LEARNING THE KERNEL VIA CONVEX OPTIMIZATION

Seung-Jean Kim, Argyrios Zymnis, Alessandro Magnani, Kwangmoo Koh, and Stephen Boyd

Information Systems Laboratory, Department of Electrical Engineering
Stanford University, Stanford, CA 94305-9510
{sjkim,azymnis,alem,deneb1,boyd}@stanford.edu

ABSTRACT

The performance of a kernel-based learning algorithm depends very much on the choice of the kernel. Recently, much attention has been paid to the problem of learning the kernel itself from given training examples. The main emphasis has been on formulating the problem as a tractable convex optimization problem. Only for a few very special cases such as support vector machines are explicit convex formulations known. In this paper, we show that, in a wide variety of kernel-based learning algorithms, the kernel learning problem can be formulated as a convex optimization problem which interior-point methods can solve globally and efficiently. The kernel learning method is illustrated with a regression problem that arises in petroleum engineering.

Index Terms— Convex optimization, kernel methods, machine learning, support vector machine

1. INTRODUCTION

1.1. Kernel-based learning algorithms

Let $\mathcal{X}$ denote an arbitrary input or instance set and $\mathcal{Y}$ denote an output set. An input-output pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is called an example. We call a symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a *kernel (function)* if it satisfies the finitely positive semidefinite property: for any $x_1, \ldots, x_m \in \mathcal{X}$, the *Gram matrix* $G \in \mathbb{R}^{m\times m}$, defined by $G_{ij} = K(x_i, x_j)$, is positive semidefinite [1]. Mercer's theorem tells us that the kernel K implicitly maps the input set $\mathcal{X}$ to a high-dimensional (possibly infinite) reproducing kernel Hilbert space $\mathcal{H}$ equipped with the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ through a mapping $\phi : \mathcal{X} \to \mathcal{H}$: $K(x, x') = \langle\phi(x), \phi(x')\rangle_{\mathcal{H}}$, $\forall\, x, x' \in \mathcal{X}$. The mapping is called the *feature mapping*, and the space $\mathcal{H}$ is called the *feature space*. The mapping and space will be denoted as ϕ_K and $\mathcal{H}_K$, when it is necessary to indicate the dependence on the kernel K. We will often write the inner product $\langle\phi_K(x), \phi_K(x')\rangle_{\mathcal{H}}$ as $\phi_K(x)^T\phi_K(x')$.

Let $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathcal{Y}$ be given training examples. In kernel-based regression, from the training example, we learn a function $h : \mathcal{X} \to \mathcal{Y}$ of the form $h(x) = \langle w, \phi(x)\rangle + v$, where $w \in \mathcal{H}$ is the weight vector and $v \in \mathbb{R}$ is the bias or intercept. This function interpolates the given training inputs to predict the value at a new point x. In kernel-based (binary) classification with $\mathcal{Y} = \{-1, +1\}$, we learn a classifier $h : \mathcal{X} \to \mathcal{Y}$ of the form $h(x) = \operatorname{sgn}(\langle w, \phi(x)\rangle + v)$, where $\operatorname{sgn}(\cdot)$ is the signum function. The function h predicts the binary label of $x \in \mathcal{X}$.

A wide variety of kernel-based machine learning algorithms can be formulated as optimization problems of the form

$$\begin{array}{ll}\text{minimize} & f_0(w^T\phi_K(x_1), \ldots, w^T\phi_K(x_m), \zeta, \langle w, w\rangle_{\mathcal{H}_K}) \\ \text{subject to} & f_i(w^T\phi_K(x_1), \ldots, w^T\phi_K(x_m), \zeta, \langle w, w\rangle_{\mathcal{H}_K}) \\ & \quad \le 0,\ i = 1, \ldots, M,\end{array} \tag{1}$$

where $w \in \mathcal{H}_K$ and $\zeta \in \mathbb{R}^p$ are the variables and f_i are functions from $\mathbb{R}^{m+p+1}$ into $\mathbb{R}$. Here, the kernel function is given and ζ represents auxiliary variables (including the intercept). In this problem, w appears in a very specific form in the objective and constraint functions: it appears in the inner product $w^T\phi_K(x_i)$ with the value of the feature mapping at x_i and in the quadratic form $\langle w, w\rangle_{\mathcal{H}_K}$.

As an example, we consider the hard margin support vector machine (SVM), which can be formulated as

$$\begin{array}{ll}\text{minimize} & \langle w, w\rangle_{\mathcal{H}_K} \\ \text{subject to} & y_i(w^T\phi_K(x_i) + v) \ge 1, \quad i = 1, \ldots, m,\end{array} \tag{2}$$

where the variables are $w \in \mathcal{H}_K$ and $v \in \mathbb{R}$. This problem has the form in (1).

As another example, we consider the problem of finding the weight vector and intercept that minimize a (regularized) loss functional

$$\frac{1}{m}\sum_{i=1}^{m}\psi(y_i, w^T\phi_K(x_i) + v) + \lambda\langle w, w\rangle_{\mathcal{H}_K}, \tag{3}$$

where $w \in \mathcal{H}_K$ and $v \in \mathbb{R}$ are the variables, $\lambda > 0$ is a regularization parameter, $\psi : \mathbb{R}^2 \to \mathbb{R}$ is a loss function (*e.g.*, the logistic, square, or hinge loss). This problem is called a convex loss minimization problem when the loss function is convex. Evidently, this problem has the form in (1) (with no constraints) in which ζ represents the intercept. Many learning problems including 1-norm soft margin SVMs, 2-norm soft margin SVMs, and kernel logistic regression are convex loss minimization problems.

1.2. Learning the kernel

Let $\mathcal{K}$ be a convex set of positive semidefinite kernel functions. Here, the convexity of $\mathcal{K}$ means that for any $K_1, K_2 \in \mathcal{K}$, $\theta K_1 + (1-\theta)K_2 \in \mathcal{K}$, $\forall \theta \in (0,1)$. A standard example is the set of all affine combinations of given positive semidefinite kernel functions $K_1, \ldots, K_p$:

$$\mathcal{K} = \left\{ K : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \;\middle|\; K = \sum_{i=1}^{p} \theta_i K_i,\ \mathbf{1}^T\theta = 1,\ \theta \geq 0 \right\}.$$

where $\mathbf{1}$ is the vector of all ones and $\theta \geq 0$ means $\theta_i \geq 0$, $i = 1, \ldots, p$. Often the kernels K_i are chosen to satisfy the normalization constraint: the Gram matrices computed with $x_1, \ldots, x_m$ have the same trace. (See [2] for a discussion on the constraint in the context of generalization performance analysis.)

The performance of a kernel-based learning algorithm depends very much on the choice of the kernel. Typically, a parameterized family of kernels, *e.g.*, the Gaussian or polynomial kernel family, is chosen and the kernel parameters are tuned via cross-validation or generalized cross-validation. Recently, much attention has been paid to the problem of learning the kernel itself along with the classifier from given training examples. For a kernel-based learning algorithm which can be formulated as (1), the corresponding *kernel learning problem* can be written as

$$\begin{array}{ll} \text{minimize} & F(K) \\ \text{subject to} & K \in \mathcal{K}, \end{array} \tag{4}$$

where $F(K)$ is the optimal value of (1). In this problem, we learn the kernel function and the parameters in the kernel-based algorithm simultaneously from given training examples. The kernel set $\mathcal{K}$ should be chosen properly to provide good generalization performance. The reader is referred to [2] for a discussion on the merits of learning the kernel.

1.3. Prior work

There has been a growing interest in learning the kernel from given training examples; [3, 4, 2, 5]. The main emphasis has been on formulating kernel learning as a tractable convex problem. However, only for a few very special cases such as support vector machines and regression are explicit convex formulations known. For instance, the problem of learning the kernel in SVMs can be cast as a semidefinite program [2].

In [6], the authors consider the kernel learning problem that arises in the convex loss minimization problem (3), which can be formulated as

$$\begin{array}{ll} \text{minimize} & \inf\limits_{w \in \mathcal{H}_K} \dfrac{1}{m} \sum\limits_{i=1}^{m} \psi\left(y_i, w^T\phi_K(x_i) + v\right) + \lambda \langle w, w \rangle_{\mathcal{H}_K} \\ \text{subject to} & K \in \mathcal{K}, \end{array} \tag{5}$$

where the variable is the kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The authors show through functional analysis and convex duality that the loss function in (3) is a *convex functional* of the variable, *i.e.*, the kernel function, when the loss function ψ is convex. Despite the convexity of the loss functional, the kernel learning problem (5) is not tractable in this form, since it is not computationally feasible to evaluate even the first derivative of the objective. Several authors have proposed solution methods for the kernel learning problem (5) such as a semi-infinite linear programming approach and an iterative method that alternates between finding the weight vector and intercept in (3) with a fixed kernel and updating the kernel [7].

1.4. Summary

The main purpose of this paper is to show that under suitable assumptions on the functions in the objective and constraints of (1), the corresponding kernel learning problem (4) can be formulated as a convex optimization problem which interior-point methods can solve globally and efficiently. We illustrate the kernel learning method on a regression problem that arises in petroleum engineering. This example shows that the kernel learning method has the potential of replacing cross-validation in tuning kernel parameters, which is in line with the observation made in the literature.

2. CONVEX FORMULATION

2.1. An extension of the representer theorem

We can easily extend the representer theorem [1] to (1).

Proposition 1. *Suppose that the functions f_i that appear in the objective and constraints in (1) are nondecreasing in the last argument. If $\alpha^\star$ and $\zeta^\star$ solve*

$$\begin{array}{ll} \text{minimize} & f_0(w^T\phi_K(x_1), \ldots, w^T\phi_K(x_m), \zeta, \langle w, w \rangle_{\mathcal{H}_K}) \\ \text{subject to} & f_i(w^T\phi_K(x_1), \ldots, w^T\phi_K(x_m), \zeta, \langle w, w \rangle_{\mathcal{H}_K}) \\ & \leq 0, \quad i = 1, \ldots, m, \\ & w = \sum_{i=1}^{m} \alpha_i \phi_K(x_i), \end{array} \tag{6}$$

then $w^\star = \sum_{i=1}^{m} \alpha_i^\star \phi_K(x_i)$ and $\zeta^\star$ solve (1).

Suppose that w is in the span of the set

$$\{\phi_K(x_i) \mid i = 1, \ldots, m\}.$$

For any $w = \sum_{i=1}^{m} \alpha_i \phi_K(x_i)$, we then have

$$w^T\phi_K(x_i) = e_i^T G_K \alpha, \quad \langle w, w \rangle_{\mathcal{H}_K} = \alpha^T G_K \alpha$$

where $G_K \in \mathbb{R}^{m \times m}$ is the Gram matrix computed with the kernel K at $x_1, \ldots, x_m$. It is now clear from Proposition 1 that problem (6) is equivalent to

$$\begin{array}{ll} \text{minimize} & f_0(G_K\alpha, \zeta, \alpha^T G_K \alpha) \\ \text{subject to} & f_i(G_K\alpha, \zeta, \alpha^T G_K \alpha) \leq 0, \quad i = 1, \ldots, M, \end{array} \tag{7}$$

with variables are $\alpha \in \mathbb{R}^m$. (The two problems are equivalent in the sense that a solution of each problem is readily determined from a solution of the other.)

Once the optimal $\alpha^\star$ is found, for a given point $x \in \mathcal{X}$, we can compute the inner product $\langle w^\star, \phi_K(x)\rangle_{\mathcal{H}_K}$ as

$$\langle w^\star, \phi_K(x)\rangle_{\mathcal{H}_K} = \sum_{i=1}^m \alpha_i^\star \phi_K^T(x_i)\phi_K(x) = \sum_{i=1}^m \alpha_i^\star K(x_i, x).$$

To compute the inner product, we evaluate the kernel function at the pairs (x_i, x), $i = 1, \ldots, m$, not the feature mapping, which is known as the kernel trick [1].

2.2. Convex formulation

Let $\mathcal{G}$ be the set of Gram matrices consistent with the assumption made on the kernel function:

$$\mathcal{G} = \{G_K \mid K \in \mathcal{K}\} \subseteq \mathbb{S}_+^m.$$

Here we use $\mathbb{S}_+^m$ to denote the set of all $m \times m$ positive semidefinite matrices. This set is convex since $\mathcal{K}$ is. The kernel learning problem (4) is equivalent to

$$\begin{array}{ll} \text{minimize} & f_0(G\alpha, \zeta, \alpha^T G\alpha) \\ \text{subject to} & f_i(G\alpha, \zeta, \alpha^T G\alpha) \leq 0, \quad i = 1, \ldots, M, \\ & G \in \mathcal{G}, \end{array} \tag{8}$$

in which the variables are $\alpha \in \mathbb{R}^m$, $\zeta \in \mathbb{R}^p$, and $G = G^T \in \mathbb{R}^{m\times m}$, which are all finite-dimensional.

We will reformulate (8) as a problem in which the variables are $z = G\alpha$ and G instead of α and G. Let $G^\dagger$ be the pseudoinverse of G, which is not necessarily invertible. Then, $GG^\dagger G = G$, which is a basic property of the pseudoinverse. We can now see that (7) can be equivalently written as

$$\begin{array}{ll} \text{minimize} & f_0(z, \zeta, z^T G^\dagger z) \\ \text{subject to} & f_i(z, \zeta, z^T G^\dagger z) \leq 0, \quad i = 1, \ldots, M, \\ & G \in \mathcal{G}, \end{array} \tag{9}$$

where the variables are $z \in \mathbb{R}^m$, $\zeta \in \mathbb{R}^p$, and $G = G^T \in \mathbb{R}^{m\times m}$. The two problems are equivalent in the following sense: if $(z^\star, \zeta^\star, G^\star)$ solves (9), then $(\alpha^\star, \zeta^\star, G^\star)$ with $\alpha^\star = G^{\star\dagger} z^\star$ solves (8), and conversely if $(\alpha^\star, \zeta^\star, G^\star)$ solves (8), then if $(\alpha^\star, \zeta^\star, G^\star)$ solves this problem, then $(z^\star, \zeta^\star, G^\star)$ with $z^\star = G^\star \alpha^\star$ solves (9).

The function $g(z, G) = z^T G^\dagger z$ is convex on $\mathbb{R}^m \times \mathbb{S}_+^m$ [8, §3]. The convexity can be seen from the fact that for any $t > 0$,

$$x^T G^\dagger x \leq t \quad \text{if and only if} \quad \left[\begin{array}{cc} t & x \\ x^T & G \end{array}\right] \succeq 0, \tag{10}$$

which is known as the Schur complement technique. (The right-hand side is a convex constraint.)

We can easily check the convexity of the equivalent formulation (9). Suppose that the functions f_i that appear in the objective and constraints in (1) are convex and nondecreasing in each argument. Then, the functions f_i are convex and hence (9) is a convex optimization problem. The convexity of f_i follows from the composition rule: $h \circ g$ is convex if $h : \mathbb{R}^p \to \mathbb{R}$ is convex and nondecreasing in each element, and $g : \mathbb{R}^n \to \mathbb{R}^p$ is convex. (See [8, §3] for more on operations that preserve convexity.)

Using the change of variables described above, we can show that the kernel learning problem (5) in convex loss minimization is equivalent to the convex problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \psi(y_i z_i + y_i v) + \lambda z^T G^\dagger z \\ \text{subject to} & G \in \mathcal{G}, \end{array} \tag{11}$$

where the variables are $z \in \mathbb{R}^m$, $v \in \mathbb{R}$, and $G = G^T \in \mathbb{R}^{m\times m}$. Similarly, we can see that the kernel learning problem in the hard margin SVM (2) can be reformulated as a convex problem using the change of variables.

We should point out that the meaning of the convexity of (11) is different from that of the convexity of the original kernel learning problem (5); we can solve (9) efficiently using standard methods for convex optimization such as interior-point methods, since there is an efficient way of evaluating the first and second derivatives the objective and the constraint functions.

When the loss function ψ in (11) is the hinge loss function, *i.e.*, $\psi_{\text{hin}}(y, u) = \max\{0, 1 - yu\}$, in [2], the authors show that the kernel learning problem (5) can be cast as a semidefinite program, using an argument based on convex duality and the minimax theorem for convex/concave functions. The convex problem (11) can be cast as the same semidefinite program, using the Schur complement technique described above.

3. NUMERICAL EXAMPLE

In this section we describe an application that arises in petroleum engineering. One of the main challenges in this field is to devise an algorithm that optimally chooses the start and end drilling locations of a well, called a well-placement scenario, so as to maximize the oil output over a given time period. For this purpose a number of oil reservoir simulators have been developed that can accurately estimate the performance of a given well-placement scenario [9]. Such a simulator can be used in conjunction with a local optimization algorithm so as to choose a good well location for a given reservoir.

The problem with such an approach is that reservoir simulations are computationally expensive. As the optimization algorithm explores the space of possible scenarios, it wastes a significant amount of time by simulating scenarios that are obviously not of interest. Therefore a better approach is to compute an estimate of the performance of a new simulation based on the results of previous simulations and then decide wether or not it is worth simulating this scenario based on the performance estimate. This approach is investigated in [10], which describes an estimator which is a slight variation of k-nearest neighbors.

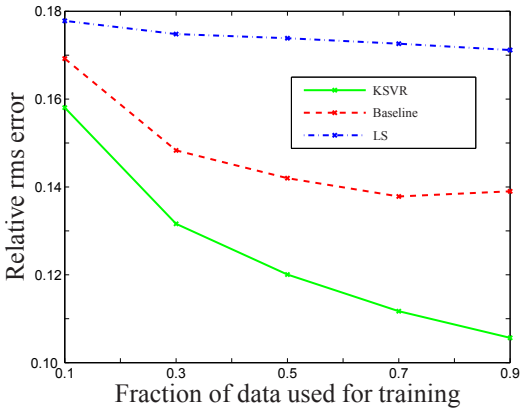

Fig. 1. Relative rms error vs. training set size

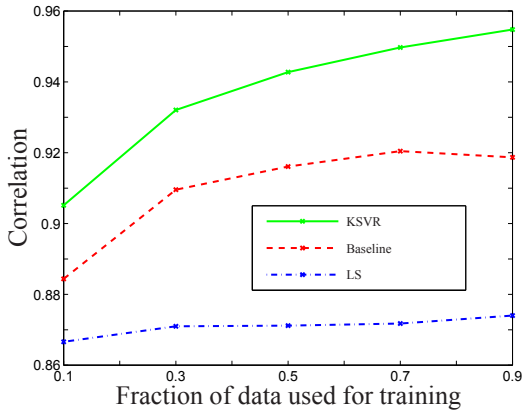

Fig. 2. Mean correlation vs. training set size

We compared the performance of optimal kernel-based regression with the quadratic loss function $\psi_{\mathrm{quad}}(y,u) = (y-u)^2$ against the baseline method used in [10]. The data set consists of the results obtained from 2000 scenario simulations and there are 11 features including geometric parameters; see [10] for the details. We used a convex combination of 10 Gaussian kernels $K(x,z) = \sum_{i=1}^{10} \theta_i e^{-\|x-z\|^2/\sigma_i^2}$, where θ_i are the nonnegative weights of the kernels to be determined and satisfy $\mathbf{1}^T\theta = 1$. The bandwidths σ_i were chosen uniformly spaced in the interval $[10^0, 10^3]$ on a logarithmic scale. The regularization parameter was fixed at $\lambda = 0.2$ in all cases. The two performance metrics that we used are the relative root mean square (rms) error $e_{\mathrm{rms}}(\hat{y}, y_{\mathrm{test}}) = \|\hat{y} - y_{\mathrm{test}}\|_2/\|y_{\mathrm{test}}\|_2$ and the correlation $\rho(\hat{y}, y_{\mathrm{test}})$.

In order to compare these methods we varied the training set size from 10% to 90% of the available data and in each case trained each method for 15 random training and test set partitions. Figure 1 shows the average relative rms error and figure 2 shows the average correlation as a function of training set size for both methods, as well as for nominal least-squares regression. Kernel-based regression (with a kernel learned from the training examples) clearly outperforms the baseline and least squares (LS) methods in both performance metrics. Because our kernel learning method can support any convex loss function in regression, we also applied other convex loss functions, such as the ϵ-sensitive quadratic loss and ℓ_1 loss, to the data set, but we found no noticeable performance improvement.

4. CONCLUSIONS

We have shown how to formulate a kernel learning problem in a general kernel-based classification problem as a convex optimization. Since convex problems are computationally tractable, optimal kernel selection is tractable in a wide variety of kernel-based learning algorithms. The convex formulation given in (9) also serves as a generic prototype for developing an efficient solution method for learning the kernel.

5. REFERENCES

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

[2] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan, "Learning the kernel matrix with semi-definite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[3] A. Argyriou, M. Hauser, C. Micchelli, and M. Pontil, "A DC algorithm for kernel selection," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[4] S.-J. Kim, A. Magnani, and S. Boyd, "Optimal kernel selection in kernel fisher discriminant analysis," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*. 2004, ACM.

[5] C. Ong, A. Smola, and R. Williamson, "Learning the kernel with hyperkernels," *Journal of Machine Learning Research*, vol. 6, pp. 1043–1071, 2005.

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

[7] A. Argyriou, M. Herbster, and M. Pontil, "Combining graph Laplacians for semi-supervised learning," in *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[9] H. Cao, *Development of Techniques for General Purpose Simulators*, Ph.D. thesis, Stanford University, 2002.

[10] V. Artus, L.J. Durlofsky, J. Onwunalu, and K. Aziz, "Optimzation of nonconventional wells under uncertainty using statistical proxies," *Computational Geosciences*, vol. 10, no. 4, pp. 389–404, 2006.