Nonparametric Transforms of Graph Kernels for Semi-Supervised Learning

Xiaojin Zhu† **Jaz Kandola**‡ **Zoubin Ghahramani**‡† **John Lafferty**†

†School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA

‡Gatsby Computational Neuroscience Unit
University College London
17 Queen Square
London WC1N 3AR UK

Abstract

We present an algorithm based on convex optimization for constructing kernels for semi-supervised learning. The kernel matrices are derived from the spectral decomposition of graph Laplacians, and combine labeled and unlabeled data in a systematic fashion. Unlike previous work using diffusion kernels and Gaussian random field kernels, a nonparametric kernel approach is presented that incorporates order constraints during optimization. This results in flexible kernels and avoids the need to choose among different parametric forms. Our approach relies on a quadratically constrained quadratic program (QCQP), and is computationally feasible for large datasets. We evaluate the kernels on real datasets using support vector machines, with encouraging results.

1 Introduction

Semi-supervised learning has been the focus of considerable recent research. In this learning problem the data consist of a set of points, with some of the points labeled and the remaining points unlabeled. The task is to use the unlabeled data to improve classification performance. Semi-supervised methods have the potential to improve many real-world problems, since unlabeled data are often far easier to obtain that labeled data.

Kernel-based methods are increasingly being used for data modeling and prediction because of their conceptual simplicity and good performance on many tasks. A promising family of semi-supervised learning methods can be viewed as constructing kernels by transforming the spectrum of a “local similarity” graph over labeled and unlabeled data. These kernels, or regularizers, penalize functions that are not smooth over the graph [7]. Informally, a smooth eigenvector has the property that two elements of the vector have similar values if there are many large weight paths between them on the graph. This results in the desirable behavior of the labels varying smoothly over the graph, as sought by, e.g., spectral clustering approaches [2], diffusion kernels [5], and the Gaussian random field approach [9]. However, the modification to the spectrum, called a spectral transformation, is often a function chosen from some parameterized family. As examples, for the diffusion kernel the spectral transformation is an exponential function, and for the Gaussian field kernel the transformation is a smoothed inverse function.

In using a parametric approach one faces the difficult problem of choosing an appropriate family of spectral transformations. For many familes the number of degrees of freedom in the parameterization may be insufficient to accurately model the data. In this paper we propose an effective nonparametric method to find an optimal spectral transformation using kernel alignment. The main advantage of using kernel alignment is that it gives us a convex optimization problem, and does not suffer from poor convergence to local minima. A key assumption of a spectral transformation is monotonicity, so that unsmooth functions over the data graph are penalized more severly. We realize this property by imposing order constraints. The optimization problem in general is solved using semi-definite programming (SDP) [1]; however, in our approach the problem can be formulated in terms of quadratically constrained quadratic programming (QCQP), which can be solved more efficiently than a general SDP.

This paper is structured as follows. In Section 2 we review some graph theoretic concepts and relate them to the construction of kernels for semi-supervised learning. In Section 3 we introduce convex optimization via QCQP and relate it to the more familiar linear and quadratic programming commonly used in machine learning. Section 4 poses the problem of kernel based semi-supervised learning as a QCQP problem with order constraints. Experimental results using the proposed optimization framework are presented in Section 5. The results indicate that the semi-supervised kernels constructed from the learned spectral transformations perform well in practice.

2 Semi-supervised Kernels from Graph Spectrum

We are given a labeled dataset consisting of input-output pairs $\{(x_1, y_1), \ldots, (x_l, y_l)\}$ and a (typically much larger) unlabeled dataset $\{x_{l+1}, \ldots, x_n\}$ where x is in some general input space and y is potentially from multiple classes. Our objective is to find a kernel that is suitable for semi-supervised learning. More specifically, we are interested in a *transductive* approach where the unlabeled data also serve as the test data. As such, we only need to find a good Gram matrix on the points $\{x_1, \ldots, x_n\}$. For this approach to be effective such kernel matrices must also take into account the distribution of unlabeled data, in order that the unlabeled data can aid in the classification task. Once these kernel matrices have been constructed, they can be deployed in standard kernel methods, for example support vector machines, for semi-supervised learning.

In this paper we motivate the construction of semi-supervised kernel matrices from a graph theoretic perspective. A graph is constructed where the nodes are the data instances $\{1, \ldots, n\}$ and an edge connecteds nodes i and j if a "local similarity" measure between x_i and x_j suggests they may have the same label. For example, the local similarity measure can be the Euclidean distance between feature vectors if $x \in \mathbb{R}^m$, and each node can connect to its k nearest neighbors with weight value equal to 1. The intuition underlying the graph is that even if two nodes are not directly connected, they should be considered similar as long as there are dense paths between them. Several semi-supervised learning methods have been proposed under the general graph theoretic theme, including random walks [8], diffusion kernels [5], and Gaussian fields [9]. Many of these methods can be unified into the regularization framework proposed by [7], which forms the basis of this paper.

The graph can be represented by an $n \times n$ weight matrix W with w_{ij} being the edge weight between node i, j, with $w_{ij} = 0$ if there is no edge. We require the entries of W to be non-negative, and assume that it forms a symmetric matrix; it is not necessary for W itself to be positive semi-definite. In semi-supervised learning W is an essential quantity, we assume it is provided by domain experts, and hence do not study its construction. Let D be a diagonal matrix where $d_{ii} = \sum_j w_{ij}$ is the degree of node i. This allows us to define the *combinatorial graph Laplacian* L as $L = D - W$ (the normalized Laplacian $\tilde{L} = D^{-1/2} L D^{-1/2}$ can be used as well). We denote L's eigensystem by $\{\lambda_i, \phi_i\}$, $L = \sum_{i=1}^n \lambda_i \phi_i \phi_i^\top$ where we assume the eigenvalues are sorted in non-decreasing order. L has many interesting properties [3]; for instance, it is always positive semi-definite, even if W is not. L has k zero eigenvalues $\lambda_1 = \cdots = \lambda_k = 0$, where k is the number of connected subgraphs. The corresponding eigenvectors $\phi_1, \ldots, \phi_k$ are constant over individual subgraphs and zero elsewhere. Perhaps the most important property of the Laplacian related to semi-supervised learning is the following: a smaller λ corresponds to a smoother (small $\sum_{ij} w_{ij}(\phi(i) - \phi(j))^2$) eigenvector ϕ over the graph. In a physical system the smoother eigenvectors correspond to the major vibration modes. Assuming the graph structure is correct, from a regularization perspective we want to encourage smooth functions, to reflect our belief that labels vary smoothly over the graph. Specifically, [2] and [7] suggest a general principle for creating semi-supervised kernel K from the graph Laplacian L: transform the eigenvalues λ into $r(\lambda)$, where the *spectral transformation* r is a non-negative and decreasing function[1]

$$K = \sum_{i=1}^{n} r(\lambda_i) \phi_i \phi_i^\top \tag{1}$$

Intuitively r reverses the order of the eigenvalues, so that smooth ϕ_i's have larger eigenvalues in K. A function (soft labels) $f = \sum c_i \phi_i$ over the graph are regularized in a kernel machine by a function of the RKHS norm $\Omega(||f||_K^2) = \Omega(\sum c_i^2 / r(\lambda_i))$. Since r is decreasing, unsmooth components in f are penalized more. In previous work r has often been chosen from a parametric family. For example, the diffusion kernel [5] corresponds to $r(\lambda) = \exp(-\frac{\sigma^2}{2}\lambda)$ and the Gaussian field kernel [10] corresponds to $r(\lambda) = \frac{1}{\lambda+\epsilon}$, and cross validation was used to find the hyperparameters σ or ϵ. Although the general principle of [7] is illustrative, it does not address the question of which parametric family to use for r. Moreover, the number of degrees of freedom (or the number of hyperparameters) may not suit the task at hand, resulting in overly constrained kernels. We address these limitations using a convex optimization approach by imposing an ordering constraint on r but otherwise not assuming any parametric form for the kernels.

3 Convex Optimization using QCQP

Let $K_i = \phi_i \phi_i^\top, i = 1 \cdots n$ be the outer product matrices of the eigenvectors. The semi-supervised kernel K is a linear combination of K_i's $K = \sum_{i=1}^{n} \mu_i K_i$ We formulate the problem of finding the spectral transformation as one that finds the interpolation coefficients $\{r(\lambda_i) = \mu_i\}$ by optimizing some convex objective function on K. To maintain the positive semi-definiteness constraint on K, one in general needs to invoke SDPs [1]. Semi-definite optimization can be described as the problem of optimizing a linear function of a symmetric matrix subject to linear equality constraints and the condition that the matrix be positive semi-definite. The well-known linear programming problem can be generalized to a semi-definite optimization by replacing the vector of variables with a symmetric matrix, and replacing the non-negativity constraints with a positive semi-definite constraints. This generalization inherits several properties: it is convex, has a rich duality theory and allows theoretically efficient solution algorithms based on iterating interior point methods to either follow a central path or decrease a potential function. However, a limitation of the SDPs is their computational complexity [1] and as such SDPs have only been applied to small scale problems [6]. An important special case of SDPs are those of *quadratically constrained quadratic programs* (QCQP) which are computationally more efficient that SDPs. Here both the objective function and the constraints are quadratic as illustrated below,

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{2} x^\top P_0 x + q_0^\top x + r_0 && (2)\\
\text{subject to} \quad & \frac{1}{2} x^\top P_i x + q_i^\top x + r_i \succeq 0 \quad\quad i = 1 \cdots m && (3)\\
& Ax = b && (4)
\end{aligned}$$

where $P_i \in \mathcal{S}_+^n,\ i = 1, \ldots, m$, where $\mathcal{S}_+^n$ defines the set of square symmetric positive semi-definite matrices. In a QCQP, we minimize a convex quadratic function over a feasible

[1] We use a slightly different notation where r is the inverse of that in [7].

region that is the intersection of ellipsoids. The required number of iterations to reach the optimal solution remains similar to those for linear programs making the approach feasible for large datasets. However, as observed by [1] not all SDPs can be relaxed to QCQPs. For the semi-supervised kernel learning task presented here solving an SDP would be computationally infeasible.

Recent work by [4] [6] has proposed *kernel target alignment* that can be used not only to assess the relationship between the feature spaces generated by two different kernels, but also to assess the similarity between spaces induced by a kernel and that induced by the labels themselves. Desirable properties of the alignment measure can be found in [4]. Importantly for this work its optimization can be formulated as a QCQP. The objective function is the empirical kernel alignment score:

$$\hat{A}(K_{tr}, T) = \frac{\langle K_{tr}, T \rangle_F}{\sqrt{\langle K_{tr}, K_{tr} \rangle_F \langle T, T \rangle_F}} \tag{5}$$

where K_{tr} is the kernel matrix restricted to the training points; $\langle M, N \rangle_F$ denotes the Frobenius product between two square matrices $\langle M, N \rangle_F = \sum_{ij} m_{ij} n_{ij} = Tr(MN^\top)$; and T is the target matrix on training data, with entry T_{ij} set to $+1$ if $y_i = y_j$ and -1 otherwise. Note for binary $\{+1, -1\}$ training labels $\mathbf{y}$ this is simply the rank one matrix $T = \mathbf{y}\mathbf{y}^\top$. K is guaranteed to be positive semi-definite by constraining $\mu_i \geq 0$. Previous work using kernel alignment did not take into account that the K_i's were derived from the graph Laplacian with the goal of semi-supervised learning. As such the μ_i's can take arbitrary values and there is no preference to penalize unsmooth component over the graph. This can be rectified by requiring smoother eigenvectors to receive larger coefficients, as shown in the next section.

4 Semi-Supervised Kernels with Order Constraints

As stated above, we would like to maintain a decreasing order on the spectral transformation $\mu_i = r(\lambda_i)$ to encourage smooth functions over the graph. This motivates the set of *order constraints*

$$\mu_i \geq \mu_{i+1}, \qquad i = 1 \cdots n-1 \tag{6}$$

And we can specify the desired semi-supervised kernel as follows.

Definition 1 *An* **order constrained semi-supervised kernel** K *is the solution to the following convex optimization problem:*

$$\begin{aligned}
max_K \qquad & \hat{A}(K_{tr}, T) && (7)\\
subject\ to \qquad & K = \textstyle\sum_{i=1}^n \mu_i K_i && (8)\\
& \mu_i \geq 0 && (9)\\
& trace(K) = 1 && (10)\\
& \mu_i \geq \mu_{i+1}, \qquad i = 1 \cdots n-1 && (11)
\end{aligned}$$

where T is the training target matrix, $K_i = \phi_i \phi_i^\top$ and ϕ_i's are the eigenvectors of the graph Laplacian.

The formulation is an extension to [6] with order constraints, and with special components K_i's from the graph Laplacian. Since $\mu_i \geq 0$ and K_i's are outer products, K will automatically be positive semi-definite and hence a valid kernel matrix. The trace constraint is needed to fix the scale invariance of kernel alignment. It is important to notice the order constraints are convex, and as such the whole problem is convex. Let $vec(A)$ be the column vectorization of a matrix A. Define $M = \left[vec(K_{1,tr}) \cdots vec(K_{m,tr})\right]$, it is not hard to show that the problem can then be expressed as

$$\max_{\mu} \quad vec(T)^\top M\mu \tag{12}$$

$$\text{subject to} \quad ||M\mu|| \leq 1 \tag{13}$$

$$\mu_i \geq 0 \tag{14}$$

$$\mu_i \geq \mu_{i+1}, \quad i = 1 \cdots n-1 \tag{15}$$

The objective function is linear in μ, and there is a simple cone constraint, making it a quadratically constrained quadratic program (QCQP). An improvement of the above order constrained semi-supervised kernel can be obtained by studying the Laplacian eigenvectors with zero eigenvalues. For a graph Laplacian there will be k zero eigenvalues, if the graph has k connected subgraphs. The k eigenvectors are piecewise constant over individual subgraphs, and zero elsewhere. This is desirable when $k > 1$, with the hope that subgraphs correspond to different classes. However if $k = 1$, the graph is connected. The first eigenvector ϕ_1 is a constant vector. The corresponding K_1 is a constant matrix, and acts as a bias term. In this situation we do not want to impose the order constraint $\mu_1 \geq \mu_2$ on the constant bias term. Instead we let μ_1 vary freely during optimization.

Definition 2 *An* **improved order constrained semi-supervised kernel** K *is the solution to the same problem in Definition 1, but the order constraints (11) only apply to non-constant eigenvectors:*

$$\mu_i \geq \mu_{i+1}, \quad i = 1 \cdots n-1, \text{ and } \phi_i \text{ not constant} \tag{16}$$

In practice we do not need all n eigenvectors of the graph Laplacian, or equivalently all n K_i's. The first $m < n$ eigenvectors with the smallest eigenvalues empirically work well. Also note we could have used the fact that K_i's are from orthogonal eigenvectors ϕ_i, to further simplify the expression. However we neglect this observation, making it easier to incorporate other kernel components if necessary.

It is illustrative to compare and contrast the order constrained semi-supervised kernels to other semi-supervised kernels with different spectral transformation. We call the original kernel alignment solution in [6] a *maximal-alignment* kernel. It is the solution to Definition 1 without the order constraints (11). Because it does not have the additional constraints, it maximizes kernel alignment among all spectral transformation. The hyperparameters σ and ϵ of the Diffusion kernel and Gaussian fields kernel (described earlier) can be learned by maximizing the alignment score also, although the optimization problem is not necessarily convex. These kernels use different amount of information from the original Laplacian eigenvalues λ_i. The maximal-alignment kernels ignore λ_i altogether. The order constrained semi-supervised kernels only use the *order* of λ_i and ignore their actual values. The diffusion and Gaussian field kernels use the actual values. In terms of the degree of freedom in choosing the spectral transformation μ_i's, the maximal-alignment kernels are completely free. The diffusion and Gaussian field kernels are restrictive since they have an implicit parametric form and only one free parameter. The order constrained semi-supervised kernels incorporates desirable features from both approaches.

5 Experimental Results

We evaluate the order constrained kernels on seven datasets. **baseball-hockey** (1993 instances / 2 classes), **pc-mac** (1943/2) and **religion-atheism** (1427/2) are document categorization tasks taken from the 20-newsgroups dataset. The distance measure is the standard cosine similarity between tf.idf vectors. **one-two** (2200/2), **odd-even** (4000/2) and **ten digits** (4000/10) are handwritten digits recognition tasks. **one-two** is digits "1" vs. "2"; **odd-even** is the artificial task of classifying odd "1, 3, 5, 7, 9" vs. even "0, 2, 4, 6, 8" digits, such that each class has several well defined internal clusters; **ten digits** is 10-way classification. **isolet** (7797/26) is isolated spoken English alphabet recognition from the UCI repository. For these datasets we use Euclidean distance on raw features. We use 10NN unweighted graphs on all datasets except isolet which is 100NN. For all datasets, we use the smallest $m = 200$ eigenvalue and eigenvector pairs from the graph Laplacian. These values are set arbitrarily without optimizing and do not create a unfair advantage to the proposed kernels. For each dataset we test on five different labeled set sizes. For a given labeled set size, we perform 30 random trials in which a labeled set is randomly sampled from the whole dataset. All classes must be present in the labeled set. The rest is used as unlabeled (test) set in that trial. We compare 5 semi-supervised kernels (improved order constrained kernel, order constrained kernel, Gaussian field kernel, diffusion kernel[2] and maximal-alignment kernel), and 3 standard supervised kernels (RBF (bandwidth learned using 5-fold cross validation),linear and quadratic). We compute the spectral transformation for order constrained kernels and maximal-alignment kernels by solving the QCQP using standard solvers (SeDuMi/YALMIP). To compute accuracy we use a standard SVM with the bound on slack variables arbitrarily set at $C = 100$ for all tasks and kernels. For multiclass classification we perform one-against-all and pick the class with the largest margin. We observe that test accuracies of the 5 semi-supervised kernels can be improved by scaling them before being fed into the SVM. The scaling (which does not change the alignment scores) is done so that the average diagonal element is one: $K \leftarrow \frac{n}{trace(K)} K$. To make the comparisons consistent, we perform the scaling on the supervised kernels as well, but note that the performance of RBF, linear and quadratic kernels with scaling is about the same as without scaling.

The results [3] are shown in Table 1, which has two rows for each cell: The upper row is the average *test set accuracy* with one standard deviation; The lower row is the average *training set kernel alignment*, and in parenthesis the average *run time in seconds* for QCQP on a 2.4GHz Linux computer. Each number is averaged over 30 random trials. To assess the statistical significance of the results, we perform paired t-test on test accuracy. We highlight the best accuracy in each row, and those that can not be determined as different from the best, with paired t-test at significance level 0.05. The semi-supervised kernels tend to outperform standard supervised kernels. The improved order constrained kernels are consistently among the best. Figure 1 shows the spectral transformation μ_i of the semi-supervised kernels for different tasks. These are for the 30 trials with the largest labeled set size in each task. The x-axis is in increasing order of λ_i (the original eigenvalues of the Laplacian). The mean (thick lines) and ± 1 standard deviation (dotted lines) of only the top 50 μ_i's are plotted for clarity. The μ_i values are scaled vertically for easy comparison among kernels. As expected the maximal-alignment kernels' spectral transformation is zigzagged, diffusion and Gaussian field's are very smooth, while order constrained kernels' are in between. The order constrained kernels (green) have large μ_1 because of the order constraint. This seems to be disadvantageous — the spectral transformation tries to balance it out by increasing the value of other μ_i's so that the constant K_1's relative influence is smaller. On the other hand the improved order constrained kernels (black) allow μ_1 to be small. As a result the rest μ_i's decay fast, which is desirable.

6 Conclusions

We have proposed and compared a novel approach for semi-supervised kernel construction using convex optimization. The method incorporates order constraints, and the resulting convex optimization problem can be solved efficiently using a QCQP. In this work the *base* kernels were derived from the graph Laplacian, and no parametric form for the spectral transformation was imposed, making the approach more general than previous approaches. Experiments show that the method is both computationally feasible and results in improvements to classification performance when used with support vector machines.

[2]The hyperparameters σ^2 and ϵ are learned with the $fminbnd()$ function in Matlab.

[3]Results on baseball-hockey and odd-even are similar and omitted for space. Full results can be found at http://www.cs.cmu.edu/∼zhuxj/pub/ocssk.pdf

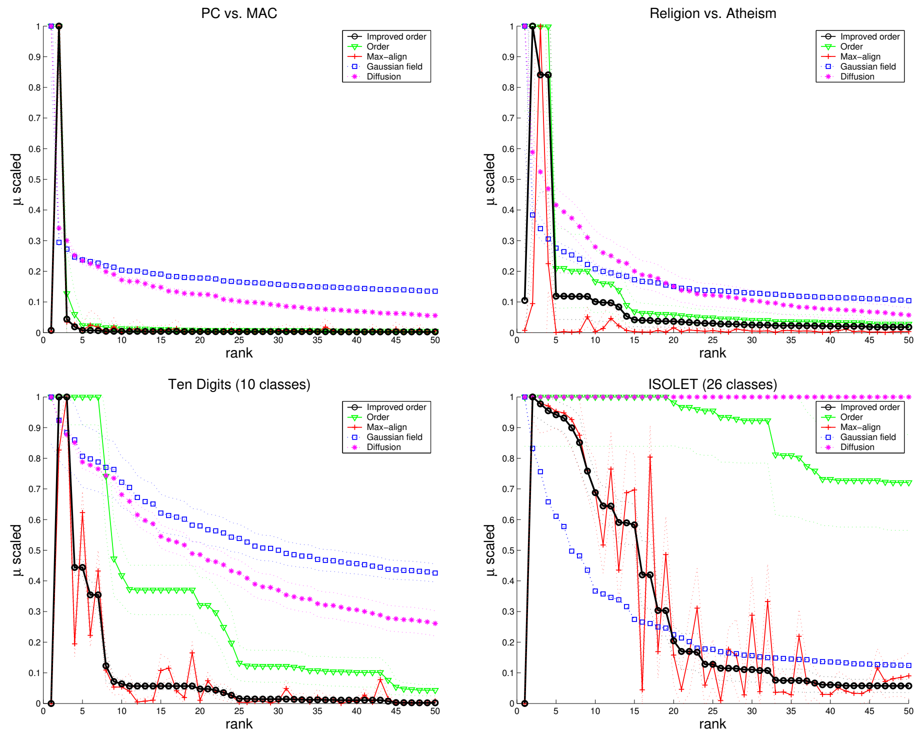

Figure 1: Comparison of spectral transformation for the 5 semi-supervised kernels.

References

[1] S. Boyd and L. Vandenberge. *Convex Optimization*. Cambridge University Press, Cambridge UK, 2004.

[2] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems, 15*, volume 15, 2002.

[3] F. R. K. Chung. *Spectral graph theory, Regional Conference Series in Mathematics, No. 92*. American Mathematical Society, 1997.

[4] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *Advances in NIPS*, 2001.

[5] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proc. 19th International Conf. on Machine Learning*, 2002.

[6] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[7] A. Smola and R. Kondor. Kernels and regularization on graphs. In *Conference on Learning Theory, COLT/KW*, 2003.

[8] M. Szummer and T. Jaakkola. Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems, 14*, volume 14, 2001.

[9] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML-03, 20th International Conference on Machine Learning*, 2003.

[10] X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: From Gaussian fields to Gaussian processes. Technical Report CMU-CS-03-175, Carnegie Mellon University, 2003.

	semi-supervised kernels					standard kernels		
Training set size	Improved Order	Order	Gaussian Field	Diffusion	Max-align	RBF	Linear	Quadratic
pc-mac								
10	**87.6** ± 4.1	84.9 ± 7.2	57.4 ± 7.0	69.8 ±13.4	71.1 ± 9.7	51.5 ± 4.0	53.1 ± 5.4	50.1 ± 0.6
	0.71 (1)	0.57 (2)	0.32	0.35	0.90 (2)	0.14	0.29	0.25
30	**90.2** ± 1.9	89.5 ± 2.4	76.7 ± 5.9	86.4 ± 3.4	84.7 ± 4.2	64.4 ± 8.4	61.1 ± 7.4	50.6 ± 1.0
	0.68 (9)	0.49 (12)	0.19	0.23	0.74 (10)	0.04	0.18	0.13
50	**90.5** ± 1.4	**90.3** ± 1.7	80.1 ± 4.4	87.5 ± 2.4	86.3 ± 3.5	68.9 ± 8.5	70.7 ± 7.0	53.0 ± 3.7
	0.64 (34)	0.46 (42)	0.16	0.20	0.68 (34)	0.02	0.14	0.10
70	**90.6** ± 1.5	**90.6** ± 1.5	83.6 ± 2.1	88.7 ± 1.6	86.6 ± 2.7	73.9 ± 8.0	74.8 ± 7.2	53.9 ± 5.7
	0.63 (69)	0.46 (88)	0.14	0.19	0.66 (74)	0.02	0.12	0.09
90	**90.9** ± 1.4	**90.8** ± 1.4	85.1 ± 2.7	89.3 ± 2.1	87.1 ± 2.7	78.7 ± 5.6	78.6 ± 6.7	53.6 ± 3.9
	0.63 (108)	0.45 (146)	0.13	0.18	0.65 (121)	0.01	0.11	0.07
religion-atheism								
10	**72.7** ±11.2	70.9 ±10.9	59.5 ± 8.3	62.3 ±10.6	60.8 ± 7.5	53.6 ± 5.8	55.6 ± 5.5	51.6 ± 5.8
	0.50 (2)	0.42 (2)	0.31	0.32	0.85 (2)	0.08	0.30	0.26
30	**83.2** ± 3.1	82.2 ± 3.2	72.6 ± 5.2	79.2 ± 4.5	72.4 ± 5.4	65.3 ± 6.8	63.8 ± 6.7	53.0 ± 5.2
	0.38 (8)	0.31 (11)	0.20	0.22	0.60 (10)	0.04	0.19	0.15
50	**83.3** ± 2.7	82.4 ± 3.4	79.9 ± 4.2	81.8 ± 3.7	74.3 ± 6.0	71.0 ± 6.3	70.4 ± 6.8	55.1 ± 6.8
	0.31 (30)	0.26 (36)	0.17	0.20	0.48 (38)	0.03	0.15	0.12
70	**84.2** ± 1.9	83.9 ± 2.0	80.9 ± 3.2	**83.9** ± 2.3	78.0 ± 3.9	73.6 ± 6.7	73.8 ± 5.9	57.2 ± 4.0
	0.29 (62)	0.25 (77)	0.16	0.19	0.43 (76)	0.04	0.13	0.10
90	**84.4** ± 2.6	**84.2** ± 2.4	81.0 ± 2.8	83.3 ± 3.1	78.1 ± 3.9	78.0 ± 4.0	78.2 ± 4.9	60.6 ± 8.5
	0.27 (101)	0.24 (130)	0.15	0.18	0.40 (116)	0.03	0.12	0.09
one-two								
10	**95.9** ± 2.9	**95.2** ± 3.9	86.8 ±13.7	93.1 ± 7.4	84.9 ±12.3	87.5 ± 8.0	85.8 ± 8.4	78.7 ± 9.9
	0.87 (1)	0.66 (2)	0.43	0.53	0.95 (2)	0.39	0.27	0.31
20	**97.4** ± 2.6	**97.3** ± 2.7	96.6 ± 3.4	**97.6** ± 2.7	94.4 ± 2.5	92.4 ± 3.1	91.7 ± 2.7	89.2 ± 5.1
	0.87 (4)	0.64 (5)	0.38	0.50	0.90 (5)	0.33	0.22	0.27
30	98.8 ± 1.8	**98.9** ± 1.7	**98.9** ± 1.5	**99.1** ± 1.6	97.4 ± 2.4	94.9 ± 2.1	94.2 ± 2.7	91.4 ± 3.4
	0.84 (9)	0.61 (12)	0.35	0.47	0.86 (10)	0.29	0.18	0.22
40	**99.3** ± 1.0	**99.3** ± 1.0	99.1 ± 1.3	**99.2** ± 1.0	98.1 ± 1.9	96.0 ± 2.6	95.1 ± 2.0	94.1 ± 2.8
	0.84 (17)	0.61 (26)	0.36	0.48	0.86 (19)	0.29	0.17	0.21
50	**99.4** ± 0.3	**99.3** ± 0.6	**99.4** ± 0.3	**99.4** ± 0.9	97.9 ± 1.9	97.0 ± 1.8	95.3 ± 2.3	95.1 ± 3.2
	0.83 (35)	0.60 (47)	0.35	0.46	0.84 (36)	0.27	0.16	0.20
Ten digits (10 classes)								
50	**76.5** ± 4.4	71.4 ± 5.1	42.9 ± 5.8	55.1 ± 8.3	70.0 ± 5.1	58.0 ± 3.5	60.1 ± 3.1	52.5 ± 4.0
	0.47 (30)	0.21 (30)	0.15	0.16	0.51 (33)	-0.64	-0.50	-0.25
100	**84.1** ± 2.2	82.7 ± 2.8	64.3 ± 3.4	75.0 ± 3.8	80.1 ± 3.1	69.8 ± 1.7	69.0 ± 2.1	66.1 ± 2.4
	0.47 (133)	0.17 (118)	0.12	0.13	0.49 (140)	-0.65	-0.52	-0.30
150	**86.3** ± 1.7	**85.8** ± 1.7	75.0 ± 3.1	82.0 ± 3.1	84.2 ± 2.4	75.5 ± 1.4	74.5 ± 1.5	73.2 ± 1.7
	0.48 (327)	0.18 (271)	0.11	0.13	0.50 (334)	-0.65	-0.53	-0.32
200	**87.9** ± 1.8	**87.6** ± 2.1	79.7 ± 2.8	84.9 ± 2.5	86.4 ± 2.2	78.8 ± 1.1	77.2 ± 1.1	76.7 ± 1.5
	0.47 (607)	0.16 (530)	0.10	0.11	0.49 (597)	-0.66	-0.54	-0.32
250	**88.9** ± 1.4	**89.0** ± 1.5	83.4 ± 2.0	87.1 ± 1.7	87.9 ± 1.5	80.7 ± 1.2	78.4 ± 1.1	79.1 ± 1.5
	0.47 (918)	0.16 (844)	0.10	0.11	0.49 (895)	-0.66	-0.54	-0.33
isolet (26 classes)								
50	**56.2** ± 2.8	41.9 ± 5.2	41.4 ± 2.9	30.5 ± 4.3	50.2 ± 3.7	28.2 ± 2.7	30.1 ± 2.4	25.7 ± 2.6
	0.27 (36)	0.13 (31)	0.03	0.11	0.31 (30)	-0.89	-0.80	-0.66
100	**64.8** ± 2.2	59.0 ± 3.6	58.6 ± 2.9	49.6 ± 3.4	63.2 ± 1.9	46.3 ± 2.0	47.1 ± 2.5	43.7 ± 2.5
	0.26 (152)	0.10 (130)	-0.02	0.08	0.29 (127)	-0.90	-0.82	-0.69
150	**68.1** ± 2.5	64.8 ± 3.2	65.3 ± 2.6	58.0 ± 2.7	**67.7** ± 2.6	56.3 ± 1.8	57.3 ± 2.5	54.7 ± 2.3
	0.26 (357)	0.09 (300)	-0.05	0.07	0.27 (318)	-0.90	-0.83	-0.71
200	**72.0** ± 2.0	70.4 ± 2.2	70.4 ± 1.9	64.5 ± 2.1	**71.9** ± 1.7	64.1 ± 1.7	64.5 ± 1.8	61.8 ± 1.5
	0.26 (975)	0.08 (615)	-0.07	0.06	0.27 (748)	-0.91	-0.83	-0.71
250	73.5 ± 1.6	73.4 ± 1.7	73.5 ± 1.2	69.4 ± 1.6	**74.3** ± 1.4	69.3 ± 1.4	68.4 ± 1.7	67.5 ± 1.3
	0.26 (1307)	0.08 (959)	-0.07	0.06	0.27 (1118)	-0.91	-0.84	-0.72

Table 1: Accuracy, alignment scores, and run times on the datasets. The table compares 8 kernels. Each cell has two rows: The upper row is test set accuracy with standard error; The lower row is training set alignment (in parenthesis QCQP run time in seconds). All numbers are averaged over 30 random trials. Highlighted accuracies are the best ones as determined by paired t-test at 0.05 significance level.