LEARNING SEQUENCE KERNELS

Corinna Cortes

Google Research
76 Ninth Avenue
New York, NY 10011
corinna@google.com

*Mehryar Mohri**

Courant Institute and
Google Research
New York, NY 10012
mohri@cims.nyu.edu

*Afshin Rostamizadeh**

Courant Institute and
Google Research
New York, NY 10012
rostami@cs.nyu.edu

ABSTRACT

Kernel methods are used to tackle a variety of learning tasks including classification, regression, ranking, clustering, and dimensionality reduction. The appropriate choice of a kernel is often left to the user. But, poor selections may lead to a sub-optimal performance. Instead, sample points can be used to learn a kernel function appropriate for the task by selecting one out of a family of kernels determined by the user. This paper considers the problem of *learning sequence kernel functions*, an important problem for applications in computational biology, natural language processing, document classification and other text processing areas. For most kernel-based learning techniques, the kernels selected must be positive definite symmetric, which, for sequence data, are found to be rational kernels. We give a general formulation of the problem of learning rational kernels and prove that a large family of rational kernels can be learned efficiently using a simple quadratic program both in the context of support vector machines and kernel ridge regression. This improves upon previous work that generally results in a more costly semi-definite or quadratically constrained quadratic program. Furthermore, in the specific case of kernel ridge regression, we give an alternative solution based on a closed-form solution for the optimal *kernel matrix*. We also report results of experiments with our kernel learning techniques in classification and regression tasks.

1. INTRODUCTION

Kernel methods are widely used in statistical learning techniques due to the computational efficiency and the flexibility they offer [1, 2]. Using a positive definite symmetric kernel, the input data is implicitly embedded in a high-dimensional feature space where linear methods can be used for learning and estimation. These methods have been used to tackle a variety of learning tasks in classification, regression, ranking, clustering, dimensionality reduction, and other areas. In particular, kernels for sequences have been successfully used in combination with support vector machines (SVMs) [3–5] and other discriminative algorithms in a variety of applications in computational biology, natural language processing, and document classification [6–12].

Any positive definite symmetric (PDS) kernel can be used within these techniques and the choice is typically left to the user. But this choice is critical to the success of the learning algorithms. Poor selections may not capture certain important aspects of the task and lead to a sub-optimal performance. Instead, sample points can be used to *learn a kernel* appropriate for the task by selecting one out of a family of kernels determined by the user.

The problem of learning kernels has been investigated in several previous studies, primarily focusing on learning a linear combination of kernels. Lanckriet et al. [13] examined this problem in a *transductive learning* scenario where the learner is given the test points [13]. In that case, the problem reduces to that of learning a *kernel matrix*. They showed that this problem can be cast as a semi-definite programming problem (SDP) when using objective functions such as the hard- or soft-margin SVMs, and analyzed more specifically the case of linear combinations of kernel matrices based on pre-specified kernels, in which case the optimization problem can be cast as a quadratically constrained quadratic programming (QCQP) problem. This optimization problem has also been recently studied by [14] and solved using interior point methods. Ong et al. considered instead the problem of learning a *kernel function* from a set of kernels that are in a Hilbert space of functions generated by a so-called hyper-kernel, which includes convex combinations of potentially infinitely many kernels [15]. Micchelli and Pontil [16] also examined the problem of learn-

*This work was partially funded by the New York State Office of Science Technology and Academic Research (NYSTAR) and was also sponsored in part by the Department of the Army Award Number W23RYX-3275- N605. The U.S. Army Medical Research Acquisition Activity, 820 Chandler Street, Fort Detrick MD 21702- 5014 is the awarding and administering acquisition office. The content of this material does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

ing a kernel function, when it is a convex combination of kernels parameterized by a compact set, for the square loss regularization. Argyriou et al. [17] extended these results to other losses and further provided a formulation of the problem as a difference of convex (DC) program [18]. Several other variants of the problem dealing with multi-task or multi-class problems have also been studied [19–21].

This paper considers the problem of learning sequence kernel functions, an important problem for sequence learning applications in computational biology, natural language processing, document classification and other text processing areas. According to [11], the sequence kernels in all of these applications are *rational kernels*. Thus, we will examine more specifically the problem of learning rational kernels. We give a general formulation of the problem of learning rational kernels and prove that, remarkably, a large family of rational kernels, count-based kernels, can be learned efficiently using a simple quadratic program (QP) both with the objective function of SVMs and that of kernel ridge regression (KRR) [22]. Count-based rational kernels include many kernels used in computational biology and text classification. We also report the results of experiments with our sequence learning techniques in both classification and regression tasks.

The remainder of this paper is organized as follows. Section 2 introduces the definition of weighted transducers and rational kernels and points out some important properties of positive definite symmetric kernels. Section 3 gives a general formulation of the problem of learning rational kernels. In Section 4, we show that the problem of learning count-based kernels can be reduced to a simple QP both in the case of the SVMs and KRR objective functions. For KRR, we further describe in Section 4.4 an an alternative solution based on a closed-form solution for the optimal kernel matrix. Section 5 reports the results of our experiments with learning count-based rational kernels in both classification and regression tasks.

2. PRELIMINARIES

This section introduces the definition of rational kernels and their main properties, which we will use in our formulation of the learning problem. We will follow the definitions and terminology of [11]. The representation and computation of rational kernels is based on *weighted finite-state transducers*.

2.1. Weighted transducers

Weighted finite-state transducers are finite automata such that each transition is augmented with an output label in addition to the familiar input label and some real-valued weight that may represent a cost or a probability [23]. Input (output) labels are concatenated along a path to form an input (output) sequence. The weights of the transducers considered here are non-negative real values.

Figure 1(a) shows an example of a weighted finite-state transducer with the same input and output alphabet. A path from an initial state to a final state is an accepting path and its weight is obtained by multiplying the weights of its constituent transitions and the weight of the final state, which is displayed after the slash in the figure. We will assume a common alphabet Σ for the input and output symbols and will denote by ϵ the empty string or null symbol. The weight associated by a weighted transducer T to a pair of strings $(x, y) \in \Sigma^* \times \Sigma^*$ is denoted by $T(x, y)$ and is obtained by summing the weights of all accepting paths with input label x and output label y. The transducer T of Figure 1(a) associates to the pair (abb, bab) the weight $T(abb, bab) = .1 \times .3 \times .5 \times 1 + .2 \times .4 \times .5 \times 1$, since it admits two paths with input label abb and output label bab.

For any transducer T, T^{-1} denotes its *inverse*, that is the transducer obtained from T by swapping the input and output labels of each transition. Thus, for all $x, y \in \Sigma^*$, we have $T^{-1}(x, y) = T(y, x)$. The *composition* of two weighted transducers T_1 and T_2 with matching input and output alphabets Σ is a weighted transducer denoted by $T_1 \circ T_2$ and for all $x, y \in \Sigma^*$ defined by:

$$(T_1 \circ T_2)(x, y) = \sum_{z \in \Sigma^*} T_1(x, z)\, T_2(z, y), \qquad (1)$$

when the sum is well-defined and in $\mathbb{R}_+ \cup \{+\infty\}$ [23]. Note that $T(x, y)$ is the sum of the weights of all the accepting paths of $X \circ T \circ Y$, where X and Y are acceptors of the strings x and y with weight one. There is an efficient algorithm for computing the composition of two weighted transducers T_1 and T_2 in time $O(|T_1||T_2|)$, where $|T_1|$ is the size of T_1 and $|T_2|$ that of T_2 [11].

2.2. Rational Kernels

A sequence kernel $K : \Sigma^* \times \Sigma^* \mapsto \mathbb{R}$ is *rational* if it coincides with the function defined by a weighted transducer U, that is if $K(x, y) = U(x, y)$ for all $x, y \in \Sigma^*$. Not all rational kernels are *positive definite and symmetric* (PDS), or equivalently verify the Mercer condition, which is crucial for the convergence of training for discriminant algorithms such as SVMs. The following is a key theorem of [11] that will guide our formulation of the problem of learning PDS rational kernels.

Theorem 1 ([11]). *Let T be an arbitrary weighted transducer. Then, the function defined by the transducer $U = T \circ T^{-1}$ is a PDS rational kernel.*

Furthermore, the rational kernels used in computational biology and natural language processing problems such as

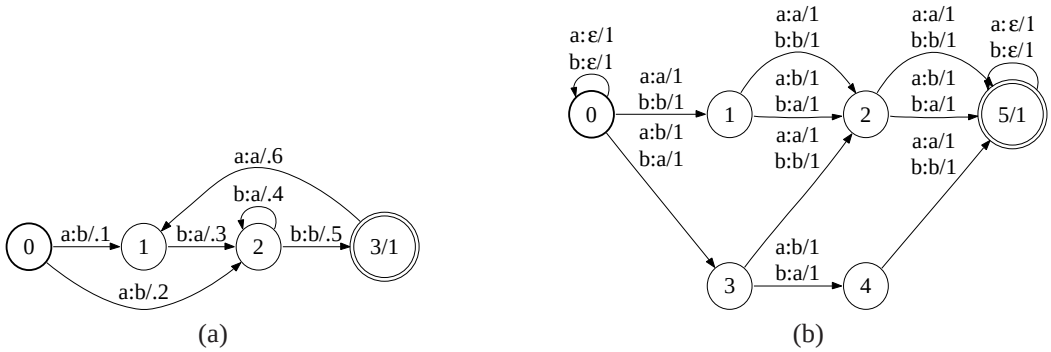


Fig. 1. (a) Example of a weighted transducer. The initial state is indicated by a bold circle, a final state by a double circle. Input and output labels are separated by a colon and the weight indicated after the slash separator. (b) Transducer T defining the mismatch kernel $T \circ T^{-1}$ [7, 11].

[6, 8, 10, 12, 24] are all of this form and it has been conjectured that in fact this represents all PDS rational kernels [11]. Thus, in what follows, we will refer by *PDS rational kernels* to the rational kernels K defined by a transducer $U = T \circ T^{-1}$. To ensure that the finiteness of the kernel values, we will also assume that T does not admit any cycle with input ϵ. This implies that for any $x \in \Sigma^*$, there are finitely many sequences $z \in \Sigma^*$ for which $T(x, z) \neq 0$.

3. PROBLEM FORMULATION

We consider the standard supervised learning setting where the learning algorithm receives a sample of m labeled points $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times Y)^m$, where X is the input space and Y the set of labels, $Y = \mathbb{R}$ in the regression case, $Y = \{-1, +1\}$ in the classification case.

We will formulate the problem in the case of SVMs. The discussion for other objective functions is similar. Let $\mathcal{K}$ represent a family of PDS rational kernels. We wish to select a kernel function $K \in \mathcal{K}$ that minimizes the generalization error of the SVM predictor. Following the structural risk minimization principle [5], the kernel should be selected by minimizing an objective function corresponding to a bound on the generalization error.

Let $\{\mathbf{K} \in \mathbb{R}^{m \times m}\}$ denote the kernel matrix of the kernel function K restricted to the sample S, $\mathbf{K}_{ij} = K(x_i, x_j)$, for all $i, j \in [1, m]$, and let $\mathbf{Y} \in \mathbb{R}^{m \times m}$ denote the diagonal matrix of the labels, $\mathbf{Y} = \mathrm{diag}(y_1, \ldots, y_m)$. We will denote by $\mathbf{0}$ the column matrices in $\mathbb{R}^{m \times 1}$ with all its components equal to zero, and similarly by $\mathbf{C}$ the constant column matrix with all elements equal to C, where C is the trade-off parameter of the SVMs optimization problem. Then, using the dual form of the SVM optimization problem [4], the general optimization problem for learning kernels can be written as

$$\begin{aligned} \min_{K \in \mathcal{K}} \max_{\boldsymbol{\alpha}} \quad & 2\boldsymbol{\alpha}^\top \mathbf{1} - \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K} \mathbf{Y} \boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{y} = 0 \ \wedge \ \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \\ & K \succeq 0 \ \wedge \ \mathrm{Tr}[\mathbf{K}] = \Lambda, \end{aligned} \tag{2}$$

where $\boldsymbol{\alpha}_m \in \mathbb{R}^{m \times 1}$ denotes the column matrix of the dual variables α_i, $i \in [1, m]$ and $\Lambda \geq 0$ a parameter controlling the trace of the kernel matrix $\mathbf{K}$, a widely used constraint when learning kernels, see [13–17].

In general, this optimization leads to SDP programs, due to the condition on the positive-definiteness of K. However, this condition is not necessary when searching for kernels of the type $T \circ T^{-1}$ since by Theorem 1, they are PDS, regardless of the weighted transducer T used. For PDS rational kernels there exists a family of weighted transducers $\mathcal{T}$ such that $\mathcal{K} = \{T \circ T^{-1} : T \in \mathcal{T}\}$. Thus for this family of kernel functions, the optimization (2) corresponds to the problem of learning a weighted transducer. It is known that the general problem of learning minimal (unweighted) finite automata, or even a polynomial approximation, is NP-hard [25]. In our case of learning weighted transducers, this suggests some limitation on the choice of the family of transducers $\mathcal{T}$. We will restrict ourselves to learning the transition weights of a transducer. Therefore we will assume $\mathcal{T}$ to be a family of transducers with the same topology and same transition labels, but different transition weights.

By our definition of PDS rational kernels, for any x the set of sequences z such that $T(x, z) \neq 0$ is finite. Let $z_1, \ldots, z_p \in \Sigma^*$ be the finite set of sequences z such that $T(x_i, z) \neq 0$ for some $i \in [1, m]$ and let $\mathbf{T} \in \mathbb{R}^{m \times p}$ denote the matrix defined by $\mathbf{T}_{ij} = T(x_i, z_j)$. Then, our general optimization problem for learning rational kernels for the objective function of SVMs can be written as follows:

$$\begin{aligned} \min_{T \in \mathcal{T}} \max_{\boldsymbol{\alpha}} \quad & 2\boldsymbol{\alpha}^\top \mathbf{1} - (\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{T})(\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{T})^\top \\ \text{subject to} \quad & \boldsymbol{\alpha}^\top \mathbf{y} = 0 \ \wedge \ \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \ \wedge \ \|\mathbf{T}\|_F^2 = \Lambda, \end{aligned} \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The matrix coefficients $\mathbf{T}_{ij} = T(x_i, z_j)$ are obtained by summing the weights of all accepting paths of T with input label x_i and output label z_j. Thus, in general, they are polynomials over the transitions weights of the transducer T. The next section examines a general family of kernels for which this optimization admits an efficient solution.

4. ALGORITHMS FOR LEARNING RATIONAL KERNELS

This section shows that learning a large family of kernels, including count-based rational kernels, can be solved efficiently as a simple QP problem.

4.1. Count-Based Rational Kernels

Many kernels used in computational biology and text categorization problems are *count-based rational kernels*. This family of kernels includes the n-gram kernels used successfully in document classification [12] or spoken-dialog classification [11]. Count-based rational kernels map each sequence to a finite set of strings that may be substrings or subsequences of various lengths.

Figure 2(a) shows a transducer T corresponding to a bigram kernel that gives equal weight (one) to all bigrams aa, ab, ba, bb. The output label of the accepting paths of this transducer are precisely the set of possible bigrams. The transducer maps an input sequence u to the set of bigrams appearing in u. It further generates as many paths labeled with a given bigram z as there are occurrences of z in u. Since the weights of the paths are added, the kernel $T \circ T^{-1}$ associates to each pair (x, y) the sum of the products of the counts of their common bigrams. Figure 2(b) gives the general form of a count-based transducer. A is an arbitrary acyclic deterministic automaton. The transition labeled with $A{:}A/1$ is a short-hand for the acyclic transducer mapping each sequence of A to itself with weight one. In the case of the bigram kernel, A is a deterministic automaton accepting the set of bigrams. This transducer similarly counts the number of occurrences of any sequence z accepted by A and $T \circ T^{-1}(x, y)$ is the sum of the product of these counts in x and y.

We are interested in learning kernels of this type but with possibly different weights assigned to the sequences z accepted by A. These weights can serve to emphasize the importance of each sequence z in the similarity measure $T \circ T^{-1}$. Let w_k be the weight assigned to the sequence z_k accepted by A. Then, by definition, for any input string x, $T(x, z_k)$ is the product of w_k and the number of occurrences of z_k in x. Thus, for $i, j \in [1, m]$,

$$\begin{aligned}(T \circ T^{-1})(x_i, x_j) &= \sum_{k=1}^{p} T(x_i, z_k) T(x_j, z_k) \\ &= \sum_{k=1}^{p} w_k^2 \, |x_i|_k \, |x_j|_k,\end{aligned} \tag{4}$$

where $|x_i|_k$ denotes the number of occurrences of z_k in x_i, for $i \in [1, m]$ and $k \in [1, p]$. Let $\mathbf{X} \in \mathbb{R}^{m \times p}$ denote the matrix defined by $\mathbf{X}_{ik} = |x_i|_k$ for $i \in [1, m]$ and $k \in [1, p]$, and let $\mathbf{X}_k$, $k \in [1, p]$, denote the kth column of $\mathbf{X}$. Then, Equation 4 can be rewritten as

$$T \circ T^{-1} = \sum_{k=1}^{p} \mu_k \mathbf{X}_k \mathbf{X}_k^\top, \tag{5}$$

where $\mu_k = w_k^2$, for all $k \in [1, p]$. We will use this identity to present efficient solutions to the problem of learning count-based rational kernels with both the SVM and KRR objective functions.

4.2. Support Vector Machines

In the case of SVM, the optimization problem can be written as

$$\begin{aligned}&\min_{\boldsymbol{\mu}} \max_{\boldsymbol{\alpha}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = 2\boldsymbol{\alpha}^\top \mathbf{1} - \sum_{k=1}^{p} \mu_k \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y} \boldsymbol{\alpha} \\ &\text{subject to} \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ &\qquad\qquad \boldsymbol{\mu} \geq \mathbf{0} \wedge \sum_{k=1}^{p} \mu_k \|\mathbf{X}_k\|^2 = \Lambda,\end{aligned} \tag{6}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{m \times 1}$, and $\boldsymbol{\mu} \in \mathbb{R}^{p \times 1}$ denotes the column vector with components μ_k, $k \in [1, p]$. Note, that this is a convex optimization problem in $\boldsymbol{\mu}$ since F is affine and thus convex in $\boldsymbol{\mu}$, the pointwise maximum over $\boldsymbol{\alpha}$ of a convex function also defines a convex function [26], and the constraints are all convex. While we seek to learn a kernel function and not a kernel matrix, the optimization problem we have derived at this stage is similar to those obtained by [13]. However, due to the specific property (5), the problem reduces to a simple standard QP problem.

Let $\mathcal{M}$ denote the convex and compact set $\mathcal{M} = \{\boldsymbol{\mu} : \boldsymbol{\mu} \geq \mathbf{0} \wedge \sum_{k=1}^{p} \mu_k \|\mathbf{X}_k\|^2 = \Lambda\}$ and $\mathcal{A}$ the convex and compact set $\mathcal{A} = \{\mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0\}$. The function $\boldsymbol{\mu} \mapsto F(\boldsymbol{\mu}, \boldsymbol{\alpha})$ is convex with respect to $\boldsymbol{\mu}$ for any $\boldsymbol{\alpha}$. For any $\boldsymbol{\mu}$, the function $\boldsymbol{\alpha} \mapsto F(\boldsymbol{\mu}, \boldsymbol{\alpha})$ is concave since $\sum_{k=1}^{p} \mu_k \mathbf{Y}^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}$ is a positive definite symmetric matrix and F is a continuous function. Thus, by the von Neumann's generalized minimax theorem [27], the $\min$ and

Fig. 2. Count-based kernels for the alphabet $\Sigma = \{a, b\}$. (a) Bigram kernel transducer. (b) General count-based kernel transducer.

max can be transposed and the optimization (6) is equivalent to:

$$\min_{\boldsymbol{\mu}\in\mathcal{M}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\boldsymbol{\mu}\in\mathcal{M}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}). \tag{7}$$

Since the term $2\boldsymbol{\alpha}^\top \mathbf{1}$ does not depend on $\boldsymbol{\mu}$, this can be further written as

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\boldsymbol{\mu}\in\mathcal{M}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}) \\ &= \max_{\boldsymbol{\alpha}\in\mathcal{A}} 2\boldsymbol{\alpha}^\top \mathbf{1} - \max_{\boldsymbol{\mu}\in\mathcal{M}} \sum_{k=1}^{p} \mu_k (\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{X}_k)^2. \end{aligned} \tag{8}$$

Note that the terms within this last sum are all non-negative, thus the optimal solution is obtained by placing all the $\boldsymbol{\mu}$ weight on the largest $(\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{X}_k)^2$. Using this observation, and the constraint $\sum_{k=1}^{p} \mu_k \|\mathbf{X}_k\|^2 = \Lambda$, the optimization problem can be rewritten as

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in\mathcal{A}} 2\boldsymbol{\alpha}^\top \mathbf{1} - \Lambda \max_{k\in[1,p]} \left(\frac{\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{X}_k}{\|\mathbf{X}_k\|} \right)^2 \\ &= \max_{\boldsymbol{\alpha}\in\mathcal{A}} 2\boldsymbol{\alpha}^\top \mathbf{1} - \Lambda \max_{k\in[1,p]} (\boldsymbol{\alpha}^\top \mathbf{u}'_k)^2, \end{aligned} \tag{9}$$

where $\mathbf{u}'_k$ is the normalized column matrix $\mathbf{u}'_k = \frac{\mathbf{Y}^\top \mathbf{X}_k}{\|\mathbf{X}_k\|} = \frac{\mathbf{Y}^\top \mathbf{X}_k}{\|\mathbf{Y}^\top \mathbf{X}_k\|}$. This leaves us with the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}, t} \quad & -2\boldsymbol{\alpha}^\top \mathbf{1} + \Lambda t^2 \\ \text{subject to} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ & -t \leq \boldsymbol{\alpha}^\top \mathbf{u}'_k \leq t, \forall k \in [1, p]. \end{aligned} \tag{10}$$

Let $\mathbf{U}' \in \mathbb{R}^{m\times p}$ be the matrix whose kth column is $\mathbf{u}'_k$ and introduce the Lagrange variables $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathbb{R}^{p\times 1}$, $\boldsymbol{\eta}, \boldsymbol{\eta}' \in \mathbb{R}^{m\times 1}$ and $\delta \in \mathbb{R}$ to write the Lagrangian:

$$\begin{aligned} L(\boldsymbol{\alpha}, t, \boldsymbol{\beta}, \boldsymbol{\beta}', \boldsymbol{\eta}, \boldsymbol{\eta}', \delta) = &-2\boldsymbol{\alpha}^\top \mathbf{1} + \Lambda t^2 - \boldsymbol{\eta}^\top \boldsymbol{\alpha} + \boldsymbol{\eta}'^\top (\boldsymbol{\alpha} - \mathbf{C}) \\ &+ \delta \boldsymbol{\alpha}^\top \mathbf{y} - \boldsymbol{\beta}^\top (\mathbf{U}'^\top \boldsymbol{\alpha} + t\mathbf{1}) + \boldsymbol{\beta}'^\top (\mathbf{U}'^\top \boldsymbol{\alpha} - t\mathbf{1}). \end{aligned} \tag{11}$$

Differentiating with respect to the primal variables we observe that the following equalities hold at the optimum:

$$\begin{aligned} &\begin{cases} \nabla_t L = 2t\Lambda - (\boldsymbol{\beta} + \boldsymbol{\beta}')^\top \mathbf{1} = 0 \\ \nabla_{\boldsymbol{\alpha}} L = -2\mathbf{1} + \delta \mathbf{y} - \boldsymbol{\eta} + \boldsymbol{\eta}' \end{cases} \\ &\Leftrightarrow \begin{cases} t = \frac{1}{2\Lambda} (\boldsymbol{\beta} + \boldsymbol{\beta}')^\top \mathbf{1} + \mathbf{U}'(\boldsymbol{\beta}' - \boldsymbol{\beta}) = \mathbf{0} \\ \mathbf{U}'(\boldsymbol{\beta}' - \boldsymbol{\beta}) - 2\mathbf{1} + \delta \mathbf{y} - \boldsymbol{\eta} + \boldsymbol{\eta}' = \mathbf{0}. \end{cases} \end{aligned} \tag{12}$$

Plugging in the first equality in the Lagrangian and taking into account the second equality, we obtain the following equivalent dual optimization:

$$\begin{aligned} \max_{\boldsymbol{\beta}, \boldsymbol{\beta}', \boldsymbol{\eta}, \boldsymbol{\eta}', \delta} \quad & -\frac{1}{4\Lambda} (\boldsymbol{\beta}' + \boldsymbol{\beta})^\top (\mathbf{1}\mathbf{1}^\top)(\boldsymbol{\beta}' + \boldsymbol{\beta}) - \boldsymbol{\eta}'^\top \mathbf{C} \\ \text{subject to} \quad & \mathbf{U}'(\boldsymbol{\beta}' - \boldsymbol{\beta}) + (\boldsymbol{\eta}' - \boldsymbol{\eta}) + \delta \mathbf{y} - 2\mathbf{1} = 0 \\ & \boldsymbol{\beta}, \boldsymbol{\beta}', \boldsymbol{\eta}, \boldsymbol{\eta}' \geq \mathbf{0} \wedge \delta \geq 0. \end{aligned} \tag{13}$$

We have reduced the problem of learning count-based kernels to a simple quadratic programming (QP) problem that can be solved by standard solvers.

4.3. Kernel Ridge Regression

Learning count-based rational kernels can also be reduced to a QP problem in the case of KRR.

Using the dual form of kernel ridge regression, the general problem can be written as

$$\begin{aligned} &\min_{\boldsymbol{\mu}} \max_{\boldsymbol{\alpha}} G(\boldsymbol{\mu}, \boldsymbol{\alpha}) = -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \sum_{k=1}^{p} \mu_k (\boldsymbol{\alpha}^\top \mathbf{X}_k)^2 + 2\boldsymbol{\alpha}^\top \mathbf{y} \\ &\text{subject to } \boldsymbol{\mu} \geq 0 \wedge \sum_{k=1}^{p} \mu_k \|\mathbf{X}_k\|^2 = \Lambda. \end{aligned} \tag{14}$$

Proceeding as in the case of the objective function of SVMs, in particular by using the convexity of function G with respect to $\boldsymbol{\mu}$ and its concavity with respect to $\boldsymbol{\alpha}$, and its continuity with respect to both arguments and other arguments similar to the case of SVMs, the optimization problem for learning count-based kernels can be written as

$$\begin{aligned} \min_{\boldsymbol{\alpha}, t} \quad & \lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + \Lambda t^2 - 2\boldsymbol{\alpha}^\top \mathbf{y} \\ \text{subject to} \quad & -t \leq \boldsymbol{\alpha}^\top \mathbf{u}_k \leq t, \forall k \in [1, p], \end{aligned} \tag{15}$$

where $\mathbf{u}_k = \frac{\mathbf{X}_k}{\|\mathbf{X}_k\|}$, $k \in [1, p]$. Let $\mathbf{U} \in \mathbb{R}^{m\times p}$ be the matrix whose kth column is $\mathbf{u}_k$ and introduce the Lagrange variables $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathbb{R}^{p\times 1}$, then again as in the SVM case, differentiating the Lagrangian and substituting for the primal

variables produces the following dual optimization problem

$$L(\boldsymbol{\alpha}, t, \boldsymbol{\beta}, \boldsymbol{\beta}') = \lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + \Lambda t^2 - 2\boldsymbol{\alpha}^\top \mathbf{y} - \boldsymbol{\beta}^\top (\mathbf{U}^\top \boldsymbol{\alpha} + t\mathbf{1}) + \boldsymbol{\beta}'^\top (\mathbf{U}^\top \boldsymbol{\alpha} - t\mathbf{1}). \quad (16)$$

At the optimum the following equalities hold:

$$\begin{cases} \nabla_t L = 2t\Lambda - (\boldsymbol{\beta}' + \boldsymbol{\beta})^\top \mathbf{1} = 0 \\ \nabla_a L = 2\lambda \boldsymbol{\alpha} - 2\mathbf{y} + \mathbf{U}(\boldsymbol{\beta}' - \boldsymbol{\beta}) = \mathbf{0} \end{cases} \Leftrightarrow \begin{cases} t = \frac{1}{2\Lambda}(\boldsymbol{\beta}' + \boldsymbol{\beta})^\top \mathbf{1} \\ \boldsymbol{\alpha} = \frac{1}{2\lambda}(2\mathbf{y} - \mathbf{U}(\boldsymbol{\beta}' - \boldsymbol{\beta})). \end{cases} \quad (17)$$

Plugging the expression for $\boldsymbol{\alpha}$ and t back into (16) yields the equivalent dual optimization problem

$$\max_{\boldsymbol{\beta}, \boldsymbol{\beta}' \geq 0} -\frac{1}{4\lambda} \|2\mathbf{y} - \mathbf{U}(\boldsymbol{\beta}' - \boldsymbol{\beta})\|^2 - \frac{1}{4\Lambda} \|\boldsymbol{\beta}' + \boldsymbol{\beta}\|_1^2. \quad (18)$$

We have thus shown that the problem of learning count-based rational kernels can be reduced to a simple QP problem in the variables $(\boldsymbol{\beta}' + \boldsymbol{\beta})$ and $(\boldsymbol{\beta}' - \boldsymbol{\beta})$.

It is not hard to see that the weights of other rational kernels used in computational biology such as the mismatch kernels Figure 1(b) can be learned using the same QP problems, provided that we impose the constraint that the weight of mapping u to z_k and u' to z_k be the same for a fixed k.

4.4. Kernel Ridge Regression – Alternative Technique

This section describes an alternative technique for solving the problem of learning count-based kernels. We will show that the problem of learning a *kernel matrix* with the KRR objective function admits a solution that in fact coincides with the one prescribed by kernel alignment techniques [28]. An alternative technique for learning the kernel function K is thus to ensure that it matches the optimal kernel matrix $\mathbf{K}$ for the given training sample. When this is possible, the solution obtained coincides with the solutions described in previous sections. Note that this technique can also be applied similarly to the problem of learning rational kernels other than count-based kernels and even to more general types of kernels other than rational kernels.

Using the dual of the KRR optimization, the problem of learning the optimal kernel matrix $\mathbf{K}$ can be formulated as

$$\begin{aligned} \min_{\mathbf{K}} \max_{\boldsymbol{\alpha}} \quad & H(\boldsymbol{\alpha}, \mathbf{K}) = -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y} \\ \text{subject to} \quad & \mathbf{K} \succeq 0 \wedge \operatorname{Tr}[\mathbf{K}] = \Lambda. \end{aligned} \quad (19)$$

Note that for a fixed $\boldsymbol{\alpha}$ the function $\mathbf{K} \mapsto H(\boldsymbol{\alpha}, \mathbf{K})$ is linear and thus convex in $\mathbf{K}$. Thus, $\mathbf{K} \mapsto \max_{\boldsymbol{\alpha}} H(\boldsymbol{\alpha}, \mathbf{K})$ is also convex since the pointwise maximum of a convex function is convex.

To avoid the semidefinite constraint, we can reformulate this problem in terms of a matrix $\mathbf{M}$ such that $\mathbf{M}\mathbf{M}^\top = \mathbf{K}$. By the Cholesky decomposition, such as matrix $\mathbf{M}$ exists. Since $\mathbf{M}\mathbf{M}^T$ is always PDS, the semidefiniteness constraint is thereby made implicit. This leads to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{M}} J(\mathbf{M}) = \max_{\boldsymbol{\alpha}} \quad & -\lambda ||\boldsymbol{\alpha}||^2 - \boldsymbol{\alpha}^\top \mathbf{M}\mathbf{M}^\top \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y} \\ \text{subject to} \quad & \operatorname{Tr}[\mathbf{M}\mathbf{M}^\top] = \Lambda. \end{aligned} \quad (20)$$

J is *not* convex in $\mathbf{M}$, however, since $\mathbf{K} \mapsto \max_{\boldsymbol{\alpha}} H(\boldsymbol{\alpha}, \mathbf{K})$ is convex, any solution $\mathbf{M}$ of this problem must lead to the same value $\mathbf{M}\mathbf{M}^\top = \mathbf{K}$ solution of the problem 19. The optimal value for $\boldsymbol{\alpha}$ in equation (20) has a closed form, which is the standard KRR solution:

$$\boldsymbol{\alpha} = (\mathbf{M}\mathbf{M}^\top + \lambda \mathbf{I})^{-1} \mathbf{y} \,. \quad (21)$$

Using this solution results in the following problem equivalent to (20):

$$\begin{aligned} \min_{\mathbf{M}} \quad & \mathbf{y}^\top (\mathbf{M}\mathbf{M}^\top + \lambda \mathbf{I})^{-1} \mathbf{y} \\ \text{subject to} \quad & \operatorname{Tr}[\mathbf{M}\mathbf{M}^\top] = \Lambda. \end{aligned} \quad (22)$$

The analysis of this optimization problem helps us prove the following theorem.

Theorem 2. *Assume that* $\mathbf{y} \neq \mathbf{0}$*. Then, the unique solution of the optimization problem (19) is* $\mathbf{K} = \frac{\Lambda}{\|\mathbf{y}\|^2} \mathbf{y}\mathbf{y}^\top$.

Proof. Let β denote the dual variable associated to the trace constraint of (22) and $L(\mathbf{M}, \beta)$ its Lagrangian. The gradient of L with respect to $\mathbf{M}$ is given by

$$\nabla_{\mathbf{M}} L(\mathbf{M}, \beta) = 2\Big[-\underbrace{(\mathbf{M}\mathbf{M}^\top + \lambda \mathbf{I})^{-1} \mathbf{y}\mathbf{y}^\top (\mathbf{M}\mathbf{M}^\top + \lambda \mathbf{I})^{-1}}_{\mathbf{N}} + \beta \mathbf{I} \Big] \mathbf{M}. \quad (23)$$

Thus, $\nabla_{\mathbf{M}} L(\mathbf{M}, \beta) = 0$ is equivalent to the vector space spanned by the columns of $\mathbf{M}$ being included in the null-space of $\mathbf{N} + \beta \mathbf{I}$. Let $\mathbf{z}$ be an element of the null-space, then

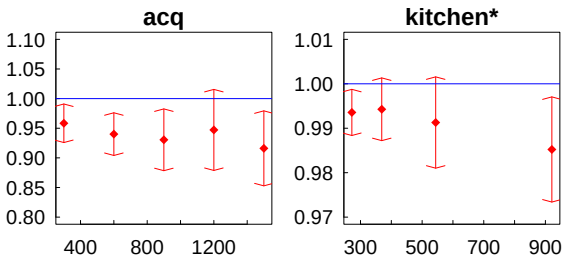
$$\mathbf{z} \in \operatorname{Null}(-\mathbf{N} + \beta \mathbf{I}) \Leftrightarrow \mathbf{N}\mathbf{z} = \beta \mathbf{z} \Leftrightarrow \boldsymbol{\eta}\boldsymbol{\eta}^\top \mathbf{z} = \beta \mathbf{z}, \quad (24)$$

where $\boldsymbol{\eta} = \mathbf{V}^{-1}\mathbf{y}$, with $\mathbf{V} = (\mathbf{M}\mathbf{M}^\top + \lambda \mathbf{I})$. This shows that $\mathbf{z}$ must be an eigenvector of $\boldsymbol{\eta}\boldsymbol{\eta}^\top$ and furthermore $\mathbf{z} \in \operatorname{Span}(\boldsymbol{\eta})$. Using this, we now observe that

$$(-\mathbf{N} + \beta \mathbf{I})\mathbf{M} = 0 \Leftrightarrow \operatorname{Range}(\mathbf{M}) \subseteq \operatorname{Null}(-\mathbf{N} + \beta \mathbf{I}) = \operatorname{Span}(\boldsymbol{\eta}) = \operatorname{Span}(\mathbf{V}^{-1}\mathbf{y}).$$

Dataset	#bigrams	Normalized Error
acq	1500	0.9161 ± 0.0633
crude	1200	0.8448 ± 0.0828
earn	900	0.9196 ± 0.0712
grain	1200	0.9707 ± 0.0294
money-fx	1500	0.9682 ± 0.0396
kitchen*	912	0.9852 ± 0.0118
electronics*	1047	0.9801 ± 0.0104
dvd*	1397	0.9906 ± 0.0125
books*	1349	0.9880 ± 0.0137

(a)



(b)

Fig. 3. Results on classification and regression tasks. (a) An asterisk indicates a regression dataset, otherwise classification. All error rates are normalized by the baseline error rate with standard deviation shown over 10 trials with of the order 1,000 parameters. (b) Results on two dataset as a function of the number of bigrams used for modeling.

Thus, the columns of $\mathbf{M}$ fall in the span of $\mathbf{V}^{-1}\mathbf{y}$, or equivalently there exists a vector $\mathbf{a}$ such that,

$$\begin{aligned}\mathbf{M} = \mathbf{V}^{-1}\mathbf{y}\mathbf{a}^\top &\Leftrightarrow \mathbf{V}\mathbf{M} = \mathbf{y}\mathbf{a}^\top \\ &\Leftrightarrow (\mathbf{M}\mathbf{M}^\top + \lambda\mathbf{I})\mathbf{M} = \mathbf{y}\mathbf{a}^\top \\ &\Leftrightarrow \mathbf{M}(\mathbf{M}^\top\mathbf{M} + \lambda\mathbf{I}) = \mathbf{y}\mathbf{a}^\top \\ &\Leftrightarrow \mathbf{M} = \mathbf{y}\left[(\mathbf{M}^\top\mathbf{M} + \lambda\mathbf{I})^{-1}\mathbf{a}\right]^\top.\end{aligned}$$

Therefore, $\mathbf{M}$ is of the form $\mathbf{y}\mathbf{b}^\top$ and $\mathbf{K} = \mathbf{M}\mathbf{M}^\top = \|\mathbf{b}\|^2\mathbf{y}\mathbf{y}^\top$. Imposing the trace constraint, that is $\mathrm{Tr}(\mathbf{K}) = \|\mathbf{b}\|^2\|\mathbf{y}\|^2 = \Lambda$, yields $\mathbf{K} = \frac{\Lambda}{\|\mathbf{y}\|^2}\mathbf{y}\mathbf{y}^\top$. □

Notice that this solution takes the same form as the one suggested by a maximum alignment type solution [28] and in fact provides a clear justification for the alignment metric.

5. EXPERIMENTS

In this section, we report the results of our experiments to learn count-based rational kernels for both SVM classification problems and KRR tasks.

For the SVM experiments, we considered several one-versus-many classification problems based on the Reuters-21578 dataset[1]. The data was arranged according to the "ModApte" split, as used in [13], which results in a test set of 3,299 documents and training set of 9,603 documents. We randomly chose 1,000 points from the training set to train with over 10 trails.

For the KRR experiments, we used the sentiment analysis dataset found in [29].[2] The data set consists of review text and rating labels, an integer between 1 and 5, taken from `amazon.com` product reviews within four different categories (domains). These four domains consist of `book`, `dvd`, `electronics` and `kitchen` reviews, where each domain contains 2,000 data points. We report values from 10-fold cross validation.

The learning kernel experiments were carried out by first solving either the QP problem in the case of SVM (13) or KRR (18) respectively. The solutions to these QP optimization problems were obtained using the MOSEK software.[3] For the solutions $\boldsymbol{\alpha}$ we found in our experiments about 30% of the k features met the constraints of the optimization (15) ($-t \leq \boldsymbol{\alpha}^\top \mathbf{u}_k \leq t$) as an equality. Thus, at the solution point many features have the same gradient with respect to the parameter t. To avoid favoring one specific feature k and generating a bias, we chose to distribute the trace evenly among the features according to this gradient.

The examination of the features meeting the equality constraint on t reveals that the learning algorithm provides interesting feature selection. Among these features we find many negatively or positively loaded bigrams, such as "recommend this", "lack of", "easy to", "an excellent", and "your money", to name a few examples from the book reviews regression task.

For a baseline, we used equal weights on all the bigrams (i.e. the standard ngram count kernel), with the weights appropriately scaled to meet the same trace constraint as in the case of the learned kernels. In the SVM experiments, we searched for C from 2^{-10} to 2^{10} and $\Lambda = 0.5$. In the KRR experiments, we did a grid search from 2^{-10} to 2^3 in powers of 2 to select the ratio of λ/Λ. The error rates reported are RMSE in the case of regression and zero-one loss in the case of classification. The values are normalized by the baseline error rate, so a value less than one corresponds to an improvement in performance. The results are presented in Figure 3 (a). Figure 3 (b) illustrates the performance as a function of the number of bigrams in the learning task. As can be seen from the figure, for larger number of bigrams, the results become significantly better than the

[1] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`.

[2] `http://www.seas.upenn.edu/~mdredze/datasets/sentiment/`.

[3] `http://www.mosek.com/`.

baseline. These results complement those of [13] give in the transductive setting.

6. CONCLUSION

We presented efficient general algorithms for learning count-based rational kernels, a family of kernels that includes most sequence kernels used in computational biology, natural language processing, and other text processing applications. Our algorithms are thus widely applicable and can help enhance learning performance in a variety of sequence learning tasks. The techniques we used could help learn other families of sequence kernels in a similar way.

7. REFERENCES

[1] Bernhard Schölkopf and Alex Smola, *Learning with Kernels*, MIT Press: Cambridge, MA, 2002.

[2] John Shawe-Taylor and Nello Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge Univ. Press, 2004.

[3] Bernhard E. Boser, Isabelle Guyon, and Vladimir N. Vapnik, "A training algorithm for optimal margin classifiers," in *COLT*, 1992, vol. 5, pp. 144–152.

[4] Corinna Cortes and Vladimir N. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[5] Vladimir N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[6] David Haussler, "Convolution Kernels on Discrete Structures," Tech. Rep. UCSC-CRL-99-10, University of California at Santa Cruz, 1999.

[7] Christina S. Leslie, Eleazar Eskin, Adiel Cohen, Jason Weston, and William Stafford Noble, "Mismatch string kernels for discriminative protein classification," *Bioinformatics*, vol. 20, no. 4, 2004.

[8] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller, "Engineering support vector machine kernels that recognize translation initiation sites.," *Bioinformatics*, vol. 16, no. 9, pp. 799–807, 2000.

[9] Asa Ben-Hur and William Stafford Noble, "Kernel methods for predicting protein-protein interactions," in *ISMB, Supplement of Bioinformatics*, 2005, pp. 38–46.

[10] Michael Collins and Nigel Duffy, "Convolution kernels for natural language," in *NIPS 14*. 2002, MIT Press.

[11] Corinna Cortes, Patrick Haffner, and Mehryar Mohri, "Rational Kernels: Theory and Algorithms," *Journal of Machine Learning Research*, vol. 5, pp. 1035–1062, 2004.

[12] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins, "Text classification using string kernels," *Journal of Machine Learning Research*, vol. 2, pp. 419–44, 2002.

[13] Gert R. G. Lanckriet, Nello Cristianini, Peter L. Bartlett, Laurent El Ghaoui, and Michael I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[14] Seung-Jean Kim, Argyrios Zymnis, Alessandro Magnani, Kwangmoo Koh, and Stephen Boyd, "Learning the kernel via convex optimization," in *Proceedings of ICASSP '08*, 2008.

[15] Cheng Soon Ong, Alexander J. Smola, and Robert C. Williamson, "Learning the kernel with hyperkernels," *Journal of Machine Learning Research*, vol. 6, pp. 1043–1071, 2005.

[16] Charles A. Micchelli and Massimiliano Pontil, "Learning the kernel function via regularization," *Journal of Machine Learning Research*, vol. 6, pp. 1099–1125, 2005.

[17] Andreas Argyriou, Charles A. Micchelli, and Massimiliano Pontil, "Learning convex combinations of continuously parameterized basic kernels," in *COLT*, 2005, pp. 338–352.

[18] Andreas Argyriou, Raphael Hauser, Charles A. Micchelli, and Massimiliano Pontil, "A DC-programming algorithm for kernel selection," in *ICML*, 2006, pp. 41–48.

[19] Tony Jebara, "Multi-task feature and kernel selection for SVMs," in *ICML*, 2004.

[20] Darrin P. Lewis, Tony Jebara, and William Stafford Noble, "Nonstationary kernel combination," in *ICML*, 2006.

[21] Alexander Zien and Cheng Soon Ong, "Multiclass multiple kernel learning," in *ICML*, 2007, pp. 1191–1198.

[22] Craig Saunders, Alexander Gammerman, and Volodya Vovk, "Ridge Regression Learning Algorithm in Dual Variables," in *ICML*, 1998, pp. 515–521.

[23] Arto Salomaa and Matti Soittola, *Automata-Theoretic Aspects of Formal Power Series*, Springer-Verlag, 1978.

[24] Christina S. Leslie and Rui Kuang, "Fast String Kernels using Inexact Matching for Protein Sequences," *Journal of Machine Learning Research*, vol. 5, pp. 1435–1455, 2004.

[25] Leonard Pitt and Manfred Warmuth, "The minimum consistent DFA problem cannot be approximated within any polynomial," *Journal of the Assocation for Computing Machinery*, vol. 40, no. 1, pp. 95–142, 1993.

[26] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[27] J. von Neumann, "Uber ein ökonomisches Gleichungssystem und eine Verallgemeinerung des Brouwerschen Fixpunktsatzes," in *Ergebn. Math. Kolloq. Wein 8*, 1937, pp. 73–83.

[28] Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz S. Kandola, "On kernel-target alignment," in *NIPS*, 2001, pp. 367–373.

[29] John Blitzer, Mark Dredze, and Fernando Pereira, "Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification," in *Association for Computational Linguistics*, 2007.