi(x) = \begin{cases} 1 & 0 \leq x < 1/2, \\ -1 & 1/2 \leq x < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Following Hoenkamp's example, the discrete version of the Haar transform relies on the following simple observation.: Any two numbers can be represented by their average, while the number that you add on one side, you subtract from the other. Haar transform can be viewed as a series of averaging and differencing operations on a discrete function. Figure 1 shows the construction for a document vector of four index terms: (2, 0, 3, 5). The numbers are taken two by two, and represented by their average (Figure 1.b) left side) and the coefficient of a Haar function (the two diagrams to the right in Figure 1.b)). The procedure is repeated until it ends in an overall average and a series of Haar coefficients.

The analog of SVD for the Haar transform is to ignore the smallest Haar coefficients to reduce noise. If we denote the operator which maps the space to a new space in which the smallest Haar coefficients are discarded by $\hat{\phi}$, then the underlying kernel is

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\hat{\phi}(\mathbf{x}_i), \hat{\phi}(\mathbf{x}_j)). \tag{4}$$

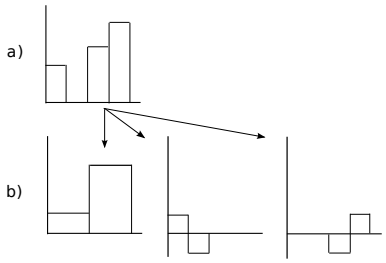


Fig. 1. The first step of Haar expansion for an object vector (2,0,3,5). (a) the vector as a function of x. (b) Each pair of terms is decomposed in its average and a suitably scaled Haar function.

In this regard, this wavelet kernel is different from the one defined by Equation 3, since the inner product of the embedding space is explicitly calculated as in Equation 1. A compactly supported wavelet kernel has been used in a similar fashion for regression [11]. While that kernel could also be used for classification, the authors used wavelets which are not non-negative. A non-negative family of kernels has certain advantages when considering the similarity between two instances, as it is shown below.

When calculating the average of a pair of features, it is assumed that there is some relation between the two features. For instance, when using similar wavelet kernels in image processing, the relationship is spatial [12], [5], and in time series analysis, the relationship is temporal [4], [13], [14], [15]. Hoenkamp introduced the Haar kernel for information retrieval, where the features are index terms of natural language documents, and relatedness is not guaranteed for two consecutive terms in the vector representation of a document. The assignment of canonical basis vectors to features is arbitrary in case of the classical vector space-based model. Let n denote the number of features. By the classical approach, one vector of the canonical basis $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ of $\mathbb{R}^l$ is assigned to each feature (note that l is the number of training instances). The assignment of features to vectors is arbitrary. Let x_{ij} be the weight of feature f_i in object x_j. Thus an object vector $\mathbf{x}_j$ is a linear combination of the canonical basis vectors.

$$\mathbf{x}_j = \sum_{i=1}^{n} x_{ij}\mathbf{e}_i, \quad 1 \leq j \leq l.$$

By writing $\mathbf{x}_j$ as a column vector, $\mathbf{x}'_j = (x_{1j}, x_{2j}, \ldots, x_{nj})$. Since the basis vectors of the canonical basis are perpendicular to one another, it implies that the features are mutually independent; an assumption that is rarely valid. One important thing to note is that the assignment of features to vectors is completely *arbitrary*, a feature can be assigned to any of the vectors of the canonical basis.

In what follows, we argue that it is possible to reorder the feature set in a way that consecutive features are related (Section 3.1), and then introduce kernels based on compactly supported basis functions that utilize this order (Section 3.2).

3 Our Framework

We wanted to design a kernel that meets the following criteria:

1) Allows the incorporation of prior knowledge or knowledge of the distributional properties of the feature set.
2) Does not have additional storage needs.
3) Its running time and computational complexity is close to that of a linear kernel.
4) Improves classification performance.

The first two criteria are addressed by an algorithm that reorders the index terms (Section 3.1). It is similar to a feature weighting algorithm, but instead of weights, it assigns positions to features such that consecutive features are related. Relatedness is defined by a distance function as defined later as required in Section 4. The representation remains sparse, since the new order of terms replaces the original (alphabetic or any arbitrary) order, and does not need an additional matrix or other object to store information.

The third criterion is addressed by a wavelet kernel (Section 3.2), which, with a support short enough, runs in the same time as a linear kernel. Section 4 presents empirical results that address the fourth criterion.

3.1 An Algorithm to Reorder the Feature Set

Let V denote a set of features $\{f_1, f_2, \ldots, f_n\}$ and let $d(f_i, f_j)$ denote the distance between the features f_i and f_j. The initial order of the features is not relevant.

Let $G = (V, E)$ denote a weighted undirected graph, where the weights on the set E are defined by the distances between the features. G is a K_n complete graph. Using such a graph is not a novelty. For instance, Ordering Points To Identify the Clustering Structure (OPTICS) uses this graph to create an order of instances [16]. OPTICS was derived from Density-Based Spatial Clustering of Applications with Noise (DBSCAN, [17]), which needs two input parameters, ϵ and MinPts, to define:

1) An ϵ-neighborhood $N_\epsilon(\mathbf{x}_i) = \{\mathbf{x}_j \in X | d(\mathbf{x}_i, \mathbf{x}_j) \leq \epsilon\}$ of the point $\mathbf{x}_i$, where X is the space of objects;
2) A core object (a point with a neighborhood consisting of more than a parameter MinPts points);
3) A concept of a point $\mathbf{x}_j$ density-reachable from a core object $\mathbf{x}_i$ (a finite sequence of core objects between $\mathbf{x}_i$ and $\mathbf{x}_j$ exists such that each next belongs to an ϵ- neighborhood of its predecessor);
4) A symmetric relation density-connectivity of two points $\mathbf{x}_i$, $\mathbf{x}_j$ (they should be density-reachable from a common core object).

All the points reachable from core objects can be factorized into maximal connected components serving as clusters. The points that are not connected to any core point can be considered as outliers, because they

are not covered by any cluster. The run time complexity of DBSCAN is $O(n \log n)$, if run on the feature set (that is, the $\mathbf{x}_i$ and $\mathbf{x}_j$ vectors above refer to feature vectors). With regard to the two parameters ϵ and MinPts, there is no straightforward way to fit them to data. To overcome this obstacle, the algorithm OPTICS was developed [16]. It builds an augmented ordering of data which is consistent with DBSCAN, but goes one step further: instead of just one point in the parameter space, OPTICS covers a spectrum of all different $\epsilon' \leq \epsilon$. The constructed ordering can be used automatically or interactively. With each point, OPTICS stores only two additional fields, the so-called core- and reachability-distances. For example, the core-distance is the distance to MinPts-nearest neighbor when it does not exceeds ϵ, or undefined otherwise. Experimentally, OPTICS exhibits runtime roughly equal to 1.6 of DBSCAN runtime, while maintaining the same complexity.

While OPTICS does generate a one-dimensional order as we require for our proposed kernel, the concept of density reachability may not translate well to minimizing the distance between subsequent features. Moreover, while it is an improvement over DBSCAN, it still has parameters. What follows, we introduce Ordering based on Hamiltonian Path (OHP), which is a parameter-free, greedy approach to minimize consecutive distances in a similar ordination [18].

Finding an ordering of features can be translated to a graph problem: a minimum-weight Hamiltonian path G' of G gives the ordering by reading the nodes from one end of the paths to the other. For instance, given three features with distance $d(f_1, f_2) = 2$, $d(f_1, f_3) = 1$, and $d(f_2, f_3) = 3$, the minimum-weight Hamiltonian path will be $f_3 \rightarrow f_1 \rightarrow f_2$ (Figure 2), and the respective feature order will be f_3, f_1, f_2.

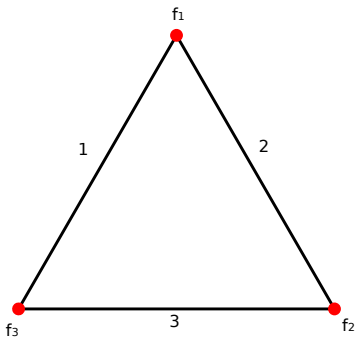


Fig. 2. A weighted K_3 for a feature set of three elements with example weights.

G is a complete graph, therefore such a path always exists, but finding it is an NP-complete problem. The greedy approach described in Algorithms 1 and 2 is similar to the nearest neighbor heuristic for the solution of the traveling salesman problem. It creates a graph $G' = (V', E')$, where $V' = V$ and $E' \subset E$. This G' graph is a spanning tree of G in which the maximum degree of a node is two, that is, the minimum spanning tree is a path between two nodes. Algorithm 1 is the main loop constructing the linear order starting from a randomly chosen seed element, and building on both the left and the right side of the existing order. In each iteration, candidates for both sides are identified, the algorithm to find the left candidate is described in Algorithm 2, the right counterpart is similar.

Algorithm 1 Ordering based on Hamiltonian Path

```
OHP(seed)
Add seed to V', remove seed from V
repeat
  x_l^c=FindLeftExpansionCandidate(V',V)
  x_r^c=FindRightExpansionCandidate(V',V)
  if x_l^c is closer to the left end of V' than x_r^c to right
  end then
    Connect x_l^c to the left of V', and remove it from
    V
  end if
  if x_r^c is closer to the right end of V' than x_l^c to the
  left end then
    Connect x_r^c to the right of V', and remove it from
    V
  end if
until V is not empty
```

Algorithm 2 FindLeftExpansionCandidate(V',V)

```
candidate=RandomElementOf(V),
candidateDistance=distance(LeftEndOf(V'),
candidate)
repeat
  currentDistance = distance(LeftEndOf(V'),
  NextElementOf(V))
  if currentDistance¡candidateDistance then
    candidate=currentElementOf(V) and candidate-
    Distance=currentDistance
  end if
until V is scanned
Return candidate
```

Algorithm 1 can be thought of as a modified Prim's algorithm [19], but it does not find the optimal minimum-weight spanning tree.

Without any index support, the computational cost of the algorithm is $O(n^2)$, since a full scan of the set of features is required in Step 2. If a tree-based spatial index can be used, the run-time is reduced to $O(n \log n)$ since searches are supported efficiently by spatial access methods such as the R*-tree [20] or the X-tree [21] for data from a vector space or the M-tree [22] for data from a metric space. The height of such a tree-based index is $O(logn)$ for a database of n objects in the worst case and, at least in low-dimensional spaces, a search with a "small" search region has to traverse only a limited number of paths.

Since the time complexity of OPTICS and OHP are the same, the crucial difference between them is that the latter is parameter-free. The two methods are compared in Section 4.1.

3.2 Wavelet Support Vector Kernels

If we assume that a set of features is reordered by an algorithm, the vector representation of an object may be regarded as a series of equally spaced observations of a continuous signal, where the consecutive observations are not in a temporal relation with one another, but the relation is defined by statistical relatedness.

We can consider reconstructing the hypothetical signal. The Whittaker-Shannon formula gives a simple example of how to reconstruct the signal from discrete values [23]. If we denote the reconstructed signal by $\hat{s}_{\mathbf{x}_i}$ of a vector $\mathbf{x}_i$, the general formula is

$$\hat{s}_{\mathbf{x}_i}(t) = \sum_{k=0}^{n-1} x_{ki} b(t-k), \quad t \in [1, n], \tag{5}$$

where t is a dummy variable, with an appropriately chosen basis $b(.)$ of L_2, or a subspace that of. Consider the following kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\hat{s}_{\mathbf{x}_i}, \hat{s}_{\mathbf{x}_j})_{L_2}, \tag{6}$$

where $(.,.)_{L_2}$ is the inner product of the L_2 space. The formula is similar to Equation 4 in the sense that it uses the inner product of the embedding space.

Using this kernel, a matching feature in two objects will be counted to its full score as in the vector space, while nearby related features will be counted less and less according their proximity to the matching feature. Assuming that the related features f_{i-1}, f_i, and f_{i+1} follow each other, consider the following example. The first object has the feature f_i, and so does the second objects. In Figure 3, it can be seen that feature f_i is counted the same way as it would be in a vector space model, the related features f_{i-1} and f_{i+1} are counted to a smaller extent, while other related features are considered even less.

If the two objects do not share the exact feature, only related features occur, for instance, f_{i-1} and f_{i+1}, respectively, then the feature f_i, placed between f_{i-1} and f_{i+1} in the same order, will be considered to some extent for the calculation of similarity (see Figure 4).

To calculate the kernel, let us expand Equation 6:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\hat{s}_{\mathbf{x}_i}, \hat{s}_{\mathbf{x}_j})_{L_2} = \tag{7}$$
$$\sum_{k=1}^{n} \sum_{k'=1}^{n} x_{ki} x_{k'j} \int_{[1,n]} b(t-k) b(t-k') d\lambda(t).$$

To simplify the above sum, we suggest to use non-orthogonal basis functions with a compact support. An orthogonal basis would give zero for all value of k and k' safe for $k = k'$, while basis functions with a non-compact support would require n^2 calculations for each

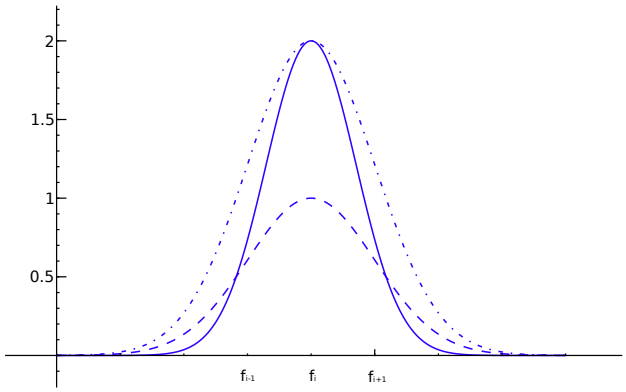


Fig. 3. Two objects with a matching feature f_i. Dotted line: Object-1. Dashed line: Object-2. Solid line: Their product.

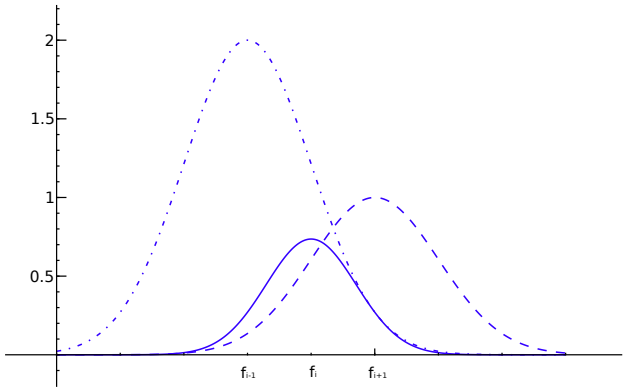


Fig. 4. Two objects with no matching features but with related features f_{i-1} and f_{i+1}. Dotted line: Object-1. Dashed line: Object-2. Solid line: Their product.

$\mathbf{x}_i$ and $\mathbf{x}_j$. If the support of the function is a continuous compact interval of length $2b$, then

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\hat{s}_{\mathbf{x}_i}, \hat{s}_{\mathbf{x}_j})_{L_2} = \tag{8}$$
$$\sum_{k=1}^{n} \sum_{k'=\max\{k-\lceil b \rceil,1\}}^{\min\{j+\lceil b \rceil,n\}} x_{ki} x_{l'j} \int_{[1,n]} b(t-k) b(t-k') d\lambda(t).$$

Thus the eventual kernel evaluation cost is O(bn).

The choice of the width may be more important than the actual functional form of the kernel. There may be little difference in the relevant part of the filter properties between e.g. a B-Spline and a Gaussian kernel [7]. Therefore we focus on a kernel which is easy to compute, and study the impact of width, instead of the choice of function. Choosing a small width of the kernels leads to high generalization error as it effectively decouples separate basis functions of the kernel expansion into very localized functions which is equivalent to memorizing the data, while a wide kernel tends to oversmooth [7].

Spline wavelets are extremely regular and usually symmetric or anti-symmetric. They can be designed to

have a compact support [24]. A B_n-spline is defined as

$$B_n(x) = \sum_{r=0}^{n+1} \frac{(-1)^r}{n!} \binom{n+1}{r} \left(x + \frac{n+1}{2} - r\right)_+^n,$$

where $(.)_+ = \max\{., 0\}$, and x is a dummy variable. B_n has a compact support $[-\frac{n+1}{2}, \frac{n+1}{2}]$.

Focusing on the integration part of Equation 8, by using the convolution property of B_n splines [25], we get

$$\int_{\mathrm{R}} B_n(t-k)B_n(l-t)d\lambda(t) = B_{2n+1}(k-l).$$

The above formula makes computation easier.

An inner product is a nondegenerate sesquilinear form, moreover it is positive definite. Let ϕ be the mapping that creates the L_2 functions from the discrete data, and X be a subspace of $\mathbb{R}^n$:

$$\phi : X \to L_2([1, n]).$$

The kernel can be expressed as

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\phi(\mathbf{x}_i), \phi(\mathbf{x}_j))_{L_2}. \tag{9}$$

This kernel is positive definite, therefore the conditions of Mercer's theorem are satisfied (see Equation 2); the proposed kernel is mathematically valid.

In the above, we assumed that the vector is a sequence of equally spaced observation. However, the observations do not have to be equally spaced, they can be randomly spaced by employing a scaling function $s(x)$ on the indices of features:

$$s(k) = \sum_{i=1, i<k} d(f_i, f_{i+1}).$$

The scaling function allows that features that are closer to each other get a higher score when calculating the kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\hat{s}_{\mathbf{x}_i}, \hat{s}_{\mathbf{x}_j})_{L_2} = \tag{10}$$

$$\sum_{k=1}^{n} \sum_{k'=1}^{n} x_{ki} x_{k'j} \int_{[1,n]} b(t - s(k))b(t - s(k'))d\lambda(t).$$

4 Experimental Results and Discussion

This section is divided into three parts. As indicated in Section 3.1 on the ordering algorithm, there may be different ways of generating the order required by the kernel. Section 4.1 compares two reordering algorithms, and shows that our ordination method fits the purpose much better.

Having chosen the optimal ordination algorithm, Section 4.2 benchmarks several kernels and the proposed kernel on binary classification problems in a range of domains.

Continuing to more complex tasks, Section 4.3 analyzes kernels for the multiclass, multilabel classification task of text categorization. An additional complexity, the measurement of distances between index terms (the features of text collections), is also briefly explained in this section. We have reported results with a similar kernel elsewhere [26], here we further expand to other text collections with the kernel described in this paper.

Data Set	OPTICS	OHP
Leukemia	8.275	4.599
Madelon	1817.804	1475.589
Gisette	28030.860	18097.961

TABLE 1
Average distance

4.1 Comparison of OPTICS and the Ordination Algorithm

Since OPTICS has parameters, we are interested in whether we are able to replicate or even improve the results with our proposed parameter-free method. We wanted to see how well the algorithms choose the features to be put next to each other. We calculated the consecutive distances, histograms and average distances by the Euclidean distance function. We used only the training part of the data sets to calculate the distances between the features. Table 1 summarizes the results. Figures 5-7 give further insight. In the diagrams of consecutive distances, the x axis corresponds to features, and the y axis to the distance between consecutive features. The histograms plot the frequency of distances.

The plot of consecutive distances on the Leukemia data set clearly indicates the greedy nature of OHP (Fig. 5(b)). In the beginning, the algorithm is able to choose features located nearby, but as the number of choices reduces, the quality of the ordination decreases. However, OPTICS shows an overall bad performance on consecutive distances, as also shown by the histogram and the average distance.

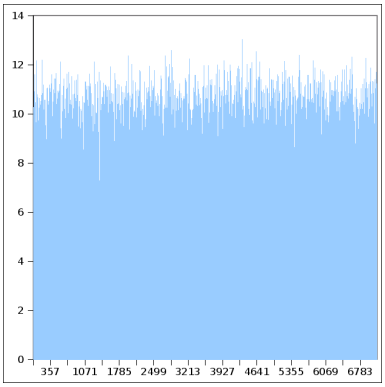
When studying the quality of ordination on the Madelon data set, the result is very similar. OHP initially greedily chooses good candidates, but towards the two ends of the spectrum the quality drops. While the histograms do not show a clear winner, the average distance is lower for OHP by nearly 20 %.

A curious phenomenon is captured in the plot of consecutive distances with OHP ordination on the Gisette data set. There are several local spikes which shows that the greedy algorithm is probably very far off from the optimal solution. However, OHP is still better then OPTICS, the latter method's average distance being one and the half times higher than the former's.

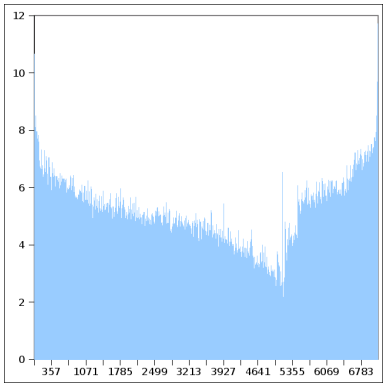
With regards to choosing the seed with OHP, similar plots of consecutive distances follow different choices of the seed element, with nearly identical average distances. Therefore the initial element has little influence over the result.

4.2 Experimental Results on General Datasets

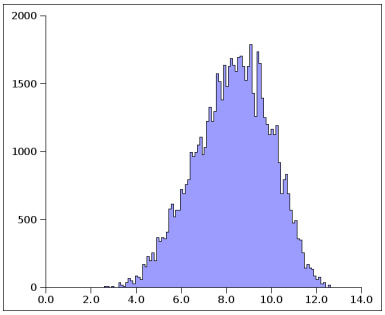
For binary experiments, we used the Leukemia, Madelon and Gisette data sets. Madelon and Gisette data sets were



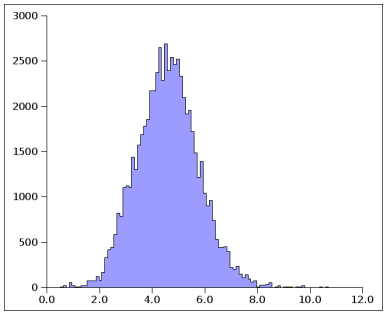
(a) Consecutive distances with OPTICS



(b) Consecutive distances with OHP

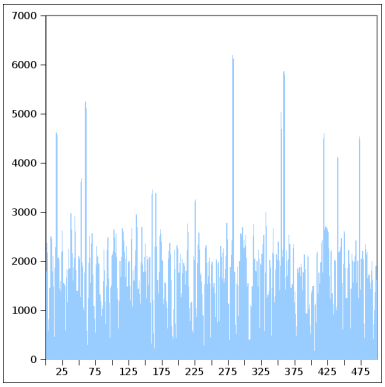


(c) Histogram of distances with OPTICS

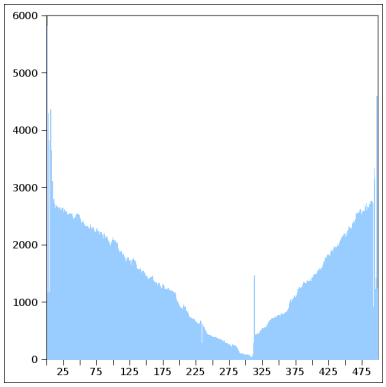


(d) Histogram of distances with OHP

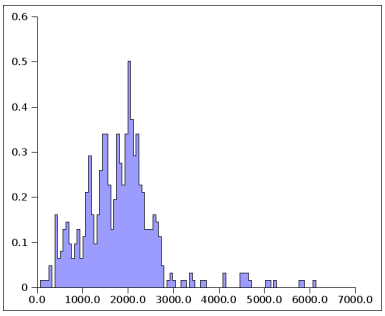
Fig. 5. The quality of ordination on the Leukemia data set



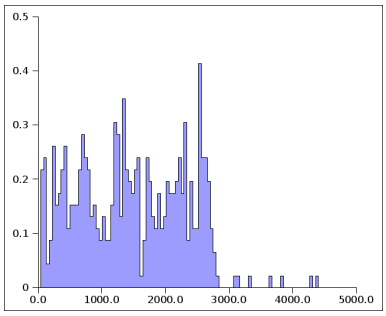
(a) Consecutive distances with OPTICS



(b) Consecutive distances with OHP

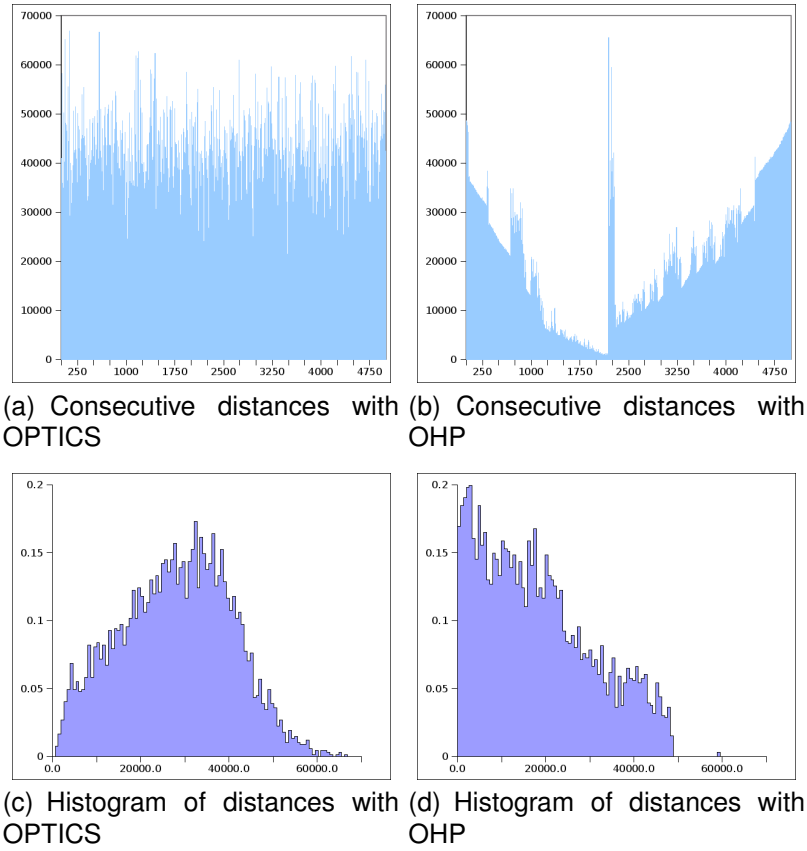


(c) Histogram of distances with OPTICS



(d) Histogram of distances with OHP

Fig. 6. The quality of ordination on the Madelon data set



(a) Consecutive distances with OPTICS

(b) Consecutive distances with OHP

(c) Histogram of distances with OPTICS

(d) Histogram of distances with OHP

Fig. 7. The quality of ordination on the Gisette data set

obtained from [27], while the Leukemia data set is from [28]. We assumed that there was no prior knowledge available on the relations between features.

The Leukemia DNA microarray data consist of 38 training and 34 testing instances, with 7129 features and a dense feature set.

Madelon is an artificial data set containing data points grouped in 32 clusters placed on the vertexes of a five dimensional hypercube and randomly labeled +1 or -1. The five dimensions constitute 5 informative features. 15 linear combinations of those features were added to form a set of 20 (redundant) informative features. Based on those 20 features one must separate the examples into the 2 classes (corresponding to the +1, -1 labels). A number of distractor features called 'probes' was added having no predictive power. The order of the features and patterns were randomized. The collection consists of 2000 training an 1800 test instances.

The task of Gisette is to discriminate between two confusable handwritten digits: the four and the nine. This is a two-class classification problem with sparse continuous input variables. The digits have been size-normalized and centered in a fixed-size image of dimension 28x28. The feature set consists of the original variables (normalized pixels) plus a randomly selected subset of products of pairs of variables. The pairs were sampled such that each pair member is normally distributed in a region of the image slightly biased upwards. The training set contained 6000 examples and test set 1000, with a total of 5000 features.

The classes of these databases are balanced in size, hence we used simple accuracy to measure the performance. Accuracy is the proportion of true results (both true positives and true negatives) in the benchmark collection.

We used the `libsvm` [28] library to benchmark the baseline kernels, and implement the suggested kernel. We used only C-SVMs, with the C penalty parameter left at the default value of one. For each kernel a wide range of kernel-specific parameters were benchmarked. each dataset. For polynomial kernels, the degree was varied (2 and 3), as well as the offset (0 and 1). RBF kernels converge very slowly in the `libsvm` implementation, therefore only the default parameter value (one over the number of features) and unit γ were benchmarked on all data sets, while a wider range of settings showed the insensitivy of the parameters on smaller collections. Table 2 summarizes the results for the baseline kernels with the best parameter settings for each kernel; the best result in a column is set in bold. Results were obtained with the same split of the respective collection across all kernels. Note that the RBF shows consistently poor results. We found that this is due to the fixed C parameter. Increasing it to 10, the RBF kernel typically performs similarly to the linear kernel with its γ parameter set to a low value (one over the number of features).

Kernel	Leukemia	Madelon	Gisette
Linear	**0.824**	0.550	0.976
Poly	0.618	**0.645**	**0.979**
RBF	0.675	0.500	0.500

TABLE 2
Results with baseline kernels

	0.5	1	2	5	10	20
Leukemia	0.824	0.824	0.824	0.824	**0.853**	**0.853**
Madelon	**0.585**	0.528	0.513	0.517	0.513	0.505
Gisette	0.976	0.976	0.976	0.979	**0.980**	0.970

TABLE 3
Accuracy, correlation distance, equally spaced observations

	0.5	1	2	5	10	20
Leukemia	0.824	0.824	0.824	0.824	**0.853**	**0.853**
Madelon	**0.581**	0.528	0.517	0.517	0.509	0.505
Gisette	0.976	0.976	0.976	**0.978**	0.975	0.970

TABLE 5
Accuracy, Euclidean distance, equally spaced observations

	0.5	1	2	5	10	20
Leukemia	0.794	0.824	0.824	**0.8636**	0.824	**0.8636**
Madelon	0.585	0.612	0.621	0.632	**0.649**	0.630
Gisette	0.976	0.978	0.978	0.979	**0.982**	**0.982**

TABLE 6
Accuracy, correlation distance, randomly spaced observations

We benchmarked three different distance functions to reorder the feature set: correlation, mutual information and Euclidean distance. The Eucledian distance between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as: $\mathrm{d}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{n}(x_{ki} - x_{kj})^2}$.

Correlation measures are typically used to measure attribute dependencies. Pearson's correlation coefficient is probably the most widely used measure for quantitative attributes [29]. The correlation coefficient $\mathrm{corr}(X_i, X_j)$ between two random variables X_i and X_j with expected values $E(X_i)$ and $E(X_j)$ and standard deviations σ_{X_i} and σ_{X_j} is defined as: $\mathrm{corr}(X_i, X_j) = \frac{E((X_i - E(X_i))(X_j - E(X_j)))}{\sigma_{X_i}\sigma_{X_j}}$, where E is the expected value operator. The correlation measure is not sensitive to nonlinear dependencies which do not manifest themselves in the covariance and can thus miss important features. This is in contrast to mutual information (MI) [30], thus MI can be a useful way to measure interdependencies between features. MI is defined as $I(X_i; X_j) = \sum_{x_j \in X_j}\sum_{x_i \in X_i} p(x_i, x_j)\log\left(\frac{p(x_i,x_j)}{p_1(x_i)\,p_2(x_j)}\right)$, swhich is not a metric. However, using the joint entropy $H(X_i, X_j) = -\sum_{x_i,x_j} p_{x_i,x_j}\log(p_{x_i,x_j})$, the expression $d(X_i, X_j) = H(X_i, X_j) - I(X_i; X_j)$ is a metric.

Tables 3, 4 and 5 show the results regarding the vectors as equally spaced observations, and Tables 6, 7 and 8 show the results with randomly spaced observations; best result in a row is set in bold.

As expected, a narrow support will result in identical accuracy as with linear kernels, since a narrow support means that no neighboring features are considered. Madelon is the only exception. The linear kernel converges extremely slowly to a solution on this data set, it needs over a million iterations, and the same holds for wavelet kernels. The difference in accuracy is probably due to numerical instability.

For the binary problems, considering equally spaced observations, there is a clear advantage over linear kernels, and with the exception of the highly nonlinear Madelon, over any baseline kernel.

Considering randomly spaced observations, the proposed kernels perform even better on the binary collections, peaking in performance with a support 10 to 20 wide. It is interesting to note that [13] have found that Daubechies wavelets with a support of 20 are the most efficient in identifying voice disorders. As opposed to B-spline wavelets, Daubechies wavelets are orthogonal.

4.3 Experimental Results in Text Classification

Text classification tasks are typically multilabel, multiclass problems, that is, there are several classes, and an

	0.5	1	2	5	10	20
Leukemia	0.824	0.824	0.824	0.824	**0.853**	**0.853**
Madelon	**0.581**	0.522	0.513	0.517	0.513	0.513
Gisette	0.976	0.976	0.976	0.976	**0.978**	0.970

TABLE 4
Accuracy, mutual information-based distance, equally spaced observations

	0.5	1	2	5	10	20
Leukemia	0.794	0.824	0.824	**0.8636**	0.824	**0.8636**
Madelon	0.585	0.611	0.624	**0.639**	0.635	0.630
Gisette	0.976	0.978	0.978	0.978	**0.983**	0.982

TABLE 7
Accuracy, mutual information-based distance, randomly spaced observations

	0.5	1	2	5	10	20
Leukemia	0.794	0.824	0.824	**0.8636**	0.824	**0.8636**
Madelon	0.583	0.610	0.628	0.632	**0.649**	0.632
Gisette	0.976	0.978	0.978	**0.981**	0.978	0.980

TABLE 8
Accuracy, Euclidean distance, randomly spaced observations

instance may belong to several classes simultanously. Since SVMs are defined on binary classes, a multilabel, multiclass problem is broken down to a series of binary problems, in which a target class is trained against the rest (also known as the 1-against-all approach) [31].

Another key difference to the problems discussed in the previous section is that prior knowledge on the relations between the features is often available. In fact, text classification has a history of incorporating distributional and prior knowledge in its representation model. Therefore this section begins with a brief review on the so-called semantic kernels to properly define the baseline for comparison and benchmark.

Contrasting Hoenkamp's approach (Section 2) with the original setup of latent semantic indexing [32], a latent semantic kernel can be written in the following linear form [33]:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i' U I_k U' \mathbf{x}_j = \mathbf{x}_i' S' S \mathbf{x}_j,$$

where U is a unitary matrix from the SVD of the space ($U\Sigma V$), I_k is the identity matrix with only the first k diagonal elements nonzero, and $S = I_k U'$ is the semantic smoothing matrix. By this rank reduction terms that occur together very often in the same documents are merged into a single dimension of the feature space. The diversity of expressions and the minor differences in senses are lost. In return, theoretically, those few hundred concepts, which make up the meaning of the documents, can be identified easily. The dimensions of the reduced space correspond to the axes of greatest variance.

The above S semantic smoothing kernel is just one of the many possible, and, in general, it defines some relation among the index terms, and modifies the original orthogonality of the vector space model accordingly. More recent papers argued in favor of constructing the semantic smoothing matrix by using external lexical resources such as WordNet: entries in this matrix are inversely proportional to the length of the WordNet hierarchy path linking the two terms [34]. The performance, measured over the 20NewsGroups corpus, showed an improvement of 2 % over the the basic vector space method. However, the semantic matrix S was almost fully dense, hence computational issues arose. In a similar construction, [35] defined the matrix entries as weights of superconcepts of the two terms in the WordNet hierarchy. Focusing on special subcategories of Reuters-21578 and on the TREC Question Answering Dataset, they showed consistent improvement over the baseline. As [36] pointed out, polysemy will remain a problem in semantic smoothing kernels. A more complex way of calculating the semantic similarity as the matrix entries was also proposed [37].

These approaches to construct the semantic smoothing matrix raise the question how to measure the semantic similarity between terms in general. Various lexical resource-based [38] and distributional measures [39] have been proposed to measure semantic relatedness and distance between terms in other fields of natural language processing.

Distributional similarity, as studied by language technology, covers an important kind of theories of word meaning. Also called the contextual theory of meaning [40], the underlying distributional hypothesis of [41] is often cited for explaining how word meaning enters information processing [42], and basically equals the claim "meaning is use" in language philosophy [43]. Before attempts to utilize lexical resources for the same purpose, this used to be the sole source of word semantics in information retrieval, inherent in the exploitation of term occurrences and term co-occurrences [44], [45], [46], [47], including multiple-level term co-occurrences [48]. As in the section on general binary classification problems, correlation, Euclidean and MI distances were benchmarked, though due to the similarity of results and for the sake of brevity, this section only presents the correlation distance based results.

All approaches to measuring semantic relatedness that use a lexical resource regard the resource as a network or a directed graph, making use of the structural information embedded in the graph [49], [38]. One of the earliest lexical resource based measures is the edge counting method. The shortest path in the network between the two target terms is determined. The more edges there are between two terms, the more distant they are. If all the edges are of equal length, then the number of intervening edges is a measure of the distance [50]. In determining the overall edge based similarity, most methods just simply sum up all the edge weights along the shortest path.

There are certain advantages in measuring semantic association by combining a network structure with corpus statistics. The incorporation of a manually built knowledge base may complement the statistical approach where understanding of the text is impossible. The statistical model can take advantage of a conceptual space structured by a lexical resource [49]. Following the notation in information theory, the information content (IC) of a concept/sense s can be quantified as follows: $\mathrm{IC}(s) = -\log P(s)$, where $P(s)$ is the probability of encountering an instance of sense s. In the case of the hierarchical structure, where a sense in the hierarchy subsumes those ones below it, this implies that $P(s)$ is monotonic as one moves up in the hierarchy. As the node's probability increases, its information content or its informativeness decreases. If there is a unique top node in the hierarchy, then its probability is 1, hence its information content is 0. The information content decreases as senses get more and more abstract (that is, they are higher in the WordNet hypernym hierarchy.) The so-called Jiang-Conrath distance function can be simplified as follows:

$$d(s_1, s_2) = \mathrm{IC}(s_1) + \mathrm{IC}(s_2) - 2\mathrm{IC}(\mathrm{LSuper}(s_1, s_2))$$

This distance measure also satisfies the properties of a metric [49].

Any lexical resource-based distance defines the distance between senses, and not between terms. Terms in natural languages can be polysemous, hence another formula is necessary to calculate the distance between terms. Word-sense disambiguation and other natural language processing tasks normally use the following formula: $\mathrm{d}(t_1, t_2) = \min_{s_1 \in \mathrm{sen}(t_1), s_2 \in \mathrm{sen}(t_2)} d(s_1, s_2)$, where t_1 and t_2 are two terms, and $\mathrm{sen}(t_i)$ is the set of senses of t_i. However, we argue in favor of taking the maximum:

$$\mathrm{d}(t_1, t_2) = \max_{s_1 \in \mathrm{sen}(t_1), s_2 \in \mathrm{sen}(t_2)} d(s_1, s_2), \qquad (11)$$

This is a more conservative approach. A term has a unique position in the semantic reordering, and the order does not account for polysemy. Equation 11 ensures that terms that are really closely related get a distance closer to zero.

Prior to the semantic ordering, terms were assumed to be in alphabetic order. Measuring the Jiang-Conrath distance between adjacent terms, the average distance was 0.84 on the Reuters corpus (see below). Note that the Jiang-Conrath distance was normalized to the interval $[0, 1]$. There were few terms with zero or little distance between them. This is due to terms which are related and start with the same word or stem. For example, *account, account executive, account for, accountable*. The same average distance after reordering the terms with the proposed algorithm and the Jiang-Conrath distance was 0.28 on the same corpus. About one third of the terms had very little distance between each other. Nevertheless, over 10 % of the total terms still had the maximum distance. This is due to the non-optimal nature of the proposed term-ordering algorithm. These terms add noise to the classification. The noisy terms occur typically at the two sides of the scale, that is, the leftmost terms and the rightmost terms. While it is easy to find close terms in the beginning, as the algorithm proceeds, fewer terms remain in the pool to be chosen. For instance, *brand, brand name, trade name, label* are in the 33rd, 34th, 35th and 36th position on the left side counting from the seed respectively, while *windy, widespread, willingly, whatsoever, worried, worthwhile* close the left side, apparently sharing little in common.

We used the Reuters-21578 and the 20News collections to benchmark the performance of the proposed new kernel. Both data sets were obtained from [27]. Stop words were removed, and stemming was performed. For Reuters, we used the ModApte split benchmarking the performance over all categories and the top ten categories. For 20News, three different splits were made, using 50 %, 60 %, and 70 % of the collection as the training set.

We also benchmarked a real-world digital library. Strathprints is "an institutional eprint repository for making research papers and other scholarly publications widely available on the Internet" at the University of Strathclyde, UK [51], its hosting and technical support provided by the Department of Computer and Information Sciences (CIS). Eprints and usage statistics software have been installed, configured and managed by the Centre for Digital Library Research (CDLR) at the same university. Its digital objects are indexed by the LCSH classification scheme. Out of 6869 records, the size of the database on 13th June 2009, we downloaded and processed 5946 abstracts, the rest being records without abstracts or duplicates. With 14 ppt files removed, only abstracts in doc, html and pdf format were indexed by their LCSH tags. Keywords were obtained by a WordNet-based stemmer using the controlled vocabulary of the lexical database (see below), resulting in 21718 keywords in the full-text documents.

To evaluate the performance, we calculated precision (p) and recall (r) with regard to a class c_k. Estimates can be obtained as [52]:$\mathrm{p}(c_k) = \frac{TP_k}{TP_k + FP_k}$, $\mathrm{r}(c_k) = \frac{TP_k}{TP_K + FN_k}$, where TP_k is the number of correctly classified instances under c_k, FP_k is the number of false positives, and FN_k is the number of false negatives, that is, the errors of omission. To obtain overall estimates of precision and recall for all classes, we used micro-averaging, where precision and recall are obtained by summing over all individual decisions. We used the F_1 measure to obtain a single number to characterize performance [53]: $F_1 = \frac{2pr}{p+r}$.

In preparing the index terms, we restricted the vocabulary to the terms of WordNet 3.0 in order to be able to calculate the similarity score between any two terms. Stop words were removed in advance. Multiple word expressions were used to fully utilize WordNet. We used the built-in stemmer of WordNet, which is able to distinguish between different parts-of-speeches if the form of the word is unambiguous. For example, {accommodates, accommodated, accommodation} was stemmed to {accommodate, accommodate, accommodation}. We used term frequency as term weighting, and the same controlled vocabulary for purely distributional models to achieve better comparability.

We applied the `libsvm` [28] library to benchmark the baseline kernels, and implement the suggested kernel. We used only C-SVMs, for each kernel a wide range of kernel-specific parameters were benchmarked. Table 9 summarizes the results for the baseline kernels with the best parameter settings for each kernel; the best result in a row is set in bold. Linear kernel refers to a simple linear kernel without semantic smoothing. The latent semantic kernel was tested with keeping 300 dimensions, as this setting gave the best results in [33]; our results confirm that the latent semantic kernel is able to closely approximate the performance of, but not to outperform, a linear kernel. Edge-counting refers to the semantic smoothing kernel defined by [34]. It should be noted that this semantic smoothing kernel improves performance if the training set is smaller, or there are categories with few training instances (20News 50 % and Reuters All).

Tables 10, 11 and 12 show the results with the proposed kernels with equally spaced observations. Correlation, edge-counting and Jiang-Conrath refers to the

	Linear	Latent Semantic	Edge-Counting
Reuters Top-10	**0.901**	0.888	0.892
Reuters All	0.845	0.839	**0.851**
20News 50 %	0.672	0.664	**0.682**
20News 60 %	0.693	0.674	**0.699**
20News 70 %	0.707	0.698	**0.711**
Strath Top-level	**0.582**	0.544	0.580
Strath Refined	0.573	0.542	**0.574**

TABLE 9
F_1 results with baseline kernels

	0.5	1	2	5	10	20
Reuters Top-10	**0.901**	0.867	0.867	0.884	0.867	0.871
Reuters All	**0.845**	**0.845**	0.829	0.825	0.818	0.819
20News 50 %	**0.672**	**0.672**	0.670	0.669	0.655	0.655
20News 60 %	**0.693**	0.690	0.690	0.690	0.688	0.690
20News 70 %	**0.707**	**0.707**	0.702	0.700	0.703	0.698
Strath Top-level	**0.582**	**0.582**	0.554	0.531	0.557	0.491
Strath Refined	**0.573**	**0.573**	0.551	0.520	0.528	0.511

TABLE 10
F_1 measure, correlation distance, equally spaced observations

	0.5	1	2	5	10	20
Reuters Top-10	**0.901**	0.899	0.904	**0.906**	**0.906**	0.905
Reuters All	**0.845**	**0.845**	0.833	0.823	0.811	0.804
20News 50 %	0.672	0.672	0.663	**0.695**	0.634	0.634
20News 60 %	0.693	0.693	**0.711**	0.665	0.649	0.640
20News 70 %	0.707	0.707	0.711	**0.723**	0.698	0.693
Strath Top-level	**0.582**	**0.582**	0.545	0.530	0.553	0.496
Strath Refined	**0.573**	**0.573**	0.558	0.523	0.521	0.510

TABLE 12
F_1 measure, Jiang-Conrath distance, equally spaced observations

	0.5	1	2	5	10	20
Reuters Top-10	**0.901**	0.826	0.874	0.867	0.867	0.871
Reuters All	**0.845**	0.829	0.814	0.806	0.806	0.819
20News 50 %	**0.672**	0.652	0.670	0.650	0.655	0.655
20News 60 %	**0.693**	0.674	0.674	0.670	0.676	0.662
20News 70 %	**0.707**	0.694	0.699	0.692	0.690	0.693
Strath Top-level	**0.572**	0.531	0.506	0.475	0.469	0.471
Strath Refined	**0.566**	0.524	0.496	0.462	0.434	0.453

TABLE 13
F1 measure, correlation distance, randomly spaced observations

distance function used by the proposed ordering algorithm. Another distributional measure, mutual information, was also benchmarked, but the results are identical to that of correlation, hence we excluded them for brevity. Compact supports with length 0.5, 1, 2, 5 and 10 were benchmarked. For each distance function, 0.5 and 1-long support provided identical results with the linear kernel. This was expected, since with such narrow supports no neighboring term is considered in calculating the similarity between two documents.

For distributional measures, the optimal support was of length 1 (or 0.5), that is, the equivalent of a linear kernel. Polysemy seems to introduce too much noise, statistical relatedness is established on the training set, and does not necessarily hold for the test set if the terms appear in different senses.

Edge-counting, as a purely lexical resource-based distance metric, fares much better. Note that we use Equation 11 instead of Equation **??**, thus making sure that the ordering algorithm places only terms next to each other that are closely related. As it was the case with the edge-counting-based linear semantic smoothing kernels, the greatest improvements can be seen if the training set was relatively small, or there were only a few training instances for some of the classes. Performance peaked with a support length of 5 or 10.

Tables 13, 14 and 15 show the results with randomly spaced observations. Results with no exception are worse than the baseline. This is due the noise inherent in terms by the polysemous nature.

By far the best results were obtained with a Jiang-Conrath distance based wavelet kernel with equally spaced observations, with support length 5 or 10. This is a composite measure incorporating distributional information, as well as information of a lexical hierarchy.

5 Conclusions

Feature weighting is a powerful tool to discount less important features, but not fully eliminate them to maintain the richness of representation. Preserving the rich-

	0.5	1	2	5	10	20
Reuters Top-10	0.901	0.901	0.898	**0.902**	0.898	0.875
Reuters All	0.845	0.845	0.847	**0.867**	0.838	0.838
20News 50 %	0.672	0.675	0.681	**0.690**	0.683	0.685
20News 60 %	0.693	0.691	**0.702**	0.697	0.690	0.687
20News 70 %	0.707	0.707	**0.712**	0.710	0.709	0.702
Strath Top-level	**0.582**	**0.582**	0.556	0.530	0.554	0.497
Strath Refined	**0.573**	**0.573**	0.558	0.525	0.521	0.510

TABLE 11
F_1 measure, edge counting distance, equally spaced observations

	0.5	1	2	5	10	20
Reuters Top-10	**0.901**	0.821	0.844	0.877	0.864	0.872
Reuters All	**0.845**	0.839	0.823	0.808	0.811	0.806
20News 50 %	**0.672**	0.657	0.652	0.652	0.647	0.646
20News 60 %	**0.693**	0.669	0.675	0.671	0.656	0.663
20News 70 %	**0.707**	0.690	0.698	0.689	0.690	0.690
Strath Top-level	**0.572**	0.543	0.509	0.484	0.469	0.473
Strath Refined	**0.566**	0.527	0.501	0.464	0.429	0.442

TABLE 14
F_1 measure, edge counting distance, randomly spaced observations

	0.5	1	2	5	10	20
Reuters Top-10	**0.901**	0.820	0.854	0.856	0.846	0.825
Reuters All	**0.845**	0.818	0.824	0.826	0.802	0.814
20News 50 %	**0.672**	0.652	0.663	0.652	0.645	0.648
20News 60 %	**0.693**	0.679	0.684	0.671	0.675	0.669
20News 70 %	**0.707**	0.693	0.702	0.688	0.691	0.693
Strath Top-level	**0.572**	0.529	0.515	0.475	0.473	0.471
Strath Refined	**0.567**	0.533	0.503	0.492	0.454	0.453

TABLE 15
F_1 measure, Jiang-Conrath distance, randomly spaced observations

ness is particularly important when dealing with sparse data, such as text collections, or data sets with many, but highly correlated features. In fact, for sparse data, feature expansion is widely used, but not with weighted expansion features. This paper offers a representation via kernel methods which expands the idea of feature expansion by weighting the expansion features. The proposed compactly supported basis function kernels draw on the concept of existing wavelet kernels, expanding on the core idea of applying wavelet analysis for support vector machines.

Supervised machine learning, and support vector machines in particular, gain a powerful and computationally feasible method for feature engineering. While feature weighting and feature selection has been researched even in the kernel space prior to this work, the proposed combination of feature weighting and expansion is novel.

A more general framework for wavelet kernels has been introduced. While other wavelet kernels have been proved to be admissible, they did not use the inner product of the embedding L_2 space. By regarding the vector representation of an object as a sample of a hypothetical signal, a new family of wavelet kernels has been identified which uses the L_2 inner product. The underlying assumption is that there is a relation between consecutive features, this assumption has been addressed by an algorithm that orders the feature set according to an almost arbitrary relatedness measure.

The proposed kernels do not have additional storage needs, yet they are able to incorporate feature interdependence into calculating a similarity score. The running time and computational complexity of the kernels is close to that of a linear kernel, and can be adjusted by the length of the support of the basis function. By increasing the size of the cache of state-of-the-art implementations of support vector machines, the running time can be reduced significantly, though the number of iterations required will not decrease.

Experiments on a variety of general data set show significant improvement over baseline kernels. Especially if many features are relevant, but the features are strongly related to one another, the proposed kernels give a clear advantage.

Text categorization showed significant improvements with the proposed kernels. This result is in line with the findings on general data sets: texts have many relevant and related features. The kernels seem to have a special edge over baseline methods if less training data is available.

The intricate nature of term relatedness gave way to several semantic relatedness measures: both distributional measures and lexical resource-based measures, as well as combined approaches have been developed. These measures were not explored for use in large-scale experiments before, they were employed only in computationally demanding natural language processing tasks, such as word-sense disambiguation. CSBF kernels integrate these measures efficiently, hence they are able to capture prior knowledge that may be encoded in the relatedness measure. If the wavelet kernels are regarded as a model of language representation. The distributional hypothesis meets the referential theory of meaning first at the semantic ordering of terms. While high-quality lexical resources enable such ordering in themselves, the ordering can benefit from data derived from a specific corpus being studied: composite semantic relatedness measures such as the Jiang-Conrath similarity operate this way. Once the semantic order is constructed, weights expressing statistical relationships between terms and documents are borrowed from the vector space model to form the basis for constructing hypothetical signals of content. By adjusting the length of the support of the basis functions, one may subtly balance the trade-off between distributional statistics and prior knowledge.

REFERENCES

[1] H. Liu and H. Motoda, *Computational methods of feature selection*. Chapman & Hall/CRC Boca Raton, 2008.

[2] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1995.

[3] L. Zhang, W. Zhou, and L. Jiao, "Wavelet support vector machine," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 1, pp. 34–39, 2004.

[4] E. Fonseca, R. Guido, A. Silvestre, and J. Pereira, "Discrete wavelet transform and support vector machine applied to pathological voice signals identification," in *Proceedings of ISM-05, 7th IEEE International Symposium on Multimedia*. IEEE Computer Society, December 2005, pp. 785–789.

[5] T. Alexandrov, J. Decker, B. Mertens, A. Deelder, R. Tollenaar, P. Maass, and H. Thiele, "Biomarker discovery in MALDI-TOF serum protein profiles using discrete wavelet transformation," *Bioinformatics*, vol. 25, no. 5, p. 643, 2009.

[6] E. Hoenkamp, "Unitary operators on the document space," *Journal of the American Society for Information Science and Technology*, vol. 54, no. 4, pp. 314–320, 2003.

[7] A. Smola, B. Schölkopf, and K. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, no. 4, pp. 637–649, 1998.

[8] T. Li, Q. Li, S. Zhu, and M. Ogihara, "A survey on wavelet applications in data mining," *SIGKDD Explorations*, vol. 4, no. 2, pp. 49–68, 2002.

[9] H. Szu, B. Telfer, and S. Kadambe, "Neural network adaptive wavelets for signal representation and classification," *Optical Engineering*, vol. 31, p. 1907, 1992.

[10] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wavecluster: A multi-resolution clustering approach for very large spatial databases," in *Proceedings of VLDB-98, 24th International Conference on Very Large Data Bases*. New York City, NY, USA: IEEE, August 1998, pp. 428–439.

[11] L. Zhang, W. Zhou, and L. Jiao, "Support vector machines based on the orthogonal projection kernel of father wavelet," *International Journal of Computational Intelligence and Applications*, vol. 5, no. 3, pp. 283–303, 2005.

[12] F. Schleif, M. Lindemann, M. Diaz, P. Maaß, J. Decker, T. Elssner, M. Kuhn, and H. Thiele, "Support vector classification of proteomic profile spectra based on feature extraction with the biorthogonal discrete wavelet transform," *Computing and Visualization in Science*, vol. 12, no. 4, pp. 1–11, 2009.

[13] E. Fonseca, R. Guido, P. Scalassara, C. Maciel, and J. Pereira, "Wavelet time-frequency analysis and least squares support vector machines for the identification of voice disorders," *Computers in Biology and Medicine*, vol. 37, no. 4, pp. 571–578, 2007.

[14] S. Tuntisak and S. Premrudeepreechacharn, "Harmonic detection in distribution systems using wavelet transform and support vector machine," in *Proceedings of Powertech-07, Conference of the IEEE Power Engineering Society*, Lausanne, Switzerland, July 2007, pp. 1540–1545.

[15] P. Hosseini, F. Almasganj, T. Emami, R. Behroozmand, S. Gharibzade, and F. Torabinezhad, "Local discriminant wavelet packet basis for voice pathology classification," in *Proceedings of ICBBE-08, 2nd International Conference on Bioinformatics and Biomedical Engineering*, Shanghai, China, May 2008, pp. 2052–2055.

[16] M. Ankerst, M. Breunig, H. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proceedings of SIGMOD-99, International Conference on Management of Data*. ACM Press, New York, NY, USA, 1999, pp. 49–60.

[17] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of SIGKDD-96, 2nd International Conference on Knowledge Discovery and Data Mining*, vol. 96. Portland, OR, USA: ACM Press, New York, NY, USA, August 1996, pp. 226–231.

[18] P. Wittek, C. Tan, and S. Darányi, "An ordering of terms based on semantic relatedness," in *Proceedings of IWCS-09, 8th International Conference on Computational Semantics*, H. Bunt, Ed. Tilburg, The Netherlands: Springer, January 2009.

[19] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to algorithms*. Cambridge, MA, USA: MIT Press, 2001.

[20] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: an efficient and robust access method for points and rectangles," *SIGMOD Record*, vol. 19, no. 2, pp. 322–331, 1990.

[21] S. Berchtold, D. Keim, and H. Kriegel, "The X-tree: An index structure for high-dimensional data," in *Readings in multimedia computing and networking*, K. Jeffay and H. Zhang, Eds. Morgan Kaufmann, 2001, p. 451.

[22] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *Proceedings of VLDB-97, 23rd International Conference on Very Large Data Bases*. Athens, Greece: IEEE, August 1997, pp. 426–435.

[23] H. Weaver, *Theory of Discrete and Continuous Fourier Analysis*. New York, NY, USA: John Wiley & Sons, 1988.

[24] M. Unser, "Ten good reasons for using spline wavelets," in *Proceedings of SPIE, Wavelet Applications in Signal and Image Processing V*, vol. 3169, 1997, pp. 422–431.

[25] M. Unser and A. Aldroubi, "Polynomial splines and wavelets: A signal processing perspective," in *Academic Press Wavelet Analysis And Its Applications Series*. Academic Press Professional, Inc. San Diego, CA, USA, 1993, pp. 91–122.

[26] P. Wittek and C. Tan, "A kernel-based feature weighting for text classification," in *Proceedings of IJCNN-09, IEEE International Joint Conference on Neural Networks*. Atlanta, GA, USA: IEEE Computer Society Press, Los Alamitos, CA, USA, June 2009, pp. 3373–3379.

[27] A. Asuncion and D. Newman. (2007) UCI machine learning repository. Department of Information and Computer Science, University of California. Irvine, CA, USA.

[28] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, 2001, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[29] D. Nazareth, E. Soofi, and H. Zhao, "Visualizing attribute interdependencies using mutual information, hierarchical clustering, multidimensional scaling, and self-organizing maps," in *Proceedings of HICCS-07, 40th Hawaii International Conference on System Sciences*, vol. 40, no. 2. Waikoloa, HI, US: IEEE, January 2007, pp. 907–917.

[30] A. Kraskov, H. Stoegbauer, R. Andrzejak, and P. Grassberger, "Hierarchical clustering using mutual information," *Europhysics Letters*, vol. 70, no. 2, pp. 278–284, 2005.

[31] C. Hsu and C. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[32] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[33] N. Cristianini, J. Shawe-Taylor, and H. Lodhi, "Latent semantic kernels," *Journal of Intelligent Information Systems*, vol. 18, no. 2, pp. 127–152, 2002.

[34] G. Siolas and F. d'Alché Buc, "Support vector machines based on a semantic kernel for text categorization," in *Proceedings of IJCNN-00, IEEE International Joint Conference on Neural Networks*. Austin, TX, USA: IEEE Computer Society Press, Los Alamitos, CA, USA, 2000.

[35] S. Bloehdorn, R. Basili, M. Cammisa, and A. Moschitti, "Semantic kernels for text classification based on topological measures of feature similarity," in *Proceedings of ICDM-06, 6th IEEE International Conference on Data Mining*, Hong Kong, December 2006.

[36] D. Mavroeidis, G. Tsatsaronis, M. Vazirgiannis, M. Theobald, and G. Weikum, "Word sense disambiguation for exploiting hierarchical thesauri in text classification," in *Proceedings of PKDD-05, 9th European Conference on the Principles of Data Mining and Knowledge Discovery*. Porto, Portugal: Springer, October 2005, pp. 181–192.

[37] R. Basili, M. Cammisa, and A. Moschitti, "Effective use of WordNet semantics via kernel-based learning," in *Proceedings of CoNLL-05, 9th Conference on Computational Natural Language Learning*. Ann Arbor, MI, USA: ACL, Morristown, NJ, USA, June 2005, pp. 1–8.

[38] A. Budanitsky and G. Hirst, "Evaluating WordNet-based measures of lexical semantic relatedness," *Computational Linguistics*, vol. 32, no. 1, pp. 13–47, 2006.

[39] S. Mohammad and G. Hirst, "Distributional measures as proxies for semantic relatedness," 2005, submitted for publication.

[40] J. Lyons, *Semantics*. New York, NY, USA: Cambridge University Press, 1977.

[41] Z. Harris, "Distributional structure," in *Papers in structural and transformational Linguistics*, ser. Formal Linguistics, Z. Harris, Ed. New York, NY, USA: Humanities Press, 1970, pp. 775–794.

[42] J. Karlgren and M. Sahlgren, "From words to understanding," in *Foundations of Real-World Intelligence*, Y. Uesaka, P. Kanerva, and H. Asoh, Eds. CSLI Publications, 2001, pp. 294–308.

[43] L. Wittgenstein, *Philosophical Investigations*. Oxford, UK: Blackwell Publishing, 1967.

[44] Y. Wilks, D. Fass, C. Guo, J. McDonald, T. Plate, and B. Slator, "Providing machine tractable dictionary tools," *Machine Translation*, vol. 5, no. 2, pp. 99–154, 1990.

[45] S. I. Gallant, "A practical approach for representing context and for performing word sense disambiguation using neural networks," *Neural Computation*, vol. 3, pp. 293–309, 1991.

[46] G. Grefenstette, "Use of syntactic context to produce term association lists for text retrieval," in *Proceedings of SIGIR-92, 15th International Conference on Research and Development in Information Retrieval*. Copenhagen, Denmark: ACM Press, New York, NY, USA, June 1992, pp. 89–97.

[47] H. Schütze and T. Pedersen, "A co-occurrence-based thesaurus and two applications to information retrieval," *Information Processing and Management*, vol. 3, no. 33, pp. 307–318, 1997.

[48] A. Kontostathis and W. Pottenger, "A framework for understanding latent semantic indexing (LSI) performance," *Information Processing and Management*, vol. 42, no. 1, pp. 56–73, 2006.

[49] J. Jiang and D. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proceedings of ROCLING-97, International Conference on Research in Computational Linguistics*, Taipei, Taiwan, 1997, pp. 19–33.

[50] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 1, pp. 17–30, 1989.

[51] A. Dawson and A. Slevin. (2008) Repository case history: University of Strathclyde Strathprints. [Online]. Available: http://www.rsp.ac.uk/repos/casestudies/pdfs/strathclyde.pdf

[52] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[53] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information Retrieval*, vol. 1, no. 1, pp. 69–90, 1999.