Regularization Paths with Guarantees for Convex Semidefinite Optimization

Joachim Giesen
Friedrich-Schiller-University Jena

Martin Jaggi
ETH Zurich

Sören Laue
Friedrich-Schiller-University Jena

Abstract

We devise a simple algorithm for computing an approximate solution path for parameterized semidefinite convex optimization problems that is guaranteed to be ε-close to the exact solution path. As a consequence, we can compute the entire regularization path for many regularized matrix completion and factorization approaches, as well as nuclear norm or weighted nuclear norm regularized convex optimization problems. This also includes robust PCA and variants of sparse PCA. On the theoretical side, we show that the approximate solution path has low complexity. This implies that the whole solution path can be computed efficiently. Our experiments demonstrate the practicality of the approach for large matrix completion problems.

1 Introduction

Our goal is to compute the entire solution path, with a continuously guaranteed approximation quality, for parameterized convex problems over the convex domain of positive semidefinite matrices with unit trace, i.e., optimization problems of the form

$$\begin{array}{rl} \min\limits_{X\in\mathbb{R}^{n\times n}} & f_t(X) \\ s.t. & \mathrm{Tr}(X) = 1\ , \\ & X \succeq 0\ , \end{array} \qquad (1)$$

where f_t is a family of convex functions, parameterized by $t \in \mathbb{R}$, that is defined on symmetric matrices $n \times n$ matrices X.

Motivation and Applications. Parameterized optimization problems of the above form have applications in various areas such as control theory or multiobjective optimization. Our work here is mainly motivated by nuclear norm regularized optimization problems, which have become central to many applications in machine learning and compressed sensing, as for example low-rank recovery [Fazel et al., 2001, Candes and Recht, 2009, Candès and Tao, 2010], robust PCA [Candes et al., 2011], and matrix completion [Srebro et al., 2004, Rennie and Srebro, 2005, Webb, 2006, Lin, 2007, Koren et al., 2009, Takács et al., 2009, Salakhutdinov and Srebro, 2010], and where the right parameter selection is often a non-trivial task

Formally, our work is motivated by parameterized optimization problems of the form

$$\min_{Z\in\mathbb{R}^{m\times n}} f(Z) + \lambda\,\|Z\|_* \qquad (2)$$

for a convex function f (the loss function), where $\|.\|_*$ is the nuclear norm. The equivalent constrained formulation for these problems reads as

$$\begin{array}{rl} \min\limits_{Z\in\mathbb{R}^{m\times n}} & f(Z) \\ s.t. & \|Z\|_* \le \frac{t}{2} \end{array} \qquad (3)$$

Both problems are parameterized by a real regularization parameter, λ or t, respectively. To relate the nuclear norm regularized Problems (2) and (3) to semidefinite optimization, a straightforward transformation (see for example [Fazel et al., 2001, Srebro et al., 2004, Jaggi and Sulovský, 2010]) comes to help, which along the way also explains why the nuclear norm is widely called the *trace* norm: any problem of the form of Problem (3) is equivalent to optimizing

$$f_t(X) := \hat{f}\left(t\begin{pmatrix} V & Z \\ Z^T & W \end{pmatrix}\right) := f(tZ) \qquad (4)$$

over positive semidefinite $(m+n)\times(m+n)$-matrices X with unit trace, where $Z \in \mathbb{R}^{m\times n}$ is the upper right part of X and V and W are symmetric $(m\times m)$- and $(n\times n)$-matrices, respectively. Note that f_t is convex whenever f is convex.

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume 22 of JMLR: W&CP 22. Copyright 2012 by the authors.

Contributions. We provide an algorithm for tracking approximate solutions of parameterized semidefinite optimization problems in the form of Problem 1 along the entire parameter path. The algorithm is very simple but comes with strong guarantees on the approximation quality as well as on the running time. The main idea is to compute at a parameter value an approximate solution that is slightly better than the required quality, and then to keep this solution as the parameter changes exactly as long as the required approximation quality can still be guaranteed. Only when the approximation quality is no longer sufficient, a new solution again with slightly better than required approximation guarantee is computed. We prove that, if an approximation guarantee of $\varepsilon > 0$ is required along the entire path, then the number of necessary solution updates is only $O\big(\frac{1}{\varepsilon}\big)$, ignoring problem specific constants. We also argue that this number of updates is essentially best possible in the worst case, and our experiments demonstrate that often, the computation of an entire ε-approximate solution path is only marginally more expensive than the computation of a single approximate solution.

Our path tracking algorithm is not tied to a specific optimizer to solve the optimization problem at fixed parameter values. Any existing optimizer or heuristic of choice can be used to compute an approximate solution at fixed parameter values.

As a side-result we also show that weighted nuclear norm regularized problems can be optimized with a solid convergence guarantee, by building upon the rank-1 update method by [Hazan, 2008, Jaggi and Sulovský, 2010].

Related Work. For kernel methods and many other machine learning techniques, the resulting optimization problems often turn out to be parameterized convex quadratic programs, and in recent years a plenitude of algorithms and heuristics have been developed to "track" these solution paths, see for example [Hastie et al., 2004, Loosli et al., 2007, Rosset and Zhu, 2007]. However, the exact piecewise linear solution path of parameterized quadratic programs (in particular for the SVM) is known to be of exponential complexity in the worst case [Gärtner et al., 2010]. In the work here by contrast we show that just constantly many intervals of the parameter are sufficient for any fixed desired continuous approximation quality $\varepsilon > 0$. For a class of vector optimization problems, such as SVMs, [Giesen et al., 2010] have introduced this new approach of approximation paths of piecewise constant solutions.

To our best knowledge, no path algorithms are known so far for the more general case of semidefinite optimization. The solution path for sparse principal component analysis (PCA) was investigated by [d'Aspremont et al., 2007a], which however is parameterized over *discrete* integral values from 1 to n, where n is the number of variables. For a variant of low-rank matrix completion, [Mazumder et al., 2010] suggest to perform a grid search on the regularization parameter interval, computing a single approximate solution at each parameter t. However, they provide no approximation guarantee between the chosen grid points.

Notation. For matrices $A, B \in \mathbb{R}^{n\times m}$, the standard *inner product* is defined via the trace as

$$A \bullet B := Tr(A^T B),$$

and the (squared) *Frobenius matrix norm* is given by

$$\|A\|_F^2 := A \bullet A.$$

By $\mathbb{S}^{n\times n}$ we denote the set of *symmetric* $n \times n$ matrices. We write $\lambda_{\max}(A)$ for the largest eigenvalue of a matrix $A \in \mathbb{S}^{n\times n}$. A is called *positive semidefinite* (PSD), written as $A \succeq 0$, iff $v^T A v \geq 0 \ \forall v \in \mathbb{R}^n$. The (squared) *spectral norm* of $A \in \mathbb{R}^{n\times n}$ is defined as

$$\|A\|_2^2 := \lambda_{\max}(A^T A),$$

and the *nuclear norm* $\|A\|_*$ of $A \in \mathbb{R}^{n\times m}$, also known as the *trace norm*, is the sum of the singular values of A, or the ℓ_1-norm of the spectrum. Its well known relation to matrix factorization is that

$$\|A\|_* = \min_{UV^T = A} \frac{1}{2}(\|U\|_F^2 + \|V\|_F^2),$$

where the number of columns of U and V is not constrained [Fazel et al., 2001, Srebro et al., 2004].

2 The Duality Gap

The following notion of the *duality gap* is central to our discussion and path tracking algorithm. The gap can serve as a practical approximation quality measure for convex optimization problems in the form of Problem (1). In the following we assume that our convex objective function $f_t(X)$ is continuously differentiable[1]. We consider the gradient $\nabla f_t(X)$ with respect to X, which is a symmetric matrix in $\mathbb{S}^{n\times n}$.

Definition 1. *For Problem (1), the duality gap at any matrix $X \in \mathbb{S}^{n\times n}$ is defined as*

$$g_t(X) = \lambda_{\max}\left(-\nabla f_t(X)\right) + X \bullet \nabla f_t(X)$$

[1] If $f_t(X)$ is convex but not differentiable, the concepts of standard Lagrange duality can still be generalized for subgradients analogously, so that any element of the subgradient will give an upper bound on the approximation error.

An important property of the duality gap is that it always holds that

$$f_t(X) - f_t(X^*) \leq g_t(X)$$

for any feasible X and any optimal solution X^* of Problem (1) at parameter value t. This property, which follows from standard duality theory, see [Boyd and Vandenberghe, 2004, Section 5.9], or [Hazan, 2008], makes the duality gap an extremely useful measure of approximation quality and a stopping criterion for practical optimizers.

Definition 2. *Let $\varepsilon > 0$. A matrix $X \in \mathbb{S}^{n\times n}$ that is feasible for Problem (1) at parameter value t is called an* ε-approximation *if the duality gap satisfies*

$$g_t(X) \leq \varepsilon \ .$$

While the optimum value $f_t(X^*)$ is usually unknown, the gap $g_t(X)$ readily guarantees a simple upper bound on the current difference to this optimum value. The quantity g_t is easily computable for any candidate solution X even for very large problems, since it reduces to a single eigenvalue computation.

The gap g_t for Problem (1) can also be interpreted as the difference of the function value to the minimum value of the linear approximation to f_t at point X, where the minimum is taken over the feasible region.

3 Approximating Solution Paths

Our goal is to compute ε-approximations for Problem (1) for all parameter values $t \in [t_{\min}, t_{\max}] \subset \mathbb{R}$. Towards this goal we follow the simple idea of computing a slightly better approximation at some starting point, keeping this solution along the parameter path as the required approximation quality is guaranteed, and updating the solution only—again with slightly better approximation guarantee—when the required approximation quality can no longer be guaranteed. Hence, the complexity of our approach is determined by the number of solution updates. The latter complexity is always lower bounded by the following approximation path complexity.

Definition 3. *The* ε-approximation path complexity *of a parameterized optimization problem is the minimum number of intervals over all possible partitions of the parameter range $[t_{\min}, t_{\max}] \subset \mathbb{R}$, such that for each individual interval, there is a single solution that is an ε-approximation for that entire interval.*

3.1 Stability of Approximate Solutions

The following simple lemma is at the core of our discussion of approximation paths. It characterizes how far we can change the parameter t such that a given $\frac{\varepsilon}{\gamma}$-approximate solution X (for $\gamma > 1$) at t stays an ε-approximate solution.

Lemma 4. *Let $X \in \mathbb{S}^{n\times n}$ be an $\frac{\varepsilon}{\gamma}$-approximation to Problem (1) for some fixed parameter value t, and for some $\gamma > 1$. Then for all parameters $t' \in \mathbb{R}$ and feasible X' that satisfy*

$$\begin{array}{l} \lambda_{\max}(-\nabla f_{t'}(X')) \;-\; \lambda_{\max}(-\nabla f_t(X)) \\ \;+\; X' \bullet \nabla f_{t'}(X') \;-\; X \bullet \nabla f_t(X) \end{array} \leq \varepsilon\left(1 - \frac{1}{\gamma}\right),$$

it holds that X' is an ε-approximation to Problem (1) at the changed parameter value t'.

Proof. We have to show that

$$g_{t'}(X') = \lambda_{\max}\left(-\nabla f_{t'}(X')\right) + X' \bullet \nabla f_{t'}(X') \leq \varepsilon.$$

To do so, we add to the condition (inequality) in the statement of this lemma the inequality stating that X is an $\frac{\varepsilon}{\gamma}$-approximate solution at value t, i.e.,

$$\lambda_{\max}\left(-\nabla f_t(X)\right) + X \bullet \nabla f_t(X) \leq \frac{\varepsilon}{\gamma} \ ,$$

to obtain the claimed bound on the duality gap at the new parameter value t'. □

The following alternative (more restrictive) criterion is simpler to evaluate in a concrete implementation of our path algorithm, as for example for more complicated parameterizations of Problem (1).

Lemma 5. *Let $X \in \mathbb{S}^{n\times n}$ be an $\frac{\varepsilon}{\gamma}$-approximation for Problem (1) for some fixed parameter value t, and for some $\gamma > 1$. Then for all $t' \in \mathbb{R}$ that satisfy*

$$(1 + \|X\|_F)\, \|\nabla f_{t'}(X) - \nabla f_t(X)\|_F \leq \varepsilon\left(1 - \frac{1}{\gamma}\right),$$

it holds that X is still an ε-approximation at parameter value t'.

Proof. We aim to apply Lemma 4, and therefore try to upper bound the terms on the left hand side the inequality stated in Lemma 4 for $X' = X$. We start by upper bounding the difference in the $\lambda_{\max}(.)$-values: Weyl's perturbation theorem on the eigenvalues of a perturbed matrix $A' = A + E$ states that

$$|\lambda_{\max}(A') - \lambda_{\max}(A)| \leq \|E\|_2 \,,$$

see for example [Nakatsukasa, 2010]. Since the matrix spectral norm always satisfies $\|E\|_2 \le \|E\|_F$, applying Weyl's theorem to $A' = -\nabla f_{t'}(X)$ and $A = -\nabla f_t(X)$ gives

$$\begin{aligned} &|\lambda_{\max}\left(-\nabla f_{t'}(X)\right) - \lambda_{\max}\left(-\nabla f_t(X)\right)| \\ &\quad \le \quad \|\nabla f_{t'}(X) - \nabla f_t(X)\|_F\,. \end{aligned} \tag{5}$$

It remains to upper bound the term

$$X \bullet (\nabla f_{t'}(X) - \nabla f_t(X)),$$

which can be done by using the Cauchy-Schwarz inequality

$$\begin{aligned} &|X \bullet (\nabla f_{t'}(X) - \nabla f_t(X))| \\ &\quad \le \quad \|X\|_F \cdot \|\nabla f_{t'}(X) - \nabla f_t(X)\|_F\,. \end{aligned}$$

Hence, the inequality in the assumption of this lemma implies that the inequality in the statement of Lemma 4 holds, for $X' = X$, from which we obtain our claimed approximation guarantee $g_{t'}(X) \le \varepsilon$. □

Using this Lemma we can now prove the main theorem on the solution path complexity.

Theorem 6. *Let f_t be convex and continuously differentiable in X, and let $\nabla f_t(X)$ be Lipschitz continuous in t with Lipschitz constant L, for all feasible X. Then the ε-approximation path complexity of Problem (1) over the parameter range $[t_{\min}, t_{\max}] \subset \mathbb{R}$ is at most*

$$\left\lceil \frac{2L\cdot\gamma}{\gamma-1} \cdot \frac{t_{\max} - t_{\min}}{\varepsilon} \right\rceil = O\left(\frac{1}{\varepsilon}\right).$$

Proof. In order for the condition of Lemma 5 to be satisfied, we first use that for any $X \succeq 0, \operatorname{Tr}(X) \le 1$,

$$\begin{aligned} &(1 + \|X\|_F)\, \|\nabla f_{t'}(X) - \nabla f_t(X)\|_F \\ &\quad \le \quad (1 + \|X\|_F) \cdot L \cdot |t' - t| \\ &\quad \le \quad 2 \cdot L \cdot |t' - t|. \end{aligned}$$

Here L is the maximum of the Lipschitz constants with respect to t of the derivatives $\nabla f_t(X)$, taken over the compact feasible domain for X. So if we require the intervals to be of length

$$|t' - t| \le \frac{\varepsilon}{2L}\left(1 - \frac{1}{\gamma}\right),$$

we have that the condition in Lemma 5 is satisfied for any $X \succeq 0, \operatorname{Tr}(X) \le 1$.

The claimed bound on the path complexity follows directly by dividing the length $|t_{\max} - t_{\min}|$ of the parameter range by $\frac{\varepsilon}{2L}\left(1 - \frac{1}{\gamma}\right)$. □

Optimality. The path complexity result in Theorem 6 is indeed *worst-case optimal*, which can be seen as follows: when $\nabla f_t(X)$ does effectively change with t with a rate of L_X (here L_X is the Lipschitz constant with respect to t of $\nabla f_t(X)$), then the interval length where $g_t(X) \le \varepsilon$ holds can not be longer than $\Theta(\varepsilon)$.

3.2 Algorithms for Approximate Solution Paths

The straightforward analysis from above does not only give optimal bounds on the path complexity, but Lemma 4 also immediately suggests a simple algorithm to compute ε-approximate solution paths, which is depicted in Algorithm 1. Furthermore, the lemma implies that we can efficiently compute the exact largest possible interval length locally for each pair (X, t) in practice. In those regions where f_t changes only slowly in t, this fact makes the algorithm much more efficient than if we would just work with the guaranteed $O(\varepsilon)$ worst-case upper bound on the interval lengths, i.e., the step-size automatically adapts to the local complexity of f_t.

Algorithm 1 Approximate Solution Path

```
Input: Convex function $f_t, t_{\min}, t_{\max}, \varepsilon, \gamma$
Output: $\varepsilon$-approximate solution path for
        Problem (1)
Set $t := t_{\min}$.
repeat
  Compute an $\frac{\varepsilon}{\gamma}$-approximation $X$
        at parameter value $t$.
  Compute $t' > t$ such that
        $(1 + \|X\|_F)\, \|\nabla f_{t'}(X) - \nabla f_t(X)\|_F$
                        $\le \varepsilon\left(1 - \frac{1}{\gamma}\right)$.
  Update $t := t'$.
until $t \ge t_{\max}$
```

By Theorem 6, the running time of Algorithm 1 is in $O\big(T\big(\frac{\varepsilon}{\gamma}\big) / \varepsilon\big)$, where $T(\varepsilon')$ is the time to compute a single ε'-approximate solution for Problem (1) at a fixed parameter value t.

Plugging-in Existing Optimizers. For completeness, we discuss some of the many solvers that can be plugged into our path Algorithm 1 to compute a single approximate solution. In our experiments we used Hazan's algorithm [Hazan, 2008, Jaggi and Sulovský, 2010] because it scales well to large inputs, provides approximate solutions with guarantees, and only requires a single approximate eigenvector computation in each of its iterations. The algorithm returns a guaranteed ε-approximation to problem (1) after at most $O\big(\frac{1}{\varepsilon}\big)$ many iterations, and provides a low-rank matrix factorization of the resulting estimate X for free, i.e., a solution of rank $O\big(\frac{1}{\varepsilon}\big)$. Other algorithms often employ low-rank heuristics for practical reasons. Since low-rank constraints do form a non-convex domain, these methods lose the merits and possible guarantees for convex optimization methods. With Hazan's algorithm we can approximate the original convex Problem (3) with guaranteed approximation quality, without any restrictions on the rank.

There are many other popular methods to solve nuclear norm regularized problems. Alternating gradient descent or stochastic gradient descent (SGD) methods have been used extensively in particular for matrix completion problems, see for example [Rennie and Srebro, 2005, Webb, 2006, Lin, 2007, Koren et al., 2009, Takács et al., 2009, Recht and Ré, 2011]. However, these methods optimize a non-convex formulation of (2) and can get stuck in local minima, and therefore—in contrast to Hazan's method with our convex transformation (4)—come in general with no convergence guarantee. On the other hand, there are also several known methods of "proximal gradient" and "singular value thresholding"-type from the optimization community, see for example [Toh and Yun, 2010], which however require a full singular value decomposition in each iteration, in contrast to the simpler steps of Hazan's algorithm [Jaggi and Sulovský, 2010]. Nevertheless, any of these other methods and heuristics can still be used as the internal optimizer in our path-tracking algorithm, as we can always compute the duality gap as a certificate for the quality of the found approximate solution.

4 Applications

Using our solution path approximation algorithm, we directly obtain piecewise constant solution paths of guaranteed approximation quality for any problem of the form (1), including all nuclear norm regularized Problems (2) and (3), such as standard matrix completion problems, and robust PCA.

4.1 Matrix Completion

Algorithm 1 applies to matrix completion problems with any convex differentiable loss function, such as the smoothed hinge loss or the standard squared loss, and includes the classical maximum-margin matrix factorization variants [Srebro et al., 2004].

The regularized matrix completion task is exactly Problem (3) with the function f given by the loss over the observed entries of the matrix, $\Omega \subseteq [n] \times [m]$, i.e. $f(Z) = \sum_{(i,j)\in\Omega} L(Z_{ij}, Y_{ij})$, where $L(.,.)$ is an arbitrary loss-function that is convex in its Z-argument. The most widely used variant employs the squared loss, given by

$$f(Z) = \frac{1}{2} \sum_{(i,j)\in\Omega} (Z_{ij} - Y_{ij})^2. \tag{6}$$

Using the notation $(A)_\Omega$ for the matrix that coincides with A on the indices Ω and is zero otherwise, $\nabla f(Z)$ can be written as

$$\nabla f(Z) = (Z - Y)_\Omega.$$

To apply Algorithm 1 we move from the problem in formulation (3) to the formulation (4). Still, the symmetric gradient matrix $\nabla f_t(X) \in \mathbb{S}^{(m+n)\times(m+n)}$ used by the algorithm is of the simple form of $\nabla f(Z)$ as above (recall the notation $X = \left(\begin{smallmatrix} V & Z \\ Z^T & W \end{smallmatrix}\right)$). As this matrix is sparse—it has only $|\Omega|$ non-zero entries—storage and approximate eigenvector computations can be performed much more efficiently than for dense problems. An equivalent *matrix factorization* of any approximation X for Problem (1) can always be obtained directly from the Cholesky decomposition of X, because X is positive semidefinite.

4.2 Weighted Nuclear Norm

A promising weighted nuclear norm regularization approach for matrix completion has been recently proposed in [Salakhutdinov and Srebro, 2010]. For fixed weight vectors $p \in \mathbb{R}^m, q \in \mathbb{R}^n$, the weighted nuclear norm $\|Z\|_{nuc(p,q)}$ of $Z \in \mathbb{R}^{m\times n}$ is defined as

$$\|Z\|_{nuc(p,q)} := \|PZQ\|_* \ ,$$

where $P = \operatorname{diag}(\sqrt{p}) \in \mathbb{R}^{m\times m}$ denotes the diagonal matrix whose i-th diagonal entry is $\sqrt{p_i}$, and analogously for $Q = \operatorname{diag}(\sqrt{q})$. Here $p \in \mathbb{R}^m$ is the vector whose entries are the probabilities $p(i) > 0$ that the i-th row is observed in the sampling Ω. Analogously, $q \in \mathbb{R}^n$ contains the probability $q(j) > 0$ for each column j. The opposite weighting has also been suggested in [Weimer et al., 2008].

Any optimization problem with a weighted nuclear norm regularization

$$\begin{array}{ll} \min\limits_{Z\in\mathbb{R}^{m\times n}} & f(Z) \\ s.t. & \|Z\|_{nuc(p,q)} \le \frac{t}{2} \ , \end{array} \tag{7}$$

and arbitrary loss function f can therefore be phrased equivalently over the domain $\|PZQ\|_* \le t/2$, such that it reads as (if we substitute $\bar{Z} := PZQ$),

$$\begin{array}{ll} \min\limits_{Z\in\mathbb{R}^{m\times n}} & f(P^{-1}\bar{Z}Q^{-1}) \\ s.t. & \|\bar{Z}\|_* \le \frac{t}{2} \ , \end{array}$$

Hence, analogously to the original nuclear norm regularization approach, we have reduced the task to our standard convex Problem (1) for f_t that here is defined as

$$f_t(X) = \hat{f}\left(t\begin{pmatrix} V & \bar{Z} \\ \bar{Z}^T & W \end{pmatrix}\right) := f(tP^{-1}\bar{Z}Q^{-1}).$$

This implies that any algorithm solving Problem (1) also serves as an algorithm for weighted nuclear norm regularized problems. In particular, Hazan's algorithm [Hazan, 2008, Jaggi and Sulovský, 2010] implies a guaranteed approximation quality of ε after $O\big(\frac{1}{\varepsilon}\big)$ many rank-1 updates. So far, to the best of our knowledge, no approximation guarantees were known for such optimization problems involving the weighted nuclear norm.

4.3 Solution Paths for Robust PCA

Principal component analysis (PCA) is still widely used for the analysis of high-dimensional data and dimensionality reduction although it is quite sensitive to errors and noise in just a single data point. As a remedy to this problem [Candes et al., 2011] have proposed a robust version of PCA, also called principal component pursuit, which is given as the following optimization problem,

$$\min_{Z\in\mathbb{R}^{m\times n}} \|Z\|_* + \lambda' \|M - Z\|_1 \ . \tag{8}$$

Here $\|.\|_1$ is the entry-wise ℓ_1-norm, and M is the given data matrix. This problem is already in the form of Problem (2), for $\lambda = \frac{1}{\lambda'}$. Therefore, we can approximate its entire solution path in the regularization parameter λ using Algorithm 1, obtain piecewise constant solutions together with a continuous ε-approximation guarantee along the entire regularization path.

4.4 Solution Paths for Sparse PCA

The idea of sparse PCA is to approximate a given data matrix $A \in \mathbb{S}^{n\times n}$ by approximate eigenvectors that are sparse, see Zhang et al. [2010] for an overview. Many algorithms have been proposed for sparse PCA, see for example [Sigg and Buhmann, 2008] and [d'Aspremont et al., 2007a], the latter algorithm is also considering a discrete solution path, as the sparsity changes.

The SDP-relaxation of [d'Aspremont et al., 2007b, Equation 3.2] for sparse PCA of a matrix A is given by

$$\begin{array}{rl} \min\limits_{X\in\mathbb{S}^{n\times n}} & \rho\cdot \mathbf{e}^T|X|\mathbf{e} - \mathrm{Tr}(AX) \\ s.t. & \mathrm{Tr}(X) = 1 \ , \\ & X \succeq 0 \ . \end{array} \tag{9}$$

Here $|X|$ is element-wise for the matrix X, and $\mathbf{e}\in\mathbb{R}^n$ is the all-ones vector. Using Algorithm 1, we can directly compute a complete approximate solution path in the penalty parameter ρ of this relaxation, which already is of the form (1).

5 Experimental Results

The goal of our experiments is to demonstrate that the entire regularization path for nuclear norm regularized problems is indeed efficiently computable with our approach for large datasets, and that our described approximation guarantees are practical. Hence, we have applied a variant of Algorithm 1 for a special case of Problem (1) to nuclear norm regularized matrix completion tasks on the standard MovieLens data sets[2] using the squared loss function, see Equation (6).

Table 2: Summary of the MovieLens data sets.

	#ratings	$m = \#users$	$n = \#movies$
MovieLens 100k	10^5	943	1682
MovieLens 1M	10^6	6040	3706
MovieLens 10M	10^7	69878	10677

As the internal optimizer within the path solver, we have used Hazan's algorithm, as detailed in [Jaggi and Sulovský, 2010], using the power method for computing approximate eigenvectors. To get an accurate bound on $\lambda_{\max}$ when computing the duality gap $g_t(X)$ for each candidate solution X, we performed 300 iterations of the power method. In all our experiments we have used a quality improvement factor of $\gamma = 2$.

Each dataset was uniformly split into 50% test ratings, and 50% training ratings. All the provided ratings were used "as-is", without normalization to any kind of prior[3]. The accuracy ε was chosen as the *relative* error, with respect to the initial function value $f_t(X = 0)$ at $t = 0$, i.e., $\varepsilon = \varepsilon' f_0(0)$. As f is the squared loss, $f_0(0)$ equals the Frobenius norm of the observed training ratings, see Equation (6).

All the results that we present below have been obtained from our (single-thread) implementation in Java 6 on a 2.4 GHz Intel Core i5 laptop with 4 GB RAM running MacOS X.

Results. We have computed the entire regularization paths for the nuclear norm $\|.\|_*$ regularized Problem (3), as well as for regularization by using the

[2] See http://www.grouplens.org and Table 2 for a summary.

[3] Users or movies which do appear in the test set but not in the training set were kept as is, as our model can cope with this. These ratings are accounted in the test RMSE, which is slightly worse therefore (our prediction X will always remain at the worst value, zero, at any such rating).

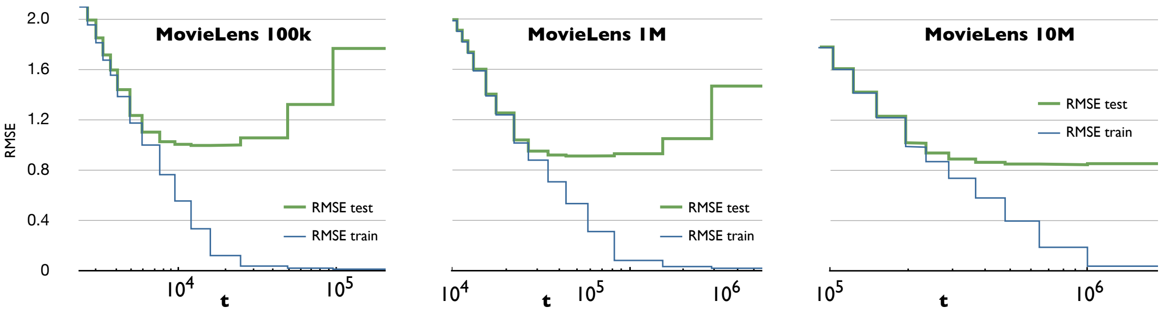

Figure 1: The nuclear norm regularization path for the three MovieLens datasets.

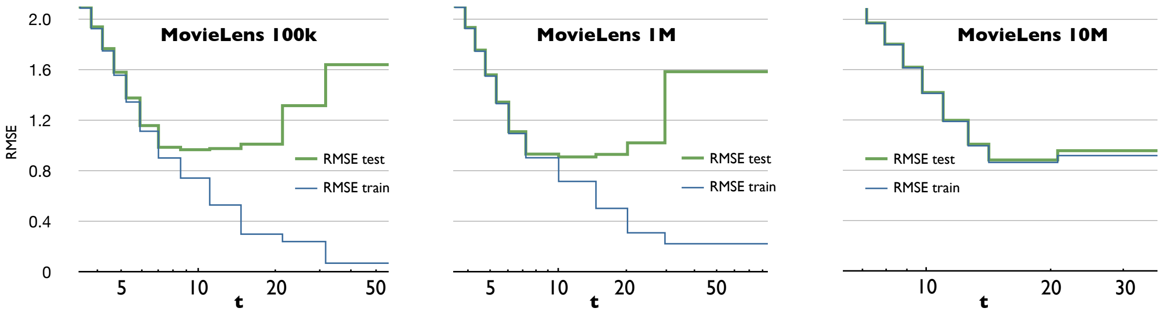

Figure 2: The regularization path for the weighted nuclear norm $\|.\|_{nuc(p,q)}$.

Table 1: Dependency of the path complexity ($\#int$) on the accuracy ε.

Regularization	Accuracy	MovieLens 100k, $\gamma = 2$					
	$\varepsilon/f_0(0)$	$t_{\min}$	$t_{\max}$	$\#int$	Δ_t^{avg}	$f_{t_{\max}}^{\text{train}}$	$f_{\text{opt}}^{\text{test}}$
Nuclear norm	0.05	1000	60000	23	9438	0.0070	0.9912
$\|.\|_*$	0.01	1000	60000	97	582	0.0054	0.9905
	0.002	1000	60000	387	175	0.0009	0.9981
Weighted	0.05	2	50	18	3.21	0.0619	0.9607
nuclear norm	0.01	2	50	73	1.18	0.0147	0.9559
$\|.\|_{nuc(p,q)}$	0.002	2	50	325	0.140	0.0098	0.9581

weighted nuclear norm $\|.\|_{nuc(p,q)}$ as in the formulation as Problem (7).

Figures 1 and 2 show the rooted mean squared error (RMSE) values along a guaranteed ($\varepsilon' = 0.05$)-approximate solution path for the three MovieLens datasets.

Table 1 shows that the dependency of the path complexity on the approximation quality is indeed favorably weak. Here $\#int$ denotes the number of intervals of constant solution with guaranteed ε-small duality gap; Δ_t^{avg} is the average length of an interval with constant solution; $f_{t_{\max}}^{\text{train}}$ is the RMSE on the training data at the largest parameter value $t_{\max}$; and finally $f_{\text{opt}}^{\text{test}}$ is the best $RMSE_{\text{test}}$ value obtained over the entire regularization path.

6 Conclusions

We have presented a simple but efficient algorithm that allows to track approximate solutions of parameterized semidefinite programs with guarantees along the entire parameter path. Many well known semidefinite optimization problems such as regularized matrix factorization/completion and nuclear norm regularized problems can be approximated efficiently by this algorithm. Our experiments show a surprisingly small path complexity when measured in the number of intervals of guaranteed ε-accurate constant solutions for the considered problems, even for large matrices. Thus, the experiments confirm our theoretical result that the complexity is independent of the input size.

In the future we plan to explore more applications of parameterized semidefinite optimization in machine learning, where our algorithm may also provide more insight into the dependence of these optimization solutions on the regularization. In particular, it will be interesting to investigate kernel learning, metric learning and other relaxations of sparse PCA in more detail.

Acknowledgements Joachim Giesen and Sören Laue have been supported by DFG grant GI-711/3-1. Martin Jaggi has been supported by a Google research award and by SNF grant 20PA21-121957.

References

S. Boyd and L. Vandenberghe. *Convex optimization.* 2004.

E. J. Candes and B. Recht. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

E. J. Candès and T. Tao. The Power of Convex Relaxation: Near-Optimal Matrix Completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011.

A. d'Aspremont, F. Bach, and L. Ghaoui. Full regularization path for sparse principal component analysis. *ICML*, 2007a.

A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A Direct Formulation for Sparse PCA Using Semidefinite Programming. *SIAM Review*, 49(3):434–448, 2007b.

M. Fazel, H. Hindi, and S. P. Boyd. A Rank Minimization Heuristic with Application to Minimum Order System Approximation. *Proceedings American Control Conference*, 6:4734–4739, 2001.

B. Gärtner, M. Jaggi, and C. Maria. An Exponential Lower Bound on the Complexity of Regularization Paths. *arXiv.org*, cs.LG, 2010.

J. Giesen, M. Jaggi, and S. Laue. Approximating Parameterized Convex Optimization Problems. In *ESA - European Symposium on Algorithms, LNCS*, pages 524–535. 2010.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The Entire Regularization Path for the Support Vector Machine. *JMLR*, 5:1391–1415, 2004.

E. Hazan. Sparse Approximate Solutions to Semidefinite Programs. In *LATIN, LNCS*, pages 306–316. Springer Berlin Heidelberg, 2008.

M. Jaggi. *Sparse Convex Optimization Methods for Machine Learning.* PhD thesis, ETH Zürich, 2011.

M. Jaggi and M. Sulovský. A Simple Algorithm for Nuclear Norm Regularized Problems. *ICML*, 2010.

Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*, 42(8):30–37, 2009.

C.-J. Lin. Projected Gradient Methods for Nonnegative Matrix Factorization. *Neural Comput.*, 19(10): 2756–2779, 2007.

G. Loosli, G. Gasso, and S. Canu. Regularization Paths for nu-SVM and nu-SVR. *ISNN, International Symposium on Neural Networks, LNCS*, 4493: 486, 2007.

R. Mazumder, T. Hastie, and R. Tibshirani. Spectral Regularization Algorithms for Learning Large Incomplete Matrices. *JMLR*, 11:1–36, 2010.

Y. Nakatsukasa. Absolute and relative Weyl theorems for generalized eigenvalue problems. *Linear Algebra and Its Applications*, 432(1):242–248, 2010.

B. Recht and C. Ré. Parallel Stochastic Gradient Algorithms for Large-Scale Matrix Completion. *submitted*, 2011.

J. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. *ICML*, 2005.

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

R. Salakhutdinov and N. Srebro. Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm. *NIPS*, 23, 2010.

C. D. Sigg and J. M. Buhmann. Expectation-Maximization for Sparse and Non-Negative PCA. *ICML*, pages 960–967, 2008.

N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. *NIPS*, 17: 1329–1336, 2004.

G. Takács, I. Pilászy, B. Németh, and D. Tikk. Scalable Collaborative Filtering Approaches for Large Recommender Systems. *JMLR*, 10, 2009.

K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 2010.

B. Webb. Netflix Update: Try This at Home, 2006. `http://sifter.org/~simon/journal/20061211.html`.

M. Weimer, A. Karatzoglou, and A. Smola. Improving maximum margin matrix factorization. *Machine Learning*, 72(3):263–276, 2008.

Y. Zhang, A. d'Aspremont, and L. E. Ghaoui. Sparse PCA: Convex Relaxations, Algorithms and Applications. *arXiv*, math.OC, 2010.