Robust and Efficient Kernel Hyperparameter Paths with Guarantees

Joachim Giesen JOACHIM.GIESEN@UNI-JENA.DE
Sören Laue SOEREN.LAUE@UNI-JENA.DE
Patrick Wieschollek PATRICK@WIESCHOLLEK.INFO
Friedrich-Schiller-Universität Jena, Germany

Abstract

Algorithmically, many machine learning tasks boil down to solving parameterized optimization problems. The choice of the parameter values in these problems can have a significant influence on the statistical performance of the corresponding methods. Thus, algorithmic support for choosing good parameter values has received quite some attention recently, especially algorithms for computing the whole solution path of a parameterized optimization problem. These algorithms can be used, for instance, to track the solution of a regularized learning problem along the regularization parameter path, or for tracking the solution of kernelized problems along a kernel hyperparameter path. Since exact path following algorithms can be numerically unstable, robust and efficient approximate path tracking algorithms have gained in popularity for regularized learning problems. By now algorithms with optimal path complexity in terms of a guaranteed approximation error are known for many regularized learning problems. That is not the case for kernel hyperparameter path tracking algorithms, where the exact path tracking algorithms can also suffer from numerical problems. Here we address this problem by devising a robust and efficient path tracking algorithm that can also handle kernel hyperparameter paths. The algorithm has asymptotically optimal complexity. We use this algorithm to compute approximate kernel hyperparamter solution paths for support vector machines and robust kernel regression. Experimental results for these problems applied to various data sets confirm the theoretical complexity analysis.

1. Introduction

Parameterized optimization problems of the form

$$\min_{x \in F_t} f_t(x)$$

are abundant in machine learning. Here $t \in \mathbb{R}$ is a parameter, $f_t : \mathbb{R}^d \to \mathbb{R}$ is some function depending on t, and $F_t \subseteq \mathbb{R}^d$ is the feasible region of the optimization problem at parameter value t.

The solution path problem is to compute an optimal or approximate solution $x_t \in F_t$ of the parameterized problem along some parameter interval $I \subseteq \mathbb{R}$. From the solution path a good parameter value t and a corresponding solution x_t can be chosen by some optimization criterion that should not be confused with the objective of the parameterized optimization problem. In a machine learning context the parameter t is typically optimized using some measure for the generalization error on test data while x_t is computed from training data.

An important example of the abstract parameterized optimization problem is

$$f_t(x) = r(x) + t \cdot l(x),$$

where $l : \mathbb{R}^d \to \mathbb{R}$ is a loss function and $r : \mathbb{R}^d \to \mathbb{R}$ is some regularizer, e.g. Euclidean regularization $r(x) = \|x\|_2^2$ that enables the so-called kernel trick, or $r(x) = \|x\|_1$ that encourages sparse solutions. This case, namely efficiently computing robust regularization paths, has received considerable attention and can be considered solved for the relevant problems in machine learning even when optimizing over positive-semidefinite matrices. Another important example that has received less attention is when f_t is given as a function $f : \mathbb{R}^d \to \mathbb{R}$ that is parameterized by a positive kernel function

$$k_t : \Omega \times \Omega \to \mathbb{R}$$

that itself is parameterized by $t \in \mathbb{R}$ on some set Ω.

Here we study a fairly general class of parameterized convex optimization problems that contains most of the regularization path and kernel hyperparameter path problems.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

We consider problems of the form

$$\begin{aligned} \min_{x\in\mathbb{R}^d} \quad & f_t(x) \\ \text{s.t.} \quad & c_t(x) \le 0, \end{aligned} \tag{1}$$

where $f_t : \mathbb{R}^d \to \mathbb{R}$ is convex and $c_t : \mathbb{R}^d \to \mathbb{R}^n$ is convex in every component

$$c_t^i : \mathbb{R}^d \to \mathbb{R},\ i = 1, \ldots, m$$

for all values of t. We assume that $f_t(x)$ and $c_t(x)$ are Lipschitz continuous in t at any feasible point x, but we do not require convexity (or concavity) of these functions in t. The feasible region at t is given as

$$F_t \;=\; \left\{x \in \mathbb{R}^d \,\middle|\, c_t(x) \le 0\right\},$$

with componentwise inequalities. Our goal in this paper is to devise a robust and efficient algorithm for computing an ε-approximate solution path for Problem (1), i.e., in contrast to the exact solution path problem we only aim for an ε-approximate solution along the parameter interval instead of an exact solution. Turning to approximate solutions leads to much more efficient and robust algorithms than the known exact solution paths algorithms.

Related work and contributions. Regularized optimization methods are in widespread use throughout machine learning. Thus, computing regularization paths has received considerable attention over the last years. The work on regularization paths started with the seminal work by (Efron et al., 2004) who observed that the regularization path of the LASSO is piecewise linear. In (Rosset & Zhu, 2007) a fairly general theory of piecewise linear regularization paths has been developed and exact path following algorithm have been devised. Important special cases are support vector machines whose regularization paths have been studied in (Zhu et al., 2003; Hastie et al., 2004), support vector regression (Wang et al., 2006b), where also the loss-sensitivity parameter can be tracked, and the generalized LASSO (Tibshirani & Taylor, 2011). From the beginning it was known, see for example (Allgower & Georg, 1993; Hastie et al., 2004; Bach et al., 2004), that exact regularization path following algorithms suffer from numerical instabilities as they repeatedly need to invert a matrix whose condition number can be poor, especially when using kernels. It also turned out (Gärtner et al., 2012; Mairal & Yu, 2012) that the combinatorial (and thus also computational) complexity of exact regularization paths can be exponential in the number of data points. This triggered the interest in approximate path algorithms (Rosset, 2004; Friedman et al., 2007). By now numerically robust, approximate regularization path following algorithms are known for many problems including support vector machines (Giesen et al., 2012b;c), the LASSO (Mairal & Yu, 2012), and regularized matrix factorization and completion problems (Giesen et al., 2012a;c). These algorithms compute a piecewise constant approximation with $O\big(1/\sqrt{\varepsilon}\big)$ segments, where $\varepsilon > 0$ is the guaranteed approximation error. Notably, the complexity is independent of the number of data points and even matching lower bounds are known (Giesen et al., 2012c).

The situation is still different for kernel hyperparameter path tracking. Exact kernel path tracking algorithms are known for kernelized support vector machines (Wang et al., 2007b), the kernelized LASSO (Wang et al., 2007a), and Laplacian-regularized semi-supervised classification (Wang et al., 2006a; 2012). The exact kernel path tracking algorithms are even more prone to numerical problems than regularization path tracking algorithms since they repeatedly need to invert a kernel matrix whose condition number tends to be poor (large), see Figure 1.

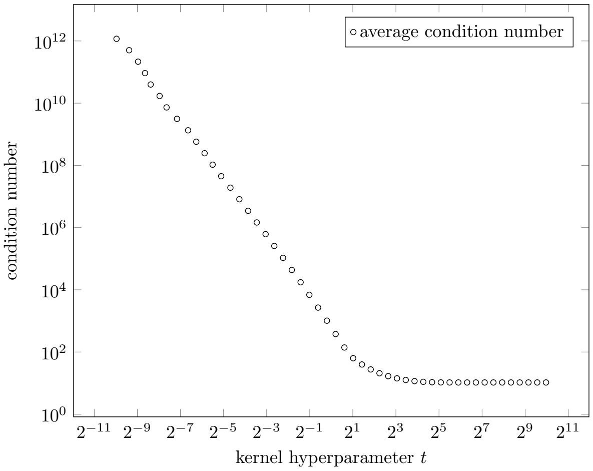

Figure 1. The condition number of the Gaussian kernel matrix $\big(k_t(x_i, x_j)\big) = \big(\exp(-t\|x_i - x_j\|_2^2)\big)$ of 100 data points drawn uniformly at random from $[0, 1]^{10}$ and for various values of t.

Here we address this problem by devising a numerically stable approximate solution path algorithm for parameterized problems of the Form (1). The algorithm can be used to compute approximate regularization paths as well as approximate kernel hyperparameter paths. We prove that the resulting path complexity is in $O(1/\varepsilon)$, where $\varepsilon > 0$ is again the guaranteed approximation error. This complexity might look disappointing considering that ε-approximation paths with complexity in $O\big(1/\sqrt{\varepsilon}\big)$ are known for many regularization path problems. Still, this is best possible. A matching lower bound of $\Omega(1/\varepsilon)$ has been first proved in (Giesen et al., 2010) for the class of Problems (1). This problem class includes problems, for instance kernel hyperparameter path problems, whose exact solution path is not piecewise linear as it is the case for the regularization path problems that exhibit a better approximation path complexity. We observed the $\Theta(1/\varepsilon)$ complexity bound also in experiments on various data sets for support vector machines and robust kernel regression that have been kernelized with a Gaussian kernel.

2. Duality and approximate solution paths

Since our approximate solution path algorithm is based on duality we review here some basic facts of duality theory for parameterized optimization problems. We also introduce our notation and define approximate solution paths for parameterized optimization problems and bound their complexity.

Lagrangian duality. The *Lagrangian* of the parameterized convex optimization problem (1) is the following function

$$\ell_t : \mathbb{R}^d \times \mathbb{R}^n_{\geq 0} \to \mathbb{R},\ (x, \alpha) \mapsto f_t(x) + \alpha^T c_t(x).$$

From the Lagrangian we can derive a *dual optimization problem* as

$$\begin{aligned} \max_{\alpha \in \mathbb{R}^n} \quad & \min_{x \in \mathbb{R}^d} \ \ell_t(x, \alpha) \\ \text{s.t.} \quad & \alpha \geq 0 \end{aligned}$$

We call

$$\hat{\varphi}_t : \mathbb{R}^n \to \mathbb{R},\ \alpha \mapsto \min_{x \in \mathbb{R}^d} \ \ell_t(x, \alpha).$$

the *dual objective function*. From the Lagrangian we can also derive an alternative expression for the primal objective function, namely

$$\varphi_t : \mathbb{R}^d \to \mathbb{R},\ x \mapsto \max_{\alpha \geq 0} \ \ell_t(x, \alpha)$$

Note that

$$f_t(x) \ = \ \varphi_t(x) \ \text{ for all } \ x \in F_t$$

since $\alpha^T c_t(x) \leq 0$ and thus $\max_{\alpha \geq 0} \ \alpha^T c_t(x) = 0$ (which can always be achieved by setting $\alpha = 0$) for all $x \in F_t$.

Weak and strong duality. At a fixed parameter value t we have the following well known *weak duality* property

$$\hat{\varphi}_t(\alpha) \ \leq \ \varphi_t(x)$$

for any $x \in \mathbb{R}^d$ and any $\alpha \in \mathbb{R}_{\geq 0}$. To see this note that

$$\min_{x' \in \mathbb{R}^d} \ \ell_t(x', \alpha) \ \leq \ \ell_t(x, \alpha)$$

for all $x \in \mathbb{R}^d$ and all $\alpha \in \mathbb{R}^n_{\geq 0}$. Thus,

$$\max_{\alpha' \geq 0} \min_{x' \in \mathbb{R}^d} \ \ell_t(x', \alpha') \ \leq \ \max_{\alpha' \geq 0} \ell_t(x, \alpha')$$

for all $x \in \mathbb{R}^d$, and finally

$$\max_{\alpha' \geq 0} \min_{x' \in \mathbb{R}^d} \ \ell_t(x', \alpha) \ \leq \ \min_{x' \in \mathbb{R}^d} \max_{\alpha' \geq 0} \ell_t(x', \alpha'),$$

which implies

$$\begin{aligned} \hat{\varphi}_t(\alpha) \ &\leq \ \max_{\alpha' \geq 0} \hat{\varphi}_t(\alpha') \\ &= \ \max_{\alpha' \geq 0} \min_{x' \in \mathbb{R}^d} \ \ell_t(x', \alpha') \\ &\leq \ \min_{x' \in \mathbb{R}^d} \max_{\alpha' \geq 0} \ell_t(x', \alpha') \\ &= \ \min_{x' \in \mathbb{R}^d} \ \varphi_t(x') \\ &\leq \ \varphi_t(x). \end{aligned}$$

In particular, we have $\hat{\varphi}_t(\alpha_t^*) \ \leq \ \varphi(x_t^*)$, where

$$\alpha_t^* \ = \ \operatorname{argmax}_{\alpha \geq 0} \hat{\varphi}_t(\alpha) \ \text{ and } \ x_t^* \ = \ \operatorname{argmin}_{x \in F_t} \varphi_t(x)$$

are the dual and primal optimal solutions, respectively. We say that *strong duality* holds if $\hat{\varphi}_t(\alpha_t^*) = \varphi_t(x_t^*)$.

Duality gap and approximate solution. At parameter value t we call

$$g_t(x, \alpha) \ = \ \varphi_t(x) - \hat{\varphi}_t(\alpha)$$

the *duality gap* at $(x, \alpha) \in F_t \times \mathbb{R}^n_{\geq 0}$. For $\varepsilon > 0$, we call $x \in F_t$ an *ε-approximate solution* of the parameterized optimization problem (1) at parameter value t, if

$$f_t(x) - f_t(x_t^*) \leq \varepsilon.$$

Assume that $g_t(x, \alpha) \leq \varepsilon$, then we have

$$\begin{aligned} f_t(x) - f_t(x_t^*) \ &= \ \varphi_t(x) - \varphi_t(x_t^*) \\ &= \ \varphi_t(x) - \hat{\varphi}_t(\alpha) + \hat{\varphi}_t(\alpha) - \varphi_t(x_t^*) \\ &= \ g_t(x, \alpha) - \big(\varphi_t(x_t^*) - \hat{\varphi}_t(\alpha)\big) \\ &\leq \ g_t(x, \alpha) \ \leq \ \varepsilon \end{aligned}$$

Approximate solution path. Let $[t_{\min}, t_{\max}] \subset \mathbb{R}$ be a compact parameter interval and $\varepsilon > 0$. We call a function

$$x : [t_{\min}, t_{\max}] \to \mathbb{R}^d,\ t \mapsto x_t$$

an *ε-approximate solution path* of the parameterized optimization problem (1), if for all $t \in [t_{\min}, t_{\max}]$

1. $x_t \in F_t$ and
2. $f_t(x_t) - f_t(x_t^*) \leq \varepsilon$.

We say that the path $x : [t_{\min}, t_{\max}] \to \mathbb{R}^d$ has *complexity* $k \in \mathbb{N}$, if x can be computed from k primal-dual optimal pairs $(x_{t_i}^*, \alpha_{t_i}^*)$ with $t_i \in [t_{\min}, t_{\max}),\ i = 1, \ldots, k$.

We can bound the complexity of approximate solution paths as follows.

Theorem 1. *Given a parameterized convex optimization problem (1), a parameter interval $[t_{\min}, t_{\max}]$, and $\varepsilon > 0$. If the following conditions*

1. *the feasible region F_t has a nonempty interior, and*
2. *the problem has an optimal solution $x_t^* \in F_t$, and*
3. *$\hat{\varphi}_t(\alpha)$ is Lipschitz continuous in t for any $\alpha \geq 0$, and*
4. *there exists a function*

   $$\tilde{x}_t : [t_{\min}, t_{\max}] \to \mathbb{R}^d$$

   such that $\tilde{x}_t(\tau)$ is feasible at parameter value τ, and

   $$\|\tilde{x}_t(\tau) - x_t^*\|_2 \;\leq\; L|\tau - t|$$

   for some constant $L > 0$,

are satisfied for all $t \in [t_{\min}, t_{\max}]$, then there exists an ε-approximate solution path for the interval $[t_{\min}, t_{\max}]$ whose complexity is in $O(1/\varepsilon)$. The constants in the big-O notation depend only on the functions f_t and c_t and on the interval $[t_{\min}, t_{\max}]$.

Proof. Let

$$r \;=\; \max_{t \in [t_{\min}, t_{\max}]} \|x_t^*\|_2.$$

Note that $r < \infty$ since x_t^* exists for all t in the compact interval $[t_{\min}, t_{\max}]$. Thus we can impose the additional constraint $\|x\|_2 \leq r$ in the parameterized convex optimization problem (1) without changing its solutions. Since the function f_t is convex on $\mathbb{R}^d$ it is also Lipschitz continuous with respect to it argument x for some constant $L' > 0$ on the set $\{x \in F_t \,|\, \|x\|_2 \leq r\}$.

By our assumptions both $f_t(x)$ and $\hat{\varphi}_t(\alpha)$ are Lipschitz continuous with respect to t for any feasible x and $\alpha \geq 0$, respectively, i.e., there exists a constant $M > 0$ such that

$$|f_t(x) - f_\tau(x)| \;\leq\; M|t - \tau|$$

and

$$|\hat{\varphi}_t(\alpha) - \hat{\varphi}_\tau(\alpha)| \;\leq\; M|t - \tau|$$

for all $t, \tau \in [t_{\min}, t_{\max}]$.

Note that $\tilde{x}_t(\tau)$ is a feasible solution for the primal problem at τ and α_t^* is a feasible solution for the dual problem at τ, because the feasible region of the dual problem does not depend on the parameter t. By Slater's Condition, see for example (Boyd & Vandenberghe, 2004), strong duality holds since F_t has a nonempty interior, and f_t and the components of c_t are convex functions, i.e., we have $g_t(x_t^*, \alpha_t^*) = 0$ for the duality gap.

Combining these properties we obtain a bound for the following duality gap

$$\begin{aligned} & g_\tau(\tilde{x}_t(\tau), \alpha_t^*) \\ &\quad = f_\tau(\tilde{x}_t(\tau)) - \hat{\varphi}_\tau(\alpha_t^*) \\ &\quad \leq f_\tau(x_t^*) + L'\|\tilde{x}_t(\tau) - x_t^*\| - \hat{\varphi}_\tau(\alpha_t^*) \\ &\quad \leq f_\tau(x_t^*) + L \cdot L'|\tau - t| - \hat{\varphi}_\tau(\alpha_t^*) \\ &\quad \leq f_t(x_t^*) - \hat{\varphi}_t(\alpha_t^*) + 2M|t - \tau| + L \cdot L'|t - \tau| \\ &\quad = g_t(x_t^*, \alpha_t^*) + (2M + L \cdot L')|t - \tau| \\ &\quad = (2M + L \cdot L')|t - \tau|, \end{aligned}$$

where the first inequality follows from the Lipschitz continuity of f_τ with respect to x, the second inequality follows from the Lipschitz continuity of the function $\tilde{x}_t$ with respect to τ, and the third inequality follows from the Lipschitz continuity of $f_t(x)$ and $\hat{\varphi}_t(\alpha)$ with respect to t for any feasible x and $\alpha \geq 0$, respectively. Hence, a primal-dual solution pair $(\tilde{x}_t(\tau), \alpha_t^*)$ is a feasible primal-dual ε-approximate solution pair for all τ with

$$|t - \tau| \;\leq\; \frac{\varepsilon}{2M + L \cdot L'}.$$

It follows that there exists an ε-approximate solution path whose complexity can be bounded by

$$\frac{2M + L \cdot L'}{\varepsilon}(t_{\max} - t_{\min}) \in O(1/\varepsilon).$$

□

The problem dependent constant in Theorem 1,

$$(2M + L \cdot L')(t_{\max} - t_{\min}),$$

might look huge at a first glance, but note that for an interval with $t_{\min} = 2^{-10}$ and $t_{\max} = 2^{10}$ it turns out that the value of this constant is at most 20 on the data sets that we have tried in our experiments for kernelized SVMs, see Section 6.

Lower bound The parameterized optimization problems in the lower bound construction in (Giesen et al., 2010) satisfy the conditions of Theorem 1. This gives us a lower bound in $\Omega(1/\varepsilon)$ on the path complexity for the class of Problems (1) and shows that the complexity analysis in Theorem 1 is asymptotically tight.

3. Approximate path tracking algorithm

In the following we assume that strong duality holds for all parameter values in the interval $[t_{\min}, t_{\max}] \subset \mathbb{R}$. The simple idea for computing an ε-approximate solution path makes use of duality and works as follows:

1. Compute the primal-dual pair (x_t^*, α_t^*) for $t = t_{\min}$.

2. Determine $\tilde{x}_t : [t_{\min}, t_{\max}] \rightarrow \mathbb{R}^d$ and $t' \in [t_{\min}, t_{\max}]$ such that
$$g_\tau(\tilde{x}_t(\tau), \alpha_t^*) \,\leq\, \varepsilon.$$
for all $\tau \in [t, t']$.

3. At t' compute a new optimal primal-dual pair $(x_{t'}^*, \alpha_{t'}^*)$ and iterate Steps 2 and 3 until the whole interval $[t_{\min}, t_{\max}]$ has been covered.

Let $t_{\min} = t_1, \ldots, t_k$ be the points in $[t_{\min}, t_{\max})$ at which an optimal primal-dual pair is computed (Step 3 of the algorithm). The path
$$x : [t_{\min}, t_{\max}] \rightarrow \mathbb{R}^d,\, t \mapsto \sum_{i=1}^{k} \mathbf{1}_{[t_1, t_{i+1})}(t)\, \tilde{x}_{t_i}(t),$$
where $t_{k+1} = t_{\max}$ and
$$\mathbf{1}_{[t_1, t_{i+1})}(t) \;=\; \begin{cases} 1, \text{ if } t \in [t_1, t_{i+1}) \\ 0, \text{ if } t \notin [t_1, t_{i+1}) \end{cases}$$
is an ε-approximate solution path of complexity k.

4. Application: Kernelized SVM

Here we specialize Theorem 1 and the approximate path algorithm to the standard hinge loss support vector machine (SVM) that has been kernelized with a Gaussian kernel matrix
$$K_t \;=\; \big(k_t(x, x')\big) \;=\; \big(\exp(-t\|x - x'\|_2^2)\big)$$
with bandwidth parameter $t > 0$. That is, we need to make sure that this SVM meets the necessary conditions of Theorem 1. The primal SVM problem is given as
$$\begin{aligned} \min_{w \in \mathbb{R}^d, b \in \mathbb{R}, \xi \in \mathbb{R}^d} \quad & \frac{1}{2} w^T K_t w + c \cdot \|\xi\|_1 \\ \text{s.t.} \quad & y \odot (K_t w + b) \geq 1 - \xi \\ & \xi \geq 0, \end{aligned}$$
where c is a regularization parameter, $y \in \mathbb{R}^d$ is a label vector with entries in $\{-1, +1\}$, and $\odot$ is the element-wise multiplication.

The dual SVM problem can be written as
$$\max_{\alpha \in \Delta_c} \; -\frac{1}{2}\alpha^T \left(K_t \odot yy^T\right)\alpha + \|\alpha\|_1 \;\big(= \hat{\varphi}(\alpha)\big),$$
where
$$\Delta_c \;=\; \big\{\alpha \in \mathbb{R}^d \,|\, 0 \leq \alpha \leq c,\, \alpha^T y = 0\big\}.$$

It is straightforward to see that Assumptions 1.-3. of Theorem 1 are satisfied for the SVM problem. It remains to ensure that also Assumption 4. holds true.

To ensure Assumption 4. we need to find a function as required in this assumption. Here we discuss two functions that satisfy the requirements. In the first function the bias b is fixed and in the second function it depends on the bandwidth parameter t'. We call the first case the *fixed bias update rule* and the second the *dynamic bias update rule*.

1. *Fixed bias updates.* For an optimal primal solution (w_t^*, b_t^*, ξ_t^*) at parameter value t we need to derive a solution $(\tilde{w}_t, \tilde{b}_t, \tilde{\xi}_t)(t')$ that is feasible at parameter value t'. Setting
$$\begin{aligned} &\Big(\tilde{w}_t, \tilde{b}_t, \tilde{\xi}_t\Big)(t') \\ &\quad = \big(w_t^*, b_t^*, \max\{1 - y \odot (K_{t'} w_t^* + b_t^*), 0\}\big) \end{aligned}$$
satisfies Assumption 4, i.e., the values of this function are by construction feasible for the primal SVM at any admissible value for t' and it is Lipschitz continuous in t' since the dependence of $K_{t'}$ on t' is differentiable. Note that $\tilde{b}_t(t') = b_t^*$, which justifies the name *fixed bias update rule*.

2. *Dynamic bias updates.* We can also adapt the bias b for each t' instead of only adapting ξ. This can be done by setting $\tilde{b}_t(t')$ to the median (for robustness reasons) of the expressions $(y - K_{t'} w_t^*)_i$, $i \in I$, where the index set
$$I \;=\; \{i \,|\, 0 < e_i^T \alpha_t^* < c\},$$
with e_i the i-th standard basis vector of $\mathbb{R}^d$, is the set of all support vectors that are exactly on the margin for the optimal dual SVM solution α_t^* at parameter value t. As for the fixed bias update rule, the entries of $\tilde{\xi}_t(t')$ are chosen such that all inequality constraints of the primal SVM problem are satisfied, i.e.,
$$\begin{aligned} &\Big(\tilde{w}_t, \tilde{b}_t, \tilde{\xi}_t\Big)(t') \\ &\; = \big(w_t^*, \tilde{b}_t(t'), \max\{1 - y \odot (K_{t'} w_t^* + \tilde{b}_t(t')), 0\}\big). \end{aligned}$$
Obviously, the dynamic bias update rule improves the primal objective function value over the fixed bias update rule and hence Theorem 1 applies here as well.

Asymptotically the complexity of the SVM kernel path is in $O(1/\varepsilon)$ in both cases since Theorem 1 applies. In practice, however, it makes a difference which of the two update rules is used, see Section 6. Although the asymptotic behavior is the same, the constants are much smaller for the dynamic bias update rule than for the fixed bias update rule.

5. Application: Robust Kernel Regression

Robust regression is an alternative to least squares regression that uses an ℓ_1-loss function instead of an ℓ_2-loss function to become more *robust* against outliers. Robust kernel

Table 1. Number of updates (complexity) of ε-approximate hyperparamter kernel path for some data sets and various values for ε for the fixed bias update rule (on the left) and the dynamic bias update rule (on the right).

		FIXED BIAS UPDATES						DYNAMIC BIAS UPDATES					
DATA SET	SIZE	$\varepsilon =$ 4	2	1	0.5	0.25	0.125	4	2	1	0.5	0.25	0.125
A1A	1605	4	8	15	26	47	89	2	5	9	17	32	61
A2A	2265	5	9	16	29	51	95	3	6	10	19	35	65
A3A	3185	6	11	19	33	59	108	4	7	13	22	40	74
A4A	4781	8	14	24	40	71	126	6	11	18	30	52	79
DIABETES	768	11	18	28	43	64	95	3	5	8	11	19	29
HEART	270	1	2	6	10	16	25	1	2	3	5	8	11
IONOSPHERE	351	10	18	31	49	81	132	2	3	7	12	20	33

regression is an extension of robust regression that accommodates the use of kernels for nonlinear regression. Here we even consider sparse robust kernel regression by adding an additional ℓ_1-regularizer that favors sparse solutions.

The sparse robust kernel regression problem is given as the following minimization problem

$$\min_{\beta \in \mathbb{R}^d} \quad \|y - K_t \beta\|_1 + \lambda \cdot \|\beta\|_1,$$

where $y \in \mathbb{R}^d$ is the output vector, and $K_t \in \mathbb{R}^{d \times d}$ is the kernel matrix that is determined by the parameterized Gaussian kernel function k_t and d data points $x_1, \ldots, x_d$. The regression function is then given as

$$f(x) = \sum_{i=1}^{d} \beta_i k_t(x_i, x).$$

The dual problem of the sparse robust kernel regression problem is the following maximization problem

$$\begin{aligned} \max_{u \in \mathbb{R}^d} \quad & -y^T u \\ \text{s.t.} \quad & \|u\|_\infty \leq 1 \\ & \|K_t^T u\|_\infty \leq \lambda. \end{aligned}$$

To apply Theorem 1 we interchange the role of the primal and the dual problem, i.e., we consider

$$\begin{aligned} \min_{u \in \mathbb{R}^d} \quad & y^T u \\ \text{s.t.} \quad & \|u\|_\infty \leq 1 \\ & \|K_t^T u\|_\infty \leq \lambda \end{aligned}$$

as the primal problem, whose dual is given as

$$\max_{\beta \in \mathbb{R}^d} \quad -\|y - K_t \beta\|_1 - \lambda \cdot \|\beta\|_1.$$

Obviously, all four conditions of Theorem 1 are met, since the primal problem has a nonempty interior, the problem is bounded (and hence a primal optimum exists), the dual function is Lipschitz continuous with respect to t, and for any optimal dual solution u_t^* we can find a feasible solution $\tilde{u}_t(\tau)$ by projecting u_t^* onto the feasible region at parameter value τ. Since K_t is differentiable in t (for a Gaussian kernel matrix) the projection itself is Lipschitz continuous. Hence, we can apply Theorem 1 that guarantees the existence of an ε-approximate hyperparameter solution path of complexity $O(1/\varepsilon)$.

6. Experiments

To validate our theoretical finding, in particular the dependence of the path complexity on the guaranteed approximation error ε, we have conducted experiments for the kernelized SVM and also for the robust kernel regression.

6.1. Kernelized SVM

We have implemented the approximate path tracking algorithm for the kernelized SVM. LIBSVM Version 3.17, whose implementation is described in (Fan et al., 2005), has been used to compute primal-dual optimal pairs. LIBSVM actually solves the dual problem. If α_t^* is the optimal dual solution at parameter value t, then the optimal primal solution can be reconstructed by setting $w_t^* = y \odot \alpha_t^*$ and b_t^* to the median of the expressions

$$(y - K_t w_t^*)_i, \; i \in \{j \,|\, 0 < e_j^T \alpha_t^* < c\}.$$

It remains to describe the implementation of the second step of the algorithm. Since we can compute the value of the primal objective function for every value of t', see Section 4, we can also compute the duality gap

$$g_\tau \left(\left(\tilde{w}_t, \tilde{b}_t, \tilde{\xi}_t \right)(\tau), \alpha_t^* \right).$$

The largest $t' > t$ for which the duality gap $g_{t'}$ is still at most ε can be simply found by binary search.

As test environment we used MATLAB, and all data sets that have been used in our experiments were retrieved from

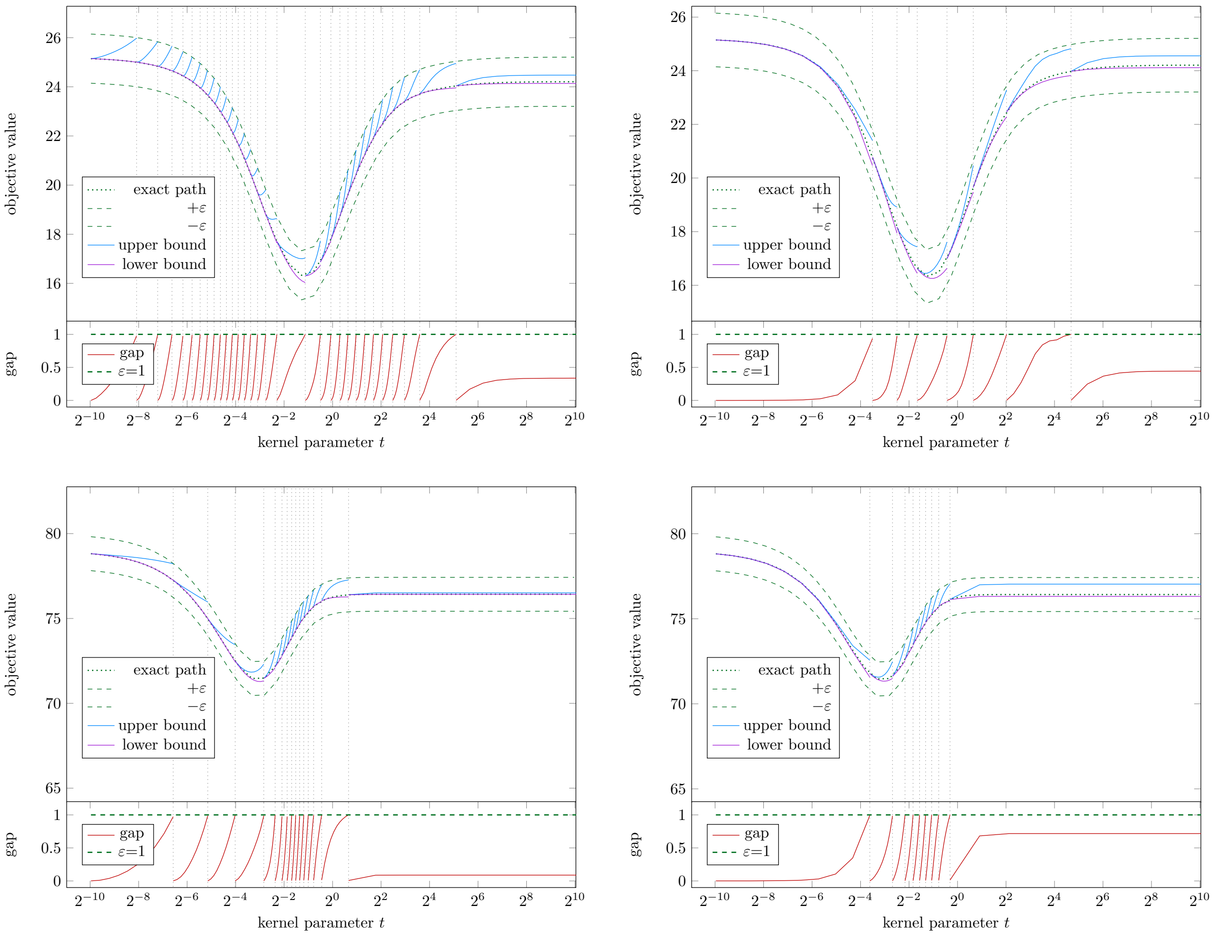

Figure 2. The kernel hyperparameter path for the IONOSPHERE data set with fixed bias updates (top, left) and dynamic bias updates (top, right), and the kernel hyperparameter path for the A1A data set with fixed bias updates (bottom, left) and dynamic bias updates (bottom, right).

the LIBSVM Website, see (Lin). The data sets and results are summarized in Table 1. The regularization parameter c was set to 0.1 in all the experiments.

Dependence on ε Our theoretical finding that $O(1/\varepsilon)$ optimal primal-dual pairs are sufficient to approximate the whole kernel hyperparameter solution path was confirmed in our experiments. Figure 3 indicates that the number of optimal primal-dual pairs computed by the algorithm depends linearly on $1/\varepsilon$.

Choice of bias update rule The experiments also show that the choice of the bias update rule has a significant influence on the approximation path complexity. As expected the dynamic bias update rule leads to a lower path complexity than the fixed bias update rule, since it improves the value of the primal objective function over the fixed bias update rule and thus needs fewer updates to maintain

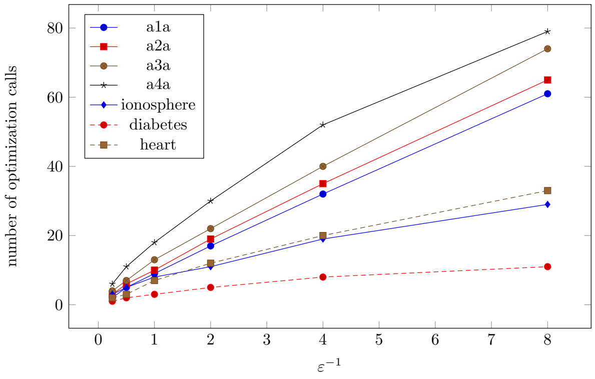

Figure 3. Number of optimal primal-dual pairs computed by the path tracking algorithm (path complexity) for various data sets at several values for $1/\varepsilon$.

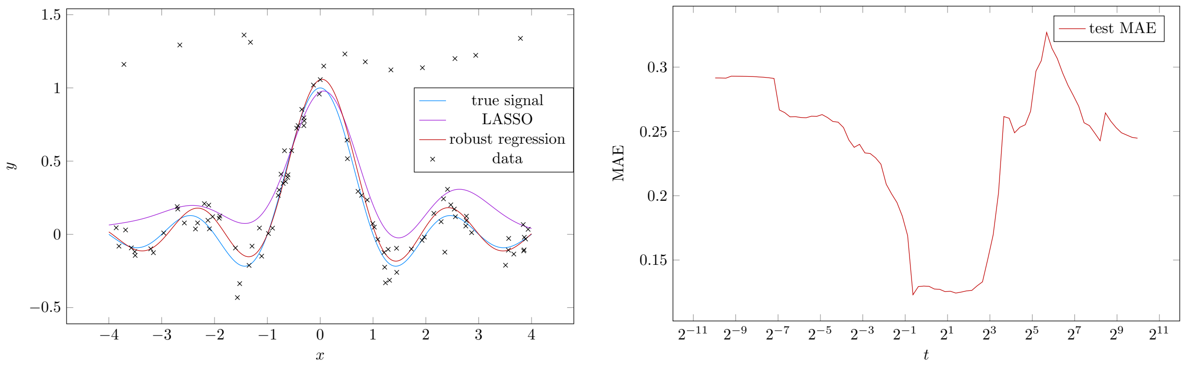

Figure 4. Synthetic data set for regression sampled with noise and outliers (on the left), and the mean absolute error (MAE) for the test data set (on the right).

the approximation guarantee. Figure 2 directly compares the two update rules and shows that indeed the dynamic bias update rule performs better. The main difference is that the upper bound, i.e., the approximation of the primal optimum, is much better for the dynamic bias update rule.

Figure 2 also shows that our path tracking algorithm, in contrast to a simple grid search, adapts well to regions of interest (especially for the dynamic bias update rule), i.e., the solution is only updated frequently in these regions.

6.2. Robust Kernel Regression

We have also implemented the approximate path tracking algorithm for robust kernel regression. The optimal primal-dual pairs at a fixed parameter value t have been computed using the SeDuMi solver (Sturm, 1999). The implementation of the second step of the algorithm is analogous to the implementation for the kernelized SVM since also here we can compute the value of the primal objective function for every value of t' and thus the duality gap $g_\tau\left(\tilde{u}_t(\tau), \beta_t^*\right)$. The largest $t' > t$ for which the duality gap $g_{t'}$ is still at most ε can be found by binary search.

As test environment we used again MATLAB and following the example of (Wang et al., 2007a) we generated a synthetic data set by randomly sampling 100 points from the following target function

$$f(x) = \frac{\sin(\pi x)}{\pi x}$$

in the interval $[-4, 4]$ and by adding Gaussian noise. Additionally, we also added 10% outliers to the data set. The data set, i.e., the sample points, and the target function are shown in Figure 4 (on the left). In this figure we also show that, as expected, robust regression performs better in the presence of outliers than for instance the LASSO (Tibshirani, 1994). The regularization parameter λ was set to 0.1 in the experiments.

It is well known that the choice of the bandwidth parameter in the Gaussian kernel has a significant influence on the performance of kernel regression methods. This can be seen also in Figure 4 (on the right), where we show the mean absolute error (MAE) on a set of test data points tracked along the kernel hyperparameter path (i.e., the bandwidth path). Note that the test error path in Figure 4 (on the right) has many local minima which is typical for this type of problems.

7. Conclusions

We have presented an algorithmic framework for tracking approximate solutions for a large class of parameterized optimization problems. In particular, the framework allows to track kernel hyperparameter paths and even has the optimal path complexity of $O(1/\varepsilon)$ in terms of the prescribed approximation error ε for this type of problems, for which no efficient approximation schemes had been devised before. The framework also allows to compute approximate regularization paths, but it is not optimal for this easier class of problems (whose exact solution path is piecewise linear which is not true for hyperparameter paths).

We have instantiated the algorithmic framework for computing approximate kernel hyperparameter paths for SVMs and the robust kernel regression problem, both with Gaussian kernel. Our experiments for these applications, in contrast to exact path algorithms, did not suffer from numerical problems, and confirmed our optimal theoretical complexity bounds.

Acknowledgments

This work has been supported by a grant of the Deutsche Forschungsgemeinschaft (GI-711/3-2).

References

Allgower, Eugene and Georg, Kurt. Continuation and path following. *Acta Numerica*, 2:1–64, 1993.

Bach, Francis R., Thibaux, Romain, and Jordan, Michael I. Computing regularization paths for learning multiple kernels. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

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

Efron, Bradley, Hastie, Trevor, Johnstone, Iain, and Tibshirani, Robert. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Fan, Rong-En, Chen, Pai-Hsuen, and Lin, Chih-Jen. Working Set Selection Using Second Order Information for Training Support Vector Machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

Friedman, Jerome, Hastie, Trevor, Höfling, Holger, and Tibshirani, Robert. Pathwise Coordinate Optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Gärtner, Bernd, Jaggi, Martin, and Maria, Clément. An Exponential Lower Bound on the Complexity of Regularization Paths. *Journal of Computational Geometry (JoCG)*, 3(1):168–195, 2012.

Giesen, Joachim, Jaggi, Martin, and Laue, Sören. Approximating Parameterized Convex Optimization Problems. In *European Symposium on Algorithms (ESA)*, pp. 524–535, 2010.

Giesen, Joachim, Jaggi, Martin, and Laue, Sören. Regularization Paths with Guarantees for Convex Semidefinite Optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 432–439, 2012a.

Giesen, Joachim, Jaggi, Martin, and Laue, Sören. Approximating parameterized convex optimization problems. *ACM Transactions on Algorithms*, 9(1):10, 2012b.

Giesen, Joachim, Müller, Jens K., Laue, Sören, and Swiercy, Sascha. Approximating Concavely Parameterized Optimization Problems. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2114–2122, 2012c.

Hastie, Trevor, Rosset, Saharon, Tibshirani, Robert, and Zhu, Ji. The Entire Regularization Path for the Support Vector Machine. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

Lin, Chih-Jen. LIBSVM Tools. Data sets available at www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

Mairal, Julien and Yu, Bin. Complexity analysis of the lasso regularization path. In *International Conference on Machine Learning (ICML)*, 2012.

Rosset, Saharon. Following curved regularized optimization solution paths. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

Rosset, Saharon and Zhu, Ji. Piecewise linear regularized solution paths. *The Annals of Statistics*, 35(3):1012–1030, 2007.

Sturm, Jos F. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

Tibshirani, Robert. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Tibshirani, Ryan and Taylor, Jonathan. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3): 1335–1371, 2011.

Wang, Gang, Chen, Tao, Yeung, Dit-Yan, and Lochovsky, Frederick H. Solution path for semi-supervised classification with manifold regularization. In *IEEE International Conference on Data Mining (ICDM)*, pp. 1124–1129, 2006a.

Wang, Gang, Yeung, Dit-Yan, and Lochovsky, Frederick H. Two-dimensional solution path for support vector regression. In *International Conference on Machine Learning (ICML)*, pp. 993–1000, 2006b.

Wang, Gang, Yeung, Dit-Yan, and Lochovsky, Frederick H. The Kernel Path in Kernelized LASSO. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 580–587, 2007a.

Wang, Gang, Yeung, Dit-Yan, and Lochovsky, Frederick H. A kernel path algorithm for support vector machines. In *International Conference on Machine Learning (ICML)*, pp. 951–958, 2007b.

Wang, Gang, Wang, Fei, Chen, Tao, Yeung, Dit-Yan, and Lochovsky, Frederick H. Solution Path for Manifold Regularized Semisupervised Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 42(2): 308–319, 2012.

Zhu, Ji, Rosset, Saharon, Hastie, Trevor, and Tibshirani, Robert. 1-norm Support Vector Machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.