
The Feature Selection Path in Kernel Methods

Fuxin Li

Institute of Numerical Simulation, University of Bonn
{fuxin.li, cristian.sminchisescu}@ins.uni-bonn.de

Cristian Sminchisescu

Abstract

The problem of automatic feature selection/weighting in kernel methods is examined. We work on a formulation that optimizes both the weights of features and the parameters of the kernel model simultaneously, using L_1 regularization for feature selection. Under quite general choices of kernels, we prove that there exists a unique regularization path for this problem, that runs from 0 to a stationary point of the non-regularized problem. We propose an ODE-based homotopy method to follow this trajectory. By following the path, our algorithm is able to automatically discard irrelevant features and to automatically go back and forth to avoid local optima. Experiments on synthetic and real datasets show that the method achieves low prediction error and is efficient in separating relevant from irrelevant features.

1 Introduction

Kernel methods are powerful universal function approximators that are guaranteed to find the best solution for a regularized learning problem, among all the functions in a possibly infinite-dimensional Hilbert space (Hofmann et al., 2008). The success of kernel methods depends heavily on a proper choice of the kernel, a positive semi-definite function that is used as a similarity measure in the input space. In practice, the quality of the similarity function can be radically affected by irrelevant features.

Several approaches have been proposed to select relevant features for kernel methods. Guyon et al. (2002) proposed SVM-RFE to eliminate features from the full

set sequentially. Bi et al. (2003) performed feature selection on a linear SVM and used the results to build a kernel for a nonlinear SVM. Lin and Zhang (2006) proposed COSSO, to select features in smoothing spline regression by breaking up the regularization term into components on individual dimensions. A related topic is multiple kernel learning, e.g. (Lanckriet et al., 2004; Bach, 2008), which solves for the optimal linear combination of a set of basic kernels.

However, feature selection is inherently a combinatorial problem, and in order to find the optimal solution, all possible feature combinations must be explored. This is usually intractable, hence most feature selection methods use heuristics which make it unclear how good the solution is. Recently, Bach (2009) has proposed a powerful algorithm (HKL) to explore all feature combinations in polynomial time. This works by choosing kernels corresponding to feature combinations mapped to a directed acyclic graph.

Another approach to feature selection is to parametrize kernels with a weight on each feature. An example is $k(x, z) = \exp(-\sum_{i=1}^d \beta_i \|x_i - z_i\|^2)$, the ARD-Gaussian kernel first used in Gaussian processes (Rasmussen & Williams, 2006) and with SVMs (Grandvalet & Canu, 2003; Chapelle et al., 2002; Mangasarian & Wild, 2007). Keerthi et al. (2007) give an efficient algorithm that alternates between learning an SVM and optimizing β , the feature weights. Varma and Babu (2009) propose a projected gradient method with L_1 regularization and report encouraging results.

However, the feature weighting problem is non-convex due to the β term inside the kernel. It is hard to even find a convex relaxation. Gradient-based methods find a single local optimum, and the starting point needs to be chosen heuristically for good results. But it remains unclear how good the solution is w.r.t. the global optimum. In this work, we attack this problem from the different perspective of regularization paths.

In machine learning, a regularization path is the continuous trace of optima created by solving a learning problem with different levels of regularization. In most convex regularized machine learning problems,

Appearing in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

all global optima reside on this path. The study of regularization paths began with the homotopy method of Osborne et al. (2000) and Efron et al. (2004). For L_1 regularized linear regression (LASSO), they have found the natural least angle direction that solutions must follow. This direction is used to compute the entire regularization path with the same cost of computing a single least squares fit. In later work, the computation of regularization paths has been explored for non-linear problems (Rosset, 2005), multiple kernel learning (Bach et al., 2004), solution paths generated by varying a single kernel parameter (Wang et al., 2007) and boosting (Zhao & Yu, 2007).

All algorithms work on convex problems. Would there be a regularization path in a non-convex problem, such as ours? One may suspect that because of local optima, a continuous regularization path may not exist: it might be broken midway, cycle, or bifurcate. There might also be multiple paths or isolated optima.

Our work reveals some of the structure in this non-convex problem. Building upon differential topology, we show that for a wide class of kernel functions, a *unique* regularization path exists from the fully regularized case (i.e. the learning machine that outputs 0 for every input) to the case of no regularization. In experiments, we visualize the path and show that not only does it find at least one solution for each level of regularization, but also automatically goes back and forth (in terms of the regularization parameter) sometimes, finding multiple stationary points for the same optimization problem. This gives a better chance to find the global optimum and provides insights into the behavior of stationary points.

The non-convex regularization paths are highly fragile and one must avoid jumping out of the path to points that do not lead to a local optimum. Therefore, we propose an ODE-based algorithm in order to closely follow the piecewise-smooth solution path. We show that the prediction accuracy of the algorithm compares favorably to HKL (Bach, 2009), a state-of-the-art algorithm which performs optimal nonlinear feature/kernel selection over an exponential number of kernels.

The paper is organized as follows: In sec. 2 we describe the optimization problem and prove the existence and uniqueness of the regularization path. Sec. 3 describes our homotopy-based ODE algorithm to trace the solution path. Experiments with synthetic and real data are shown in Sec. 4 and we conclude in Sec. 5.

2 The Regularization Path

Let n be the number of training examples, d the number of input dimensions, I the identity matrix, and

$\mathbf{1}$ the vector of all ones. Denote $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ the centering matrix. Let $\{x_1, \dots, x_n\}$ be the training set and y the vector of training responses. We assume the data is normalized with $\sum_{i=1}^d x_{ij} = 0$, $\sum_{i=1}^d x_{ij}^2 = 1$, and $\sum_{i=1}^n y_i = 0$.

2.1 Feature Weighting in Kernel Ridge Regression

For simplicity, we work with kernel ridge regression (KRR). However the proofs in this section hold for any second-order smooth loss function. We work with the following formulation:

$$\begin{aligned} \min_{\alpha, \beta, \alpha_0} \quad & \|y - K(\beta)\alpha - \alpha_0\mathbf{1}\|^2 \\ \text{s.t.} \quad & \alpha^T K(\beta)\alpha \leq C, \|\beta\|_1 \leq D, \end{aligned} \quad (1)$$

where $K(\beta)$ represents the kernel matrix of the training set, parametrized by β . With the KRR loss function, α is uniquely determined if β, C, D are fixed, making the analysis simpler.

The pioneering work of Efron et al. (2004) suggested that for L_1 regularization, additional structures on the minimizer can be used to construct efficient homotopy algorithms that trace the solution path. Rosset (2005) stated the following results from the KKT conditions of an L_1 regularized problem, given here as a theorem:

Theorem 1 (Rosset, 2005) *For any loss function $L(y, f(X, \beta))$ that is differentiable on β , any stationary point β^* of the optimization problem:*

$$\min_{\beta} L(y, f(X, \beta)), \text{ s.t. } \|\beta\|_1 \leq D$$

has the property that: (a) $\beta_i \neq 0 \Rightarrow \left| \frac{\partial L}{\partial \beta_i} \right| = \max_j \left| \frac{\partial L}{\partial \beta_j} \right|$; (b) $\text{sign}(\beta_i) = -\text{sign}\left(\frac{\partial L}{\partial \beta_i}\right)$.

We analyze the derivatives of the objective function in order to examine the points that satisfy Theorem 1. First, the problem (1) is transformed to a less constrained form using Lagrange multipliers. Note that $\lambda > 0$ is a variable here.

$$\begin{aligned} L(\alpha, \beta, \alpha_0) &= \|y - K(\beta)\alpha - \alpha_0\mathbf{1}\|^2 + \lambda(\alpha^T K(\beta)\alpha - C) \\ \text{s.t.} \quad & \|\beta\|_1 \leq D \end{aligned} \quad (2)$$

Setting the derivatives to 0 in (2) and assuming $K(\beta)$ is full-rank, we obtain:

$$\begin{aligned} \lambda\alpha &= y - K(\beta)\alpha - \alpha_0\mathbf{1} \\ (HK(\beta) + \lambda I)\alpha &= y; \frac{\partial L}{\partial \beta_k} = -\lambda\alpha^T \frac{\partial K}{\partial \beta_k} \alpha \end{aligned} \quad (3)$$

Denote $\eta = \max_i \left| \frac{\partial L}{\partial \beta_i} \right|$, and suppose the current active set is $\{k_1, k_2\}$. Theorem 1 implies that $\eta = \left| \frac{\partial L}{\partial \beta_{k_1}} \right| = \left| \frac{\partial L}{\partial \beta_{k_2}} \right| \geq \left| \frac{\partial L}{\partial \beta_i} \right|$. We then have:

$$\left| \alpha^T \frac{\partial K}{\partial \beta_{k_1}} \alpha \right| - \left| \alpha^T \frac{\partial K}{\partial \beta_{k_2}} \alpha \right| = 0 \quad (4)$$

Collecting (3) and (4), note that we must have $\alpha^T K(\beta)\alpha = C$ when $\lambda > 0$, we have one more variable than equations. From the implicit function theorem, this gives a smooth trajectory of solutions (α, β, λ) when the sign of β_{k_1} and β_{k_2} does not change. In the spirit of homotopy algorithms, we start at $D = 0$ and follow the trajectory (Fig. 1). It can be seen that the trajectory is smooth until one of the two possible events occurs: **E1**: Another variable k_3 joins the active set as $\left|\frac{\partial L}{\partial \beta_{k_3}}\right| = \eta$; **E2**: Any of the β_{k_i} becomes 0. We call the points where an event occur *event points*.

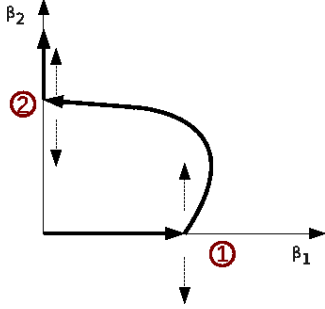


Figure 1: Illustration of events. 1) β_2 enters the path. Its sign is determined by Theorem 1 (b); 2) β_1 leaves the path. The direction of the path is determined by probing both directions to check which one satisfies Theorem 1 (a).

If a new variable joins the active set, another equation needs to be added to the system. If a particular β_{k_i} becomes 0, the condition $\eta = \left|\frac{\partial L}{\partial \beta_{k_i}}\right|$ is no longer needed, thus the equation corresponding to that β_{k_i} can be dropped from the system.

To conclude, the trajectory can be followed from $D = 0$. At special points where any of the events **E1** and **E2** occur, we stop and update the system, then continue to follow the trajectory until we reach a solution of a problem without L_1 regularization ($D \rightarrow \infty$) where $\frac{\partial L}{\partial \beta_{k_i}} = 0$. At this point, a piecewise smooth solution path on α and β , corresponding to the original problem (1), has been obtained.

The KKT conditions with active set $\{k_1, \dots, k_p\}$ are:

$$\begin{aligned} (HK(\beta) + \lambda I)\alpha &= y \\ \alpha^T \left(\frac{\partial K}{\partial \beta_{k_i}} - \frac{\partial K}{\partial \beta_{k_{i+1}}} \right) \alpha &= 0, i = 1, \dots, p-1 \\ \alpha^T K(\beta)\alpha &= C \end{aligned} \quad (5)$$

2.2 Proof for existence and uniqueness

Although we have described a procedure to follow the solution path, an important prerequisite is to prove that the solution path exists and extends as $D \rightarrow \infty$.

The proof is inspired by ideas in probability-one homotopy methods (Chow et al., 1978). The gist is shown in Figure 2, (1) - (3). We make use of the 1-D manifold classification theorem in differential topology (Milnor, 1978), which states that a 1-dimensional smooth manifold must be homeomorphic either to a line segment or to a circle. Therefore we need to show two things: first, the path is (close to) a 1-D manifold (it does not self-intersect); second, this manifold cannot be a circle. The first is a local property, and the implicit function theorem is used to prove it. To prove that the mani-

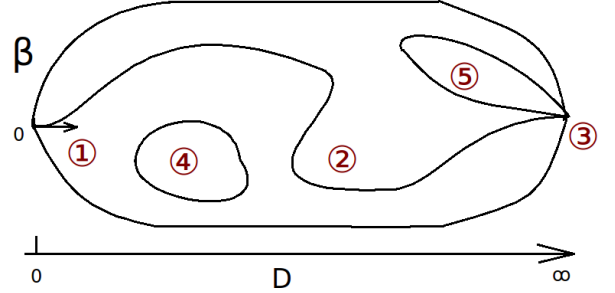


Figure 2: Illustration of various aspects of the regularization path. 1) At $D = \|\beta\|_1 = 0$, the path has one fixed direction; 2) The path sometimes travels backwards in D ; 3) When the path is not intersecting itself and we have fixed the region, it must reach the endpoint where it converges to an unconstrained solution; 4) There might be other paths but they must be circles, since line segments have to start at 0, one of the only two endpoints; 5) The circular path can also be initiated from the endpoint.

fold of solutions cannot be a circle, we make sure that the path starts at 0, in a single direction. When this holds, it cannot be a circle, since a circle will imply two directions to choose from at each point. Interestingly, with our L_1 constraint, it is easy to show that the point at $D = 0$ would have this property, subject to very weak assumptions.

Lemma 1 Assume the kernel function is C^3 smooth and the kernel matrix is always full-rank. For any C there exists a unique Lagrange multiplier λ_0 for the optimization problem (1). Set $\alpha_0 = (HK_0 + \lambda_0 I)^{-1}y$, if there exists a unique i with $\left|\alpha_0^T \left[\frac{\partial K}{\partial \beta_i}\right]_{\beta=0} \alpha_0\right| = \max_j \left|\alpha_0^T \left[\frac{\partial K}{\partial \beta_j}\right]_{\beta=0} \alpha_0\right|$, the path starts at $\mathbf{0}$ with only β_i turning non-zero with a fixed sign.

Proof: At $\beta = 0$, problem (1) becomes convex and a positive Lagrange multiplier always exists. From the condition in the Lemma 1, we have that $\left|\frac{\partial L}{\partial \beta_i}\right| = |\lambda_0 \alpha^T \left[\frac{\partial K}{\partial \beta_i}\right]_{\beta=0} \alpha|$ is maximal, which means β_i should be the first item in the active set, by Theorem 1. \square

Clearly, Lemma 1 almost always holds. Even if it does not, we can still slightly perturb the data to make it hold. Next we prove that the trajectory does not stop when a feature is eliminated from the active set. This

relies on the non-degeneracy of the Jacobian.

Lemma 2 *Assume the conditions in Lemma 1. For (α, β) that satisfies the KKT system (5), suppose the active set is $\mathcal{P} \cup \{k\}$, but k is just eliminated from the active set. So for $i \in \mathcal{P}, \beta_i \neq 0$ and for $i \in \mathcal{P}, \beta_i = 0$. Let $\eta = |\frac{\partial L}{\partial \beta_i}|, i \in \mathcal{P}$. If the Jacobian matrix:*

$$J = \begin{bmatrix} HK(\beta) + \lambda I & H \frac{\partial K}{\partial \beta} \alpha \{\mathcal{P}, k\} & \alpha \\ \alpha^T \frac{\partial K}{\partial \beta} \{\mathcal{P}, k\} & \partial^2 K \{\mathcal{P}, k\} & 0 \\ 2\alpha^T K & \alpha^T \frac{\partial K}{\partial \beta} \alpha \{\mathcal{P}, k\} & 0 \end{bmatrix} \quad (6)$$

is full rank, then (α, β) cannot be the end point of a KKT path. Here $\frac{\partial K}{\partial \beta} \alpha \{\mathcal{P}, k\}$ extracts only the entries $[\frac{\partial K}{\partial \beta_i}] \alpha$ with index $i \in \mathcal{P} \cup \{k\}$, $\partial^2 K \{\mathcal{P}, k\} = [\alpha^T \frac{\partial^2 K}{\partial \beta_i \partial \beta_j} \alpha]$, with index $i, j \in \mathcal{P} \cup \{k\}$.

Proof: Adding $|\alpha^T \frac{\partial K}{\partial \beta_j} \alpha| = \phi$ to (5), we obtain:

$$\begin{aligned} (HK(\beta) + \lambda I)\alpha &= y; \alpha^T K \alpha = C \\ \left| \alpha^T \frac{\partial K(\beta)}{\partial \beta_i} \alpha \right| &= \eta, i \in \mathcal{P}; \left| \alpha^T \frac{\partial K}{\partial \beta_j} \alpha \right| = \phi \end{aligned} \quad (7)$$

At an event point, we start with $\phi = \eta \neq 0$, hence we can drop the absolute sign in a small neighborhood of η and ϕ without loss of generality. The last equation can be eliminated by subtracting it from all previous ones that contain $|\alpha^T \frac{\partial K(\beta)}{\partial \beta_i} \alpha|$, to obtain

$$\begin{aligned} (HK(\beta) + \lambda I)\alpha &= y; \alpha^T K \alpha = C \\ \alpha^T \left(\frac{\partial K}{\partial \beta_i} - \frac{\partial K}{\partial \beta_j} \right) \alpha &= \eta - \phi, i = 1, \dots, p \end{aligned} \quad (8)$$

Taking $\eta - \phi$ as variable, this becomes a nonlinear system of $(\alpha, \beta, \lambda, \eta - \phi)$. Suppose (a, b, c, d) span the null space. If the matrix J is non-degenerate, the Jacobian on (α, β, λ) for (8) is also non-degenerate since it is a linear transform of J . Then $(a, b, c, 0)$ cannot be in the null space, so $d \neq 0$. Of the two possible directions (a, b, c, d) and $(-a, -b, -c, -d)$, there must be one and only one with $d > 0$. Taking a small step along it will make $\eta > \phi$ so that, the path continues in that direction. Therefore (α, β) cannot be the endpoint in this case. \square

Finally, we show that for a controlled set of kernels, the trajectory converges eventually.

Theorem 2 *Suppose that the condition for Lemma 1 and 2 holds, further assume the Jacobian (6) is full-rank everywhere. If there exists a positive number M , such that for every β , $K(\beta) \preceq MI$ and for $i = 1, \dots, d$, $\frac{\partial K}{\partial \beta_i} \preceq MI$, $\lim_{\|\beta\|_1 \rightarrow \infty} \frac{\partial K}{\partial \beta_i} = \mathbf{0}$, then the regularization path continues until $\max_i |\frac{\partial L}{\partial \beta_i}| = 0$.*

Proof: First we give the KKT system of (1) with slack variables ξ that come from the L_1 constraint added to the system:

$$\begin{aligned} (HK(\beta) + \lambda I)\alpha &= y; -\lambda \alpha^T \frac{\partial K}{\partial \beta_i} \alpha + \xi_i = \eta \\ \beta_i \xi_i &= 0, i = 1, \dots, p; \alpha^T K(\beta) \alpha = C \end{aligned}$$

This is a piecewise smooth system, with the only singular points at events $\beta_i = \xi_i = 0$. Note that aside from events, either β_i or $\xi_i \neq 0$. It is easy to infer that the Jacobian of the augmented system is full-rank, from the non-degeneracy of the original Jacobian (6).

From Lemma 2, we know that the path does not stop at event **E2**. And it is trivial that the path does not stop at event **E1**. Therefore, the behavior near the singular point is fixed: in the case of **E1**, the trajectory goes from $\beta_i = 0, \xi_i = \epsilon$ to $\beta_i = \epsilon, \xi_i = 0$ (without loss of generality since the sign of β_i is decided from Theorem 1), and vice versa in the case of **E2**. Standard approximation theorems in differential topology (Hirsch, 1994) indicate that, there exists a C^3 smooth surrogate function to $\beta_i \xi_i = 0$ around $\beta_i = \xi_i = 0$. Using that to replace $\beta_i \xi_i = 0$, we remove the singular points and obtain a smooth nonlinear system.

Then, from non-degeneracy of the augmented Jacobian, an application of the implicit function theorem and the 1-D manifold classification theorem (Milnor, 1978) (p.16, Lemma 1 and Appendix, Theorem 1) shows that the smoothed trajectory is a curve that does not self-intersect. Furthermore, the curve is not homeomorphic to a circle since there is only one route to go at $\beta = 0$, from Lemma 1. Therefore, the trajectory must be homeomorphic to a line segment.

From (3), we have $\alpha^T K(\beta) \alpha = y^T (HK + \lambda I)^{-1} K (HK + \lambda I)^{-1} y = C$. Since $K \succ 0$ and $K \preceq MI$, it can be shown that for $C > 0$, there exists a λ_0 with every $\lambda \leq \lambda_0$. Together with the condition $\frac{\partial K}{\partial \beta_i} \preceq MI$, we know that the trajectory is bounded. The only stopping condition we have is that $\max_i |\frac{\partial L}{\partial \beta_i}| = 0$, therefore the trajectory either stops at some finite $\|\beta\|_1$, or extends to $\|\beta\|_1 \rightarrow \infty$. Since $\lim_{\|\beta\|_1 \rightarrow \infty} \frac{\partial K}{\partial \beta_i} = \mathbf{0}$ and $\lambda \leq \lambda_0$, the trajectory must converge.

Finally, it is simple to see for each fixed $(\alpha, \beta, \lambda, \eta)$, ξ is unique. Thus the projection from $(\alpha, \beta, \lambda, \eta, \xi)$ to $(\alpha, \beta, \lambda, \eta)$ is homeomorphic, which means the smooth trajectory in $(\alpha, \beta, \lambda, \eta, \xi)$ is projected to a smooth trajectory in $(\alpha, \beta, \lambda, \eta)$. \square

The main condition for the kernel is $\lim_{\|\beta\|_1 \rightarrow \infty} \frac{\partial K}{\partial \beta_i} = \mathbf{0}$. This means that the kernel function varies less with β as $\|\beta\|$ increases. The ARD-Gaussian kernel described in the introduction satisfies this condition.

Another example is the normalized ARD-polynomial kernel $k(x, z) = \left(\frac{\sum \beta_i x_i z_i}{\sum \beta_i + 1} + 1\right)^d$. Jacobian conditions are hard to verify, but in our experiments we observed no degenerate Jacobians.

Suppose the regularization path stops with $\|\beta\|_1 = D_0$. Since the path is continuous, it passes through at least one local optimum of the problem (1) for every $D < D_0$. A natural question is, when will this path find the global optimum? In the experiments we implemented an LBFSGS-B method on the same problem and never found any better local optimum than the ones on the path. But theoretically, we are still not able to guarantee it. Since in Lemma 1 we have made sure that at 0 the path only extend in one direction, the only possible cases that some global optima fall outside the path are marked as (4) and (5) in Fig. 2. A preliminary result is that if these do not occur, we recover all the optima.

Corollary 1 *Suppose the condition for Theorem 2 holds. For a null direction $v = (v_\alpha^T, v_\beta^T, v_\lambda^T)^T$ with $Jv = 0$, if $v_\beta^T \text{sign}(\beta) \neq 0$ everywhere, the regularization path starting from 0 captures all the optima of (1).*

Proof: The condition means that the path is strictly monotone in D . Using the arguments in the proof of Theorem 2, from every local optimum at $D = D_0$, we are able to find a path that is strictly monotonically decreasing in D . Then since it does not stop anywhere according to Lemma 2, it will reach 0 in the end, coinciding with our solution path starting from 0. \square

The strict monotonicity condition is rather restrictive, basically meaning that the path cannot turn back in D . This may not hold for many complicated problems. Even in the synthetic experiments shown in Fig. 3 the path does turn back. But this still provides a condition to get the global optimum without assuming convexity. Improving on this result should be possible and interesting.

3 Tracing Nonlinear Systems

Bach et al. (2004) and Rosset (2005) used a prediction-correction algorithm to traverse nonlinear solution paths. However their algorithm relies on fixed ϵ -step sizes. Since ϵ must be very small to guarantee convergence, the algorithm is not as efficient for non-convex problems. In this paper, we propose a new ODE method, L_1 KR, to obtain the regularization path.

To follow the solution path produced by (5), we consider α , β and λ as functions of time (t). Note that the t mainly characterizes the arc length of the solution path. It is different from the D we used in the

last section to characterize the amount of regularization. For every t , D could always be computed by taking the L_1 norm of $\beta(t)$. Representing the system (5) as $F(\alpha(t), \beta(t), \lambda(t)) = 0$ and taking derivatives w.r.t. t , we obtain (Nocedal & Wright, 2006):

$$\frac{\partial F(\alpha, \beta, \lambda)}{\partial \alpha} \frac{d\alpha}{dt} + \frac{\partial F(\alpha, \beta, \lambda)}{\partial \beta} \frac{d\beta}{dt} + \frac{\partial F(\alpha, \beta, \lambda)}{\partial \lambda} \frac{d\lambda}{dt} = 0,$$

which means that the direction $\left(\frac{d\alpha}{dt}, \frac{d\beta}{dt}, \frac{d\lambda}{dt}\right)$ lies in the null space of the system Jacobian $J = \begin{bmatrix} \frac{\partial F(\alpha, \beta, \lambda)}{\partial \alpha} & \frac{\partial F(\alpha, \beta, \lambda)}{\partial \beta} & \frac{\partial F(\alpha, \beta, \lambda)}{\partial \lambda} \end{bmatrix}$.

There is one more variable than equations here, so if the Jacobian (6) has full rank, the null space always has dimension 1 and the direction is fixed up to a sign flip. The null direction v_t is obtained by QR decomposition of the matrix J^T , taking the last row of the orthogonal matrix Q and normalizing it to unit norm. The sign is chosen to have positive inner product with the previous one. When **E2** happens, we probe on both directions and choose the one that satisfies Theorem 1. In the probe we have some heuristic rules to cope with round-off errors. In most cases, these rules are guaranteed to find the right direction.

Since for every (α, β, λ) , we can obtain $\left(\frac{d\alpha}{dt}, \frac{d\beta}{dt}, \frac{d\lambda}{dt}\right)$ using the above procedure, identifying the regularization path becomes an initial value problem in standard form: $\left(\frac{d\alpha}{dt}, \frac{d\beta}{dt}, \frac{d\lambda}{dt}\right) = v_t$, and a classic ODE solver can provide results for many trajectory points (Nocedal & Wright, 2006). We use a classic Adams-Moulton predictor-corrector method to solve the ODE, which uses high-order function approximations in both the prediction and correction steps. This is different from Bach et al. (2004) and Rosset (2005), where prediction steps are based on linear function approximations.

In our algorithm, the solution path is traced by ODE until one of the events **E1** or **E2** occur. We do line search to find the configuration where an event occurs, then update the system by adding an equation for a new feature, or removing an equation for an eliminated feature that is eliminated from the active set. The line search is described in the next subsection.

Usually, we run the ODE solver with relatively low precision in order to increase speed. We need however, to make it precise at event points, otherwise the error will accumulate in the sequence of ODEs and may lead to convergence problems. At event points, we have one more equation to satisfy: e.g. when we include a feature k_{p+1} , we know that the additional equation:

$$\alpha^T \left(\frac{\partial K}{\partial \beta_{k_p}} - \frac{\partial K}{\partial \beta_{k_{p+1}}} \right) \alpha = 0 \quad (9)$$

needs to hold. We now have $n + p + 1$ variables and $n + p + 1$ equations, which gives an exact solution,

provided the Jacobian is non-degenerate. For this we solve the linear system:

$$\begin{bmatrix} \frac{\partial F}{\partial \alpha} & \frac{\partial F}{\partial \beta} & \frac{\partial F}{\partial \lambda} \end{bmatrix} \begin{bmatrix} \Delta \alpha \\ \Delta \beta \\ \Delta \lambda \end{bmatrix} = -F(\alpha, \beta, \lambda), \quad (10)$$

where F represents the system at the current (α, β, λ) , with $n+p+1$ equations including (9). The next iterate is given by $(\alpha + \Delta\alpha, \beta + \Delta\beta, \lambda + \Delta\lambda)$ and the process continues until the desired precision is reached. Note that this correction procedure is not needed if the ODE solver is run at high resolution, and it is only called on events. Therefore the number of calls is much smaller than the number of calls to Jacobians (< 100 for the entire path in our experiments).

3.1 Line Search

To find the point where the next event occurs, we use a line search that combines binary splits and interpolation. Starting at an event point, we run a small step ϵ to probe for the function variation. Denote $d(\epsilon) = -\alpha^T \frac{\partial K}{\partial \beta} \alpha \Big|_{\epsilon}$ the derivative vector, $\eta(0) = \max d(0)$, and $\eta(\epsilon) = \max d(\epsilon)$. We use linear interpolation to estimate the event point for each feature:

$$q_i = \frac{[\eta(0) - d(0)]_i \epsilon}{[d(\epsilon)]_i - d(0)_i - [\eta(\epsilon) - \eta(0)]}$$

We take the step size $q = \min_i q_i$ in order to advance to the nearest event. When the interpolation fails, we either resort on bisection between the known maximal point (after an event occurrence) and minimal point (where no event have occurred), or use a step size three times the last one if we do not know any maximal point. Higher-order interpolation may be used to improve accuracy, but one must also factor in the increased computational costs. In the line search, if any probed point has already been computed, we directly use the cached solution. Each line search takes less than 20 ODE calls in our experiments.

3.2 The L_1 KR Algorithm

The complete L_1 KR algorithm is outlined in the plate referred as Algorithm 1. In the algorithm, a variable order ODE solver selects the step size automatically. More Jacobian calls would be allocated to critical regions and less to regions of slow variation (Fig. 3(c)). This makes the method both faster and more reliable.

Denoting with p the size of the active set, the main computational effort comes from $O(n^2 p^2)$ to compute the Jacobian (6), $O((n+p)^3)$ to solve for the null direction and the correction system (10), and $O(n^2 d)$ for linear interpolation. The overall computational complexity is $O(n^3 + p^3 + n^2 p^2 + n^2 d)$. It has only linear

dependence on the dimension d of the training set. Thus it is preferable to be used to select a few features from a dataset with lots of putative features. Its time complexity is lower than HKL (Bach, 2009), a method designed for optimal feature selection. HKL has time complexity of $O(pn^3 + n^2 p^2)$, and since in HKL, combinations of features are used as base kernels, the size of its active set – for the same number of features – is usually much larger than L_1 KR.

Algorithm 1 The algorithm for tracking the path combines ODE, and correction at events.

input : Training set X , training response y , regularization parameter λ , $\epsilon = 10^{-3}$, $\gamma = 1$, $l = 0.1$.
 For $\beta(0) = 0$, solve (3) with $\alpha^T K \alpha = C$ for $\alpha(0)$ and λ .
 Find $i = \arg \max_j \left| \alpha(0)^T \frac{\partial K}{\partial \beta_j} \alpha(0) \right|$.
 Line search on β_i to find the first event point (α, β, λ) , where index j has $\left| \alpha^T \frac{\partial K}{\partial \beta_i} \alpha \right| = \left| \alpha^T \frac{\partial K}{\partial \beta_j} \alpha \right|$.
 Add i, j to the active set \mathcal{A} .
while $\lambda \max_i \left| \alpha_0^T \frac{\partial K}{\partial \beta_i} \alpha_0 \right| > \epsilon$ **do**
 Line search with ODE to find the break point t_p .
 Set $\alpha = \alpha(t_p), \beta = \beta(t_p), \lambda = \lambda(t_p)$.
 if $j \in \mathcal{A}$ has $j = \arg \max_i \left| \alpha^T \frac{\partial K}{\partial \beta_i} \alpha \right|$ **then**
 Add j to the active set \mathcal{A} . {E1 occurred.}
 end if
 if an indice j has $\beta_j = 0$ **then**
 Drop j from the active set \mathcal{A} . {E2 occurred.}
 end if
 while $\|H(\alpha, \beta)\|_1 > \epsilon$ **do**
 Run the correction step (10) to get $(\Delta\alpha, \Delta\beta, \Delta\lambda)$.
 $\alpha = \alpha + \Delta\alpha, \beta = \beta + \Delta\beta, \lambda = \lambda + \Delta\lambda$
 end while
 if E2 occurred **then**
 Run the ODE for time length l in both directions to find the valid direction.
 end if
end while

4 Experiments

We report experiments on both ARD-Gaussian kernels and ARD-polynomial kernels. β needs to be positive to ensure the positive definiteness of the kernel. This changes the first condition in Theorem 1 to $\beta_i \neq 0 \Rightarrow \frac{\partial L}{\partial \beta_i} = \max_j \left(-\frac{\partial L}{\partial \beta_j} \right)$. We compare our method with the HKL algorithm of (Bach, 2009), which also selects features based on a KRR optimization objective. Another comparison is done against standard kernel ridge regression (KRR), without optimizing β . We did not compare with some other feature weighting methods based on SVMs, because the difference in loss functions would produce difficulties in discriminating the factors that affect performance.

The algorithms are tested on synthetic and real datasets. The synthetic datasets are from (Friedman, 1991). The responses for the three datasets are: #1 : $f_1(x) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + x_5$; #2 : $f_2(x) = \sqrt{x_1^2 + (x_2 x_3 - \frac{1}{x_2 x_4})^2}$; #3 : $f_3(x) = \tan^{-1} \frac{x_2 x_3 - \frac{1}{x_2 x_4}}{x_1}$. For the synthetic datasets, we add 5 irrelevant uniformly distributed variables. We use two settings. The first is noise-free. The second has Gaussian noise on responses $f(x)$, scaled to make the SNR 10. We use 200 training examples and a hold-out of size 100 for testing. The results are averaged over 30 random trials. HKL and KRR use a Gaussian kernel, with width parameters selected by cross-validation.

The results are shown in Table 1. In both Friedman #1 and #2, we recovered the full set of relevant features in the noise-free setting, giving near zero residuals. In Friedman #3, we sometimes fail to retrieve feature x_4 (5 out of 30 times) but always succeed to retrieve the other 3. In the noisy settings, occasionally a few irrelevant features are selected to model the noise, and some relevant features are missed. The results of L_1KR are consistently better than HKL and KRR. HKL outperforms KRR in most cases, showing that it is selecting the relevant features, as well. However, it cannot automatically weight features as our method does. The performance differences might be accounted to this.

Table 1: MSE on synthetic datasets.

Data	L_1KR		HKL	KRR	LASSO
	Gauss	Poly			
#1 Clean	0.02	0.10	0.43	3.74	6.93
#1 Noisy	3.57	3.38	5.50	7.37	9.65
#2 Clean	0.02	0.00	0.05	7.10	16.27
#2 Noisy	10.91	11.05	13.70	19.49	27.54
#3 Clean	0.94	1.13	1.72	2.96	2.92
#3 Noisy	2.57	2.43	4.79	4.55	3.78

One solution path for the dataset Friedman #1 is shown in Figure 3(a). It can be seen that although it overfits to the noisy features 6-10 in the end, a large part of the path selects only the 5 real features (1-5). Also, the path is not monotone with D . The turning part is magnified in Figure 3(b). It can be seen that the path included some noisy features initially, but when the true feature 3 kicks in, the algorithm automatically backtracks, removing the noisy components and reducing the weight of other true features to compensate for the inclusion of feature 3. This interesting empirical behavior hints that forward-backward strategies like BLasso (Zhao & Yu, 2007) may also work for this problem. Research on designing efficient algorithms along these lines could be interesting.

We also plot the number of Jacobian calls against D in the same dataset as above (Fig. 3(c)). Overall, we obtain about 9100 solutions with 2883 Jacobian calls for

traversing the range $D = [0, 92.6]$. For the method in (Rosset 2004), this would amount to around $\epsilon = 0.03$, for the same number of Jacobian calls. But the solution path is highly fragile in our non-convex problem and a slight deviation may cause drifting to a bad local optimum. Our method automatically allocated around 60% of the Jacobian calls in the region with $D \leq 10$, where the solution path varies rapidly.

The results reported for real data (Table 2) are 10 fold cross-validated and averaged over 10 random splits. We compare our L_1KR algorithm with HKL, KRR, linear least squares (OLS) and linear LASSO. The results are shown in Table 2. It can be seen that L_1KR is usually on par or better than KRR and HKL. Besides, it selects fewer features than KRR, especially in the **Motif** dataset where it achieves the same result with only 77 out of 2155 features.

Interestingly, for **Diabetes**, L_1KR selected 8 features whereas linear LASSO selected all 10. The dropped features are 2 measurements from blood serum tests, described as LDL and TCH. The acronyms are hard to identify, but we assume that LDL may stand for *low-density lipoprotein* and TCH for *total cholesterol*, as commonly abbreviated. These two help to improve linear regression, but may have nonlinear dependency with feature 6: HDL, or *high-density lipoprotein* (both LDL and HDL are a part of total cholesterol). Our L_1KR algorithm appear to have captured the nonlinear relations with a smaller feature subset and safely discarded the remaining ones. These preliminary results show that L_1KR has a competitive feature selection capability, as compared with linear LASSO.

5 Conclusion

We proved the existence and uniqueness of a regularization path for L_1 feature selection/weighting problems in kernel methods. To follow the solution path in kernel ridge regression, we proposed the L_1KR algorithm, a homotopy method based on solving ODEs. The algorithm is shown to be effective for computation and feature selection. The path traverses through multiple local optima of the same regularization problem automatically, and avoid low-quality local solutions. For future work, it is worth studying whether – or under what additional conditions – the algorithm achieves global optimality. Generalizing the methodology to other kernel methods is also interesting.

Acknowledgements

This work was supported in part by the European Commission award MCEXT-025481 (FL and CS) and the NSFC (NSF China) grant 60835002 (FL).

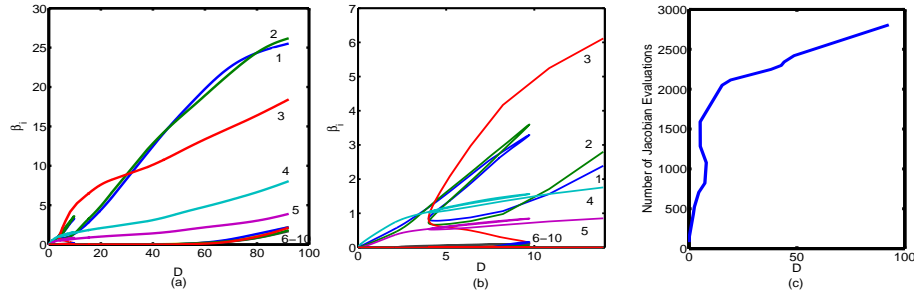


Figure 3: (Best viewed in color) (a) Solution path for β on Friedman #1 dataset. The solution overfits slightly towards the end, but runs for a long interval with the right features. (b) Detail of the initial part of the path: after feature 3 is added, the weights of other relevant ones automatically change to compensate. The algorithm drops noisy features automatically once feature 3 is included. (c) Number of Jacobian calls against D . In the initial part where the solution varies rapidly, significantly more Jacobian calls are made.

Table 2: MSE on real datasets. Cross-validation is used to select a point on the path and the MSE on this point from a separate test set is reported. Number inside the parentheses for L_1 KR and LASSO shows the number of selected features.

Dataset	# Features	L_1 KR		HKL	KRR	LASSO	OLS
		Gauss	Poly				
Diabetes	10	2981.61 (8)	2996.67 (8)	2923.80	2984.26	3007.10 (10)	3007.10
Boston	12	9.94 (12)	10.00 (12)	10.07	11.42	37.05 (12)	37.05
Wisconsin	31	1004.99 (11)	1004.17 (11)	1152.40	1007.94	983.91 (11)	1106.89
Motif	2155	0.16 (77)	0.16 (78)	0.29	0.16	0.13 (375)	0.16

References

Bach, F. (2008). Consistency of the group lasso and multiple kernel learning. *JMLR*, 9, 1179–1225.

Bach, F. (2009). Exploring large feature spaces with hierarchical multiple kernel learning. *NIPS21*.

Bach, F., Thibaux, R., & Jordan, M. I. (2004). Computing regularization paths for learning multiple kernels. *NIPS17*.

Bi, J., Bennett, K., Embrechts, M., Breneman, C. M., & Song, M. (2003). Dimensionality reduction via sparse support vector machines. *JMLR*, 3, 1229–1243.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, 46, 131–159.

Chow, S.-N., Mallet-Paret, J., & Yorke, J. A. (1978). Finding zeroes of maps: Homotopy methods that are constructive with probability one. *Mathematics of Computation*, 32, 887–899.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *Ann. Stat.*, 32, 407–499.

Friedman, J. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 19, 1–141.

Grandvalet, Y., & Canu, S. (2003). Adaptive scaling for feature selection in svms. *NIPS 15*.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46, 91–129.

Hirsch, M. W. (1994). *Differential topology*. Springer-Verlag.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *Ann. Stat.*, 36, 1171–1220.

Keerthi, S., Sindhwan, V., & Chapelle, O. (2007). An efficient method for gradient-based adaptation of hyperparameters in svm models. *NIPS 19*.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *JMLR*, 5, 27–72.

Lin, Y., & Zhang, H. H. (2006). Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34, 2272–2297.

Mangasarian, O., & Wild, E. (2007). Feature selection for nonlinear kernel support vector machines. *ICDM 2007*.

Milnor, J. W. (1978). *Topology from the differentiable viewpoint*. The University Press of Virginia.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization (2nd ed.)*. Springer-Verlag.

Osborne, M., Presnell, B., & Turlach, B. (2000). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20, 389–403.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press.

Rosset, S. (2005). Tracking curved regularized optimization solution paths. *NIPS 17*.

Varma, M., & Babu, B. R. (2009). More generality in efficient multiple kernel learning. *ICML 2009*.

Wang, G., Yeung, D.-Y., & Lochovsky, F. H. (2007). A kernel path algorithm for support vector machines. *ICML*.

Zhao, P., & Yu, B. (2007). Stagewise lasso. *JMLR*, 8, 2701–2726.