Epigraph projections for fast general convex programming

Po-Wei Wang POWEIW@CS.CMU.EDU
Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Matt Wytock MWYTOCK@STANFORD.EDU
Electrical Engineering Department, Stanford University, Stanford, CA 94305 USA

J. Zico Kolter ZKOLTER@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Abstract

This paper develops an approach for efficiently solving general convex optimization problems specified as disciplined convex programs (DCP), a common general-purpose modeling framework. Specifically we develop an algorithm based upon fast *epigraph projections*, projections onto the epigraph of a convex function, an approach closely linked to proximal operator methods. We show that by using these operators, we can solve any disciplined convex program without transforming the problem to a standard cone form, as is done by current DCP libraries. We then develop a large library of efficient epigraph projection operators, mirroring and extending work on fast proximal algorithms, for many common convex functions. Finally, we evaluate the performance of the algorithm, and show it often achieves order of magnitude speedups over existing general-purpose optimization solvers.

1. Introduction

Although convex optimization techniques underly a large number of machine learning algorithms, there has been a traditional tension between general purpose optimization methods and specialized algorithms. General purpose algorithms, exemplified by standard cone form solvers like linear, second order, and semidefinite cone solvers, with the addition of modeling languages such as cvx (Grant & Boyd, 2008) or cvxpy (Diamond & Boyd, 2015) (which convert problems to these forms), provide a very flexible "rapid prototyping" framework for convex optimization. However, these methods typically do not scale beyond medium-size problems, and are not well-suited to the larger problems that make up much current machine learning work. This has lead to the development of a number of specialized solvers for many problems of interest including support vector machines (Platt, 1999; Shalev-Shwartz et al., 2011; Chang & Lin, 2011), ranking approaches (Joachims, 2002), sparse inverse covariance estimation (Friedman et al., 2008; Hsieh et al., 2014), etc, just to name a very small number of examples. Recently, many special purpose algorithms have started to use *proximal operators* as a key building block.

This current work looks to begin bridging the gap between specialized and general purposes solvers. The first main contribution of this work is a method for converting *any* disciplined convex program (DCP) into a form that can be directly solved by proximal methods, without converting to cone form. Doing so requires a new set of operators for *epigraph projections*, an operator related to but distinct from most existing proximal operators; specifically, for some convex function $f : \mathbb{R}^n \to \mathbb{R}$, epigraph projections are optimization problems of the form

$$\mathrm{epi}_f(v, s) = \operatorname*{argmin}_{x,t} \{\|x-v\|_2^2+(t-s)^2 \mid f(x) \le t\} \quad (1)$$

i.e., they are a projection of some $(v \in \mathbb{R}^n, s \in \mathbb{R})$ onto the *epigraph* set $\{(x, t) \mid f(x) \le t\}$. Thus, the second main contribution of this work is the development of a wide set of epigraph projection algorithms (plus some additional proximal operators), which in turn can solve a very broad range of DCPs without ever resorting to standard cone transformations. To build these fast epigraph projections operators we develop new optimization approaches, including an implicit dual Newton method and a $O(n)$ "sum-of-clip" solver, both of which we detail in later sections. Finally, we implement these methods in a generic optimization solver, and show significant speedup over existing general-purpose approaches, often an order of magnitude or more.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

2. Background

Disciplined convex programming Disciplined convex programing (DCP) (Grant et al., 2006) is a modeling framework for convex programs that allows very general convex problems to be specified using a relatively small set of convex atomic function and set of compositional rules that preserve convexity. For instance, a basic rule states that for $h : \mathbb{R}^k \to \mathbb{R}$, $g_i : \mathbb{R}^n \to \mathbb{R}$, and one of the following holds for each $i = 1, \ldots, k$:

- g_i convex, h convex nondecreasing in argument i
- g_i concave, h concave nonincreasing in argument i
- g_i affine, h convex,

then $f = h(g_1(x), \ldots, g_k(x))$, is convex (see e.g. Boyd & Vandenberghe (2004, §3.2.4) for details). The DCP ruleset is sufficient but *not necessary* for a problem to be convex: the "log-sum-exp" function is convex, but cannot be verified as such by the above rules. However, log-sum-exp *can* be represented as a separate atomic function that is convex (and monotonic in its arguments). In practice, most convex problems can be written using the DCP ruleset with a relatively small set of atoms.

In addition to this verification, DCP libraries also provide a means of converting the problems to a standard form cone problem. Each functional atom also provides an *graph implementation* of the function, a representation of the function as the solution to a linear cone problem. For example, the ℓ_1 norm has the graph form

$$\|x\|_1 = \min_y 1^T y, \ \text{ subject to} -y \le x \le y. \tag{2}$$

Once an optimization problem has been verified as convex by the DCP rules, we replace all instances of DCP atoms with their corresponding epigraph implementation (introducing new variables as needed). The resulting problem, which must be itself a linear cone problem, is guaranteed to be equivalent to the original optimization problem, and can then be solved by standard form solvers.

Proximal methods A recent trend in machine learning optimization methods has been the increased development of algorithms based upon proximal operators. Given a convex function $f : \mathbb{R}^n \to \mathbb{R}$, the proximal operator is defined as

$$\text{prox}_{\lambda f}(v) = \operatorname*{argmin}_x \lambda f(x) + \frac{1}{2}\|x - v\|_2^2. \tag{3}$$

Crucially, for many functions f, including many nonsmooth functions, we can compute the proximal operator in closed form (or if not, at least compute it to numerical precision very efficiently). For instance, the proximal operator for the ℓ_1 norm is given by soft thresholding (Donoho, 1995), $\text{prox}_{\lambda\|\cdot\|_1}(v) = \max\{|v| - \lambda, 0\} \cdot \text{sign}(v)$, where all operators are applied elementwise.

Generally speaking, "proximal algorithms" refer to any optimization method that uses a proximal operator in its iteration. Such algorithms are not new, with the original proximal point algorithm proposed in 1976 (Rockafellar, 1976), but they have seen increased usage in recent years, often in conjunction with the increasing use of ℓ_1 regularization; see e.g. (Parikh & Boyd, 2013) for recent survey of several such methods.

Operator splitting techniques are one class of proximal algorithm, which solve a composite optimization problem

$$\operatorname*{minimize}_x \quad f(x) + g(x), \tag{4}$$

typically by exploiting fast proximal operators for f and g. A general review of operator splitting algorithms is given in (Ryu & Boyd, 2016), and two algorithms of particular recent interest are Douglas-Rachford splitting (Douglas & Rachford, 1956) and the alternating direction method of multipliers (ADMM) (Gabay & Mercier, 1976; Boyd et al., 2011). Ultimately, our algorithm uses ADMM to solve the resulting problems after transforming them to a suitable form, though other operator splitting methods can be applied as well. Of particular relevance to our problem is the splitting cone solver (SCS) (O'Donoghue et al., 2013), an application of ADMM to linear cone programs; together with existing DCP libraries, this system is very comparable to our own (a solver capable of solving DCP problems using proximal methods), and comparisons to SCS will be one of the primary focuses in later sections. Also related is the TFOCS algorithm (Becker et al., 2011), which uses first order methods to solve several classes of cone problems, though it does not solve general DCPs.

In the general theme of proximal algorithms, there have been a few papers that do consider epigraph projections in a limited manner. For example, Tofighi et al. (2014) use a epigraph projection onto a total-variation-type norm described via a linear program for image deconvolution; Chierchia et al. (2012) describe projections on to the epigraph of the ℓ_2 and ℓ_∞ norms (in the later case using a naive $O(n \log n)$ algorithm), for a few specialized optimization problems; and Harizanov et al. (2013) use an epigraph projection onto a very specific function useful for image processing. None of these approaches relate to solving general optimization problems, nor do they the develop the range of epigraph projections that we cover.

3. Converting DCPs to proximal form with epigraph projections

In this section, we show that any DCP can be solved using a proximal algorithm that employs standard proximal operators plus epigraph projections. In particular we show that any DCP composed entirely of atoms from some set of functions $\mathcal{G}$ (but which can be composed according to the DCP ruleset to form much more complex functions that do not admit efficient proximal operators or epigraph projections) can be solved using *only proximal operators and epigraph projections for functions* $g \in \mathcal{G}$; thus, if we implement proximal and epigraph projection operators for every element in a DCP atom library, we can directly solve the problem without ever using any of the DCP graph implementations, but just the relevant operators directly. This is formalized as follows

Theorem 1. *Consider the DCP optimization problem*

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & f_i(x) \le 0, \ i = 1, \ldots, p \\ & Ax = b \end{array} \tag{5}$$

with optimization variable x, affine constraints $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, and convex functions $f_i : \mathbb{R}^n \to \mathbb{R}$ built via the DCP ruleset from convex atoms in a set $\mathcal{G}$. Then this optimization problem can be written equivalently as

$$\begin{array}{ll} \underset{\bar{x}}{\text{minimize}} & \sum_{i=1}^{N} \bar{f}_i(\bar{x}) \\ \text{subject to} & \bar{A}\bar{x} = \bar{b} \end{array} \tag{6}$$

for some expanded set of variables $\bar{x} \in \mathbb{R}^{\bar{n}}$ such that $x = \bar{x}_{\mathcal{I}_0}$ for some set $\mathcal{I}_0$; affine constraints $\bar{A} \in \mathbb{R}^{\bar{m} \times \bar{n}}$ and $\bar{b} \in \mathbb{R}^{\bar{m}}$; and convex functions $\bar{f} : \mathbb{R}^{\bar{n}} \to \mathbb{R}$ of the form

$$\bar{f}_i(\bar{x}) = \left\{ \begin{array}{l} g(\bar{x}_{\mathcal{I}_i}) \\ I\{g(\bar{x}_{\mathcal{I}_i}) \le \bar{x}_{t_i}\} \end{array} \right. \quad \textit{or} \tag{7}$$

for some atomic function $g \in \mathcal{G}$ where $\mathcal{I}_i$ selects some subject of the indices and $t_i \notin \mathcal{I}$ selects the epigraph variable.

Again, the key point here is that although atom $g \in \mathcal{G}$ admits efficient operators, the functions f_i made up of compositions of these functions can be much more complex (one simple example for a robust SVM is below). But the theorem states that the problem can be transformed into an equivalent one that *only* requires proximal operators and epigraph projections for functions in the atom set $\mathcal{G}$.

The proof of Therem 1 is straightforward, but requires a slightly more formal definition of the representation of DCP functions; we describe this only briefly here, with more detail about DCPs in general available in Grant et al. (2006). Each DCP function is represented as a expression tree, where each non-leaf node in the tree must be a DCP atom (which itself can be either convex, concave, or affine), and each leaf node must be a variable or a constant. For example, an ℓ_∞ robust SVM (see Appendix A.4 for details about the derivation of this objective term) can be written as the optimization problem

$$\underset{\theta}{\text{minimize}} \ \ \frac{\lambda}{2}\|\theta\|_2^2 + \sum_{i=1}^{m} \max\{0, 1 - y_i \cdot \theta^T x_i + \|P^T\theta\|_1\}. \tag{8}$$

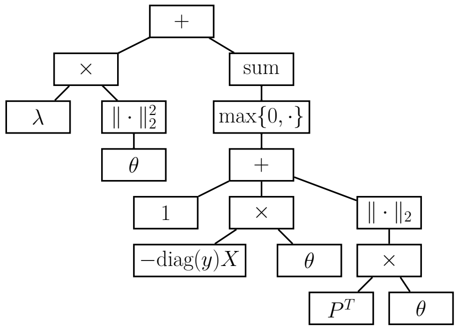

Figure 1. A DCP expression tree representing an ℓ_∞ robust SVM optimization objective.

Figure 1 shows the DCP expression tree for this objective: $\|\cdot\|_2^2$, $\|\cdot\|_1$, and $\max\{0,\cdot\}$ atoms are convex, the sum, $+$ and $\times$ atoms are affine (the $\times$ atom under DCP must have a constant on one side), the leaf node θ is a variable, and the λ, 1, $-\text{diag}(y)X$ and P^T leaf nodes are constants.

Proof of Theorem 1. The proof proceeds by considering each DCP function f_i as an expression tree of depth n, and employs a type of "bottom up" reduction to reduce the tree to one of level $n-1$, plus some additional equality and epigraph constraints. We select some leaf node at level n, which we denote l_1 and consider its immediate parent g and siblings, $l_2, \ldots, l_n$. We introduce a new variable $\hat{l}$ and, based upon whether g is affine, convex, or concave, we add the constraint $g(l_1, \ldots, l_n)\ \{=, \le, \ge\}\ \hat{l}$: we add an equality constraint for g affine and place these constraints into the $\bar{A}\bar{x} = \bar{b}$ matrix; and we add $\le$ constraints for g convex and $\ge$ constraints for g concave, and introduce the appropriate epigraph indicator functions $\bar{f}_i = I\{g(l_1, \ldots, l_n)\ \{\ge, \le\}\ \hat{l}\}$ in these two cases. Finally, we replace the entire subtree with the $\hat{l}$ variable. We repeat this process for all leaves at level n, resulting in an equivalent expression tree of depth $n-1$ (plus additional equality constraints and epigraph indicators).

When we reach the root node of the tree for the objective function f_0, we simply add the expression itself as a function $\bar{f}_i = g$ (or by way of optimization, we immediately terminate if the root node is an addition operator and the nodes at depth 2 already have efficient proximal operators or epigraph projections). For the root nodes of expression trees for f_i, $i \ge 1$, we add the epigraph constraint $\bar{f}_i = I(g(l_1, \ldots, l_n) \le \hat{l})$ plus the constraint $\hat{l} = 0$. ☐

Example The theorem is best illustrated by example, so we consider here how we can use this approach to transform the robust SVM problem (8). This objective composes the hinge loss, a linear function of the parameters θ, and an ℓ_1 norm inside the hinge loss, and there is no proximal operator that can be applied directly to the entire top-level objective terms. However, the objective still admits a DCP formulation, as shown in Figure 1, and applying the operations from Theorem 1 (using atoms for the squared ℓ_2 norm, hinge function $\max\{0, \cdot\}$, and ℓ_1 norm) results in the new variables and constraints

$$\begin{aligned} \hat{l}_1 &\Rightarrow P^T x = \hat{l}_1 \\ \hat{l}_2 &\Rightarrow \|\hat{l}_1\|_1 \le \hat{l}_2 \\ \hat{l}_3 &\Rightarrow 1 - \mathrm{diag}(y) X\theta + 1\hat{l}_2 = \hat{l}_3 \end{aligned} \tag{9}$$

and the optimization problem over new variable $\bar{x} = (x, \hat{l}_1, \hat{l}_2, \hat{l}_3)$

$$\begin{aligned} \underset{\bar{x}}{\text{minimize}} \;\; & \frac{\lambda}{2}\|\theta\|_2^2 + \sum \max\{\hat{l}_3, 0\} + I\{\|\hat{l}_1\|_1 \le \hat{l}_2\} \\ \text{subject to} \;\; & P^T x = \hat{l}_1 \\ & 1 - \mathrm{diag}(y) X\theta + 1\hat{l}_2 = \hat{l}_3. \end{aligned} \tag{10}$$

This is an optimization problem in the form (6): each term in the objective is either one of the atomic functions, or an indicator of the epigraph of an atomic function, plus additional linear constraints over these variables.

Optimization approaches Finally, as an immediate corollary of the above theorem comes the fact that we can solve the modified DCP problem using just a sequence of proximal and epigraph projection operators of just the atomic functions. The proof here is algorithmic, and comes by simply applying an operator splitting algorithm like ADMM directly to the modified problem (6). Indeed, the solver we use in Section 5 uses an ADMM solver, though many operator splitting approaches can solve problems of the form (6) equally well. We omit the full derivation for brevity (see e.g. Boyd et al. (2011) for a full derivation of ADMM in this so-called consensus setting), but briefly, to solve (6) we can introduce N copies of the $\bar{x}$ variable, plus a consensus variable z, and solve the optimization problem

$$\begin{aligned} \underset{\bar{x}_1,\ldots,\bar{x}_N, z}{\text{minimize},} \;\; & \sum_{i=1}^N \bar{f}_i(\bar{x}_i) + I\{\bar{A}z = \bar{b}\} \\ \text{subject to} \;\; & \bar{x}_i = z, \; i = 1, \ldots, N \end{aligned} \tag{11}$$

which results in the update rules (assuming for simplicity an augmented Lagrangian parameter $\rho = 1$)

$$\begin{aligned} \bar{x}_i^{k+1} &\leftarrow \mathrm{prox}_{\bar{f}_i}(u_i^k - z^k) \\ z^{k+1} &\leftarrow \begin{bmatrix} I & \bar{A}^T \\ \bar{A} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{N}\sum_{i=1}^N (\bar{x}_i^{k+1} + u^k) \\ \bar{b} \end{bmatrix} \\ u_i^{k+1} &\leftarrow u_i^k + \bar{x}_i^{k+1} - z^{k+1}. \end{aligned} \tag{12}$$

Since the proximal operator for the indicator of a set is simply the projection onto that set, the proximal operator for $\bar{f}_i$ terms that are indicators of an epigraph set is exactly

$$\mathrm{prox}_{\bar{f}_i}(v) = \mathrm{epi}_g(v_{\mathcal{I}_i}, v_{t_i}). \tag{13}$$

Thus, the solution above requires computing only proximal and epigraph projection operators for the underlying atoms in the DCP, as stated originally. In practice, we make a few further refinements of this algorithm (the splitting consensus variable z can be created jointly with the constraints $A\bar{x} = b$, and some prox operators can directly incorporate affine terms), but the above method is suffices to show the key point: that we can solve *any* DCP problem built from atoms for which we can implement efficient proximal operators and epigraph projections.

4. Fast epigraph projection solvers

While proximal operators for many functions have been studied extensively in the machine learning and optimization literature, there has been very little treatment of epigraph projections. Therefore, because we need these operators to solve general DCP problems using the method described previously, in this section we develop a number of methods for efficiently computing the epigraph projection for a wide set of convex atoms.

Consider the optimization problem for epigraph projection

$$\begin{aligned} \underset{x,t}{\text{minimize}} \;\; & \frac{1}{2}\|(x,t) - (v,s)\|_2^2 \\ \text{subject to} \;\; & f(x) \le t. \end{aligned} \tag{14}$$

The Lagrangian of this problem is given by

$$\mathcal{L}(x,t,\lambda) = \frac{1}{2}\|(x,t) - (v,s)\|_2^2 + \lambda(f(x) - t). \tag{15}$$

Minimizing over x and t gives the solutions

$$x^\star = \mathrm{prox}_{\lambda f}(v), \;\; t^\star = s + \lambda, \tag{16}$$

which correspond to the dual problem

$$\underset{\lambda \ge 0}{\text{maximize}} \;\; P_{\lambda f}(v) - \frac{1}{2}\lambda^2 - \lambda s, \tag{17}$$

where

$$P_{\lambda f}(v) = \min_x \lambda f(x) + \frac{1}{2}\|x - v\|_2^2 \tag{18}$$

denotes the objective obtained by the proximal operator. The dual problem is concave with a single variable λ, so it can always be solved via bisection. However, the bisection algorithm takes $O(\log 1/\epsilon)$ to iterations to reach accuracy ϵ, which can itself become a bottleneck. A main contribution of this paper, therefore, is the development of more efficient epigraph projection algorithms in many settings, often with practical (and theoretical) efficiency that is nearly that of the equivalent proximal operator itself. In developing these methods, we also develop some number of new proximal algorithms as well, for which we know of no existing methods in the literature. A summary of the operator we develop is shown in the Appendix, Table 2.

In the remainder of this section, we detail several of the approaches that we use to develop fast epigraph projection operators (and sometimes also proximal operators) for the convex atoms listed in this table. Although many of the epigraph projection operators require specialized algorithms, they broadly fall into five categories: those solved by an exact analytical solution, by a primal-dual Newton method, by a new optimization approach we refer to as the implicit dual Newton method, by a new approach we call the sum-of-max algorithm for finding zeros of piecewise linear functions, and orthogonal matrix methods. As a final note, for all functions that apply elementwise to vectors, (i.e., the absolute value $f(x) = |x|$), we will be constructing epigraph projections for the *sum* over all the entries

$$\mathrm{epi}_f(v, s) = \underset{x,t:\sum_i f(x_i)\leq t}{\mathrm{argmin}} \frac{1}{2}\|(x,t) - (v,s)\|^2 \tag{19}$$

as this strictly generalizes the case of constructing an epigraph for each element individually.

4.1. Exact methods

While many proximal operators have a closed form solution, a relatively smaller number of epigraph projections have them. A few notable exceptions are the elementwise square $f(x) = x^2$ and the ℓ_2 norm. For the square function, for instance, note that the dual epigraph problem can be solved by the following derivation

$$\begin{aligned}
\mathrm{prox}_{\lambda(\cdot)^2}(v) &= \underset{x}{\mathrm{argmin}}\ \lambda x^2 + \frac{1}{2}(x-v)^2 = \frac{v}{1+2\lambda}\\
&\underset{(18)}{\Longrightarrow} P_{\lambda f}(v) = \frac{\lambda}{1+2\lambda}v^2\\
&\underset{(17)}{\Longrightarrow} \frac{d}{d\lambda}\left(\frac{\lambda v^2}{1+2\lambda} - \frac{1}{2}\lambda^2 - \lambda s\right) = 0\\
&\Longrightarrow (\frac{1}{2}\lambda + s)(1+2\lambda)^2 - v^2 = 0
\end{aligned} \tag{20}$$

which is a cubic equation in λ. Similarly the epigraph projection for the ℓ_2 norm is simply the well-known projection on to the second order cone, which has a standard closed form solution.

4.2. Primal-dual Newton method

For cases where the proximal operator itself has no closed form, and the domain of the function is unconstrained, we can employ a primal-dual Newton method to solve the resulting epigraph form. Intuitively, this simply involves using Newton's method to directly solve the Lagrangian minimax problem for the epigraph projection operator, resulting in the optimality conditions

$$r(x,t,\lambda) = \begin{bmatrix} x - v + \lambda\nabla_x f(x) \\ t - s - \lambda \\ f(x) - t \end{bmatrix} = 0, \tag{21}$$

which can be solved by computing the Newton direction Δ

$$\begin{bmatrix} I + \lambda\nabla_x^2 f(x) & 0 & \nabla_x f(x) \\ 0 & 1 & -1 \\ \nabla_x f(x)^T & -1 & 0 \end{bmatrix} \Delta = -r(x,t,\lambda) \tag{22}$$

and taking step sizes according to the standard line search for primal-dual methods (Boyd & Vandenberghe, 2004, pg. 612). Note that although we are optimizing over $\lambda \geq 0$, this constraint does not affect the optimization problem, because we know the optimal solution must have $\lambda > 0$ unless $f(v) \leq s$, so we can treat the overall maximization as an unconstrained problem. If f is an elementwise function, or if its Hessian has special structure (as in the case of log-sum-exp, where the Hessian is diagonal plus rank one), then the Newton step can be computed in $O(n)$ time, and the overall complexity of the algorithm is $O(n)$ times the number of Newton iterations (which in theory can vary between $O(\log 1/\epsilon)$ or $O(\log\log 1/\epsilon)$, but which in practice is very small, often around 10 iterations to reach machine precision for these problems). Epigraph projections of the exponential, logistic, and log-sum-exp functions all take this form.

4.3. Implicit dual Newton method

Sometimes the domain of the function is constrained and the proximal operator has no closed-form solution, but can still be solved efficiently. In these cases, the primal-dual Newton method for epigraph projection is not usable, as we would need to explicitly include the primal constraints. However, it is still possible to solve the dual problem even if we do not have a explicit dual formulation. Let

$$L(x,t,\lambda) \equiv \frac{1}{2}\|(x,t) - (v,s)\|^2 + \lambda(f(x) - t) \tag{23}$$

be the Lagrangian for the epigraph projection problem. Consider $\nabla_{(x,t)}L(x) = 0$ to be a constraint, whose solution is the prox operator. Further, the solution $(x(\lambda), t(\lambda)) = (\mathrm{prox}_{\lambda f}(v), s+\lambda)$ can be regarded as a function of λ. Thus, if we apply the implicit function theorem (see e.g. Dontchev & Rockafellar, 2009) on $\nabla_x L = 0$, we have that

$$\frac{dx}{d\lambda} = -(I + \lambda\nabla_x^2 f(x))^{-1}\nabla_x f(x), \tag{24}$$

for all $x = \mathrm{prox}_{\lambda f}$. Then, for the dual objective

$$D(\lambda) \equiv \min_{x,t} L(x,t,\lambda) = L(x(\lambda), t(\lambda), \lambda), \tag{25}$$

Algorithm 1 Implicit dual newton method for epigraph projection

```
$\lambda := 1$
while not yet converged do
    $x := \mathrm{prox}_{\lambda f}(v)$;
    $\frac{dD(\lambda)}{d\lambda} := f(x) - \lambda - s$;
    $\frac{d^2D(\lambda)}{d\lambda^2} := -\nabla f(x)^T (I + \lambda \nabla^2 f(x))^{-1} \nabla f(x) - 1$;
    if $|\frac{dD(\lambda)}{d\lambda}| \le \epsilon$ then break;
    $\lambda := \max(0, \lambda - \frac{dD(\lambda)}{d\lambda} / \frac{d^2D(\lambda)}{d\lambda^2})$;
end while
return $(\mathrm{prox}_{\lambda f}(v), s + \lambda)$.
```

we can derive the first and second derivative by the chain rule and equation (25):

$$\begin{aligned} \frac{dD(\lambda)}{d\lambda} &= \frac{d}{d\lambda} L(x(\lambda), t(\lambda), \lambda) = f(x) - s - \lambda, \\ \frac{d^2D(\lambda)}{d\lambda^2} &= \frac{d}{d\lambda}\frac{dD(\lambda)}{d\lambda} = \left(\frac{df(x)}{dx}\right)^T \frac{dx}{d\lambda} - 1. \end{aligned} \tag{26}$$

Note that the relation holds only on $x = \mathrm{prox}_{\lambda f}(v)$. By using these first and second derivatives, we can apply Newton's method on the dual problem without computing it directly, an approach we call the implicit dual Newton method. If the proximal operator is also solved by a Newton method, the total number of iteration performed can be $(\#Newton)^2$, but in practice is still usually very small.

Negative log epigraph The proximal operator of $-\sum_i \log(x_i)$ has the analytic solution

$$x_i = \frac{v_i + \sqrt{v_i^2 + 4\lambda}}{2}, \quad \text{where } i = 1, \ldots, n. \tag{27}$$

By the implicit function theorem, we have

$$\frac{dx_i}{d\lambda} = \frac{\frac{1}{x_i}}{1 + \lambda \frac{1}{x_i^2}}, \ \forall x = \mathrm{prox}_{\lambda f}(v). \tag{28}$$

Thus, the derivatives of D can be computed as

$$\begin{aligned} \frac{d}{d\lambda} D &= -\lambda - s - \sum_i \log x_i, \\ \frac{d^2}{d\lambda^2} D &= -\lambda \sum_{i=1}^n \frac{1}{x_i^2} \left(\frac{1}{1 + \lambda \frac{1}{x_i^2}} \right) - 1. \end{aligned} \tag{29}$$

4.4. Sum of max solvers

Several atomic convex functions are non-smooth, presenting problems for the above approaches. For example, the hinge loss, the absolute value, and the max of elements are all piecewise linear functions. By inspecting the KKT conditions, we find that the dual solution λ of these problems all satisfies the equation

$$F(\lambda) = \sum_{i=1}^n \max(0, a_i\lambda + y_i) + b\lambda + c = 0, \tag{30}$$

in which $F(\lambda)$ is also a piecewise linear function. The above equation can be solved in $O(n)$ time by enumerating all the knot points and performing a quick-select or median-of-median algorithm (Cormen, 2009). The sum-k-largest proximal operator can also be solved by a similar equation with the max functions replaced by the clipping functions. We present the simplified sum-of-max algorithm for the absolute value epigraph in Algorithm 2.

Algorithm 2 Linear time algorithm for absolute epigraph

```
$y := u, \ a = -s, \ b = 0$;
while $y$ is not empty do
    Choose a mid point $y_m$ in $y$, assign $\lambda = |y_m|$;
    Denote $y_{(op)}$ as the subvector satisfying $[y \ (op) \ \lambda]$,
    and $\#(y_{(op)})$ as the length of the vector;
    Partition $y$ to $y_<$, $y_>$, and $y_=$ by $\lambda$,
    $g := a + \|y_>\|_1 - \lambda \cdot (1 + b + \#(y_>))$;
    if $g < 0$ then
        $a := a + \|y_=\|_1 + \|y_>\|_1, b := b + \#(y_=) + \#(y_>)$;
        $y := y_\le$;
    else if $g > 0$ then $y := y_\ge$;
    else break;
end while
$x_i := \mathrm{sign}(x_i)\max(0, |x_i| - \lambda), \ \forall i$,
$t = s + \lambda$;
return $(x, t)$;
```

Absolute value epigraph (ℓ_1 norm) As a concrete example, the epigraph projection problem of $f(x) = \|x\|_1$ can be formulated as

$$\min_{x,t} \frac{1}{2}\|(x,t) - (v,s)\|^2, \quad \text{such that } \|x\|_1 \le t. \tag{31}$$

The above problem is equivalent to solving

$$F(\lambda) = \sum_i \max(0, \ |v_i| - \lambda) - \lambda - s = 0 \tag{32}$$

which can be solved by the sum-of-max method. We then recover the epigraph projection (x, t) by

$$x_i = \mathrm{sign}(v_i)\max(0, \ |v_i| - \lambda), \ \ t = \lambda + s. \tag{33}$$

4.5. Orthogonal matrix epigraph solvers

Finally, we consider the case of functions that take matrix inputs $f : \mathbb{R}^{m \times n} \to \mathbb{R}$. Although many such functions can be expressed just as equivalent vector/elementwise operations, we focus here on the class of orthogonally invariant matrix functions, meaning that

$$f(UXV^T) = f(X) \tag{34}$$

for all orthogonal U, V. The immediate implication of this fact is that f can depend only on the singular values $\sigma(X)$ of X, since given the singular value decomposition $X = USV^T$

$$F(X) = F(U^T XV) = F(S). \tag{35}$$

A further consequence is that if $f(X) = g(\sigma(X))$ for some vector (or elementwise) function g, then the epigraph projection of f can be computed from the epigraph projection of g (a similar property holds for proximal operators)

$$\mathrm{epi}_f(W, s) = \left(\begin{smallmatrix} U & 0 \\ 0 & 1 \end{smallmatrix}\right) \mathrm{epi}_g(\sigma(W), s) \left(\begin{smallmatrix} V & 0 \\ 0 & 1 \end{smallmatrix}\right)^T \tag{36}$$

for $W = U\sigma(W)V^T$. This property follows from the fact that the squared Frobenius norm is also orthogonally invariant. Using this property, we can immediately develop epigraph operators for matrix expressions such as the negative logdet term ($-\log$ applied to singular values), the nuclear norm (ℓ_1 norm applied to singular values), and the operator norm (maximum singular value).

5. Experiments

In this section we compare our method to existing methods for general convex programming based on conic solvers. Our approach is implemented in Epsilon (Epigraph Proximal Solver), a library based on the ideas described in this paper: Epsilon accepts general convex programs specified according to the DCP ruleset, transforms them to a sum of proximal and epigraph operators as in Theorem 1, and solves them by employing the library of operator implementations described previously. Epsilon is open source and available at `http://epopt.io`, and all examples here are included in the distribution. We highlight the benchmark problems briefly in this section, and include a full description in Appendix A. Epsilon integrates directly with cvxpy (Diamond & Boyd, 2015) and thus we make the natural comparison between Epsilon and the existing solvers which solve the conic form. In particular, we compare Epsilon to ECOS (Domahidi et al., 2013), an interior point method, and SCS (O'Donoghue et al., 2013), the "splitting conic solver". In general, interior point methods achieve highly accurate solutions but have trouble scaling to larger problems and so it is unsurprising that Epsilon is able to solve problems to moderate accuracy several orders of magnitude faster than ECOS. On the other hand, SCS employs an operator splitting method that is similar in spirit to Epsilon, both being variants of ADMM. The main difference between Epsilon and SCS is in the intermediate representation to which operator splitting is applied: SCS solves problems that have been reduced to cone form (with operators for cone projections) while Epsilon solves the higher-level representation with a larger library of proximal and epigraph operators.

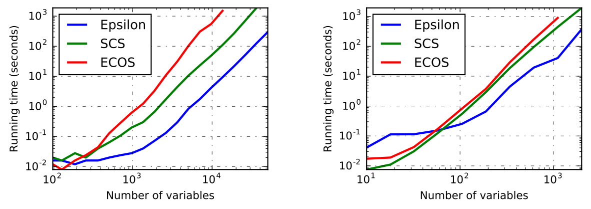

Figure 2. Scaling comparison on Lasso (left) for problems with n examples and $10n$ variables, and for support vector data description (right) with $10n$ examples and n variables.

Standard problems We begin with a few standard machine learning problems: Lasso (Tibshirani, 1996), sparse inverse covariance estimation (Banerjee et al., 2008) and image classification on MNIST (LeCun et al., 1998). Table 1 shows that on these problems Epsilon achieves the same level of accuracy as existing approaches often several orders of magnitude faster. These are moderately-sized problems by ML standards, i.e. 1500 examples, 5000 features for Lasso; 2000 examples, 1000 features for MNIST; and a 200×200 covariance estimation problem. However, interior point methods (ECOS) take minutes to hours on these problems, demonstrating the lack of scalability traditionally associated with general convex programming. SCS (which itself is relatively recent work) does improve on ECOS by an order of magnitude, but Epsilon is still 1-2 orders of magnitude faster. Figure 2 compares the running times of each approach on the Lasso problem as problem size increases: Epsilon solves larger problems in minutes which require hours for SCS and ECOS.

Next, we consider several problems involving more complex objectives that go beyond the standard regularized loss functions for regression and classification found in traditional machine learning models (again, described much more fully in Appendix A).

Robust SVM We consider the "ℓ_∞" robust SVM (based upon those in (Lanckriet et al., 2003; Shivaswamy et al., 2006) but with an ℓ_∞ uncertainy ball) described above by the optimization problem (8).

Robust regression We consider the Chebyshev-like regression problem (Boyd & Vandenberghe, 2004, pg. 323), which solves a least-squares problem with A in some uncertainty set described by $\bar{A} + \sum_{i=1}^k u_i A_i$ for $\|u\|_2 \le 1$. This leads to the optimization problem

$$\underset{x}{\text{minimize}} \quad \max_{i=1,\ldots,k} \left(\|A_i x\|_2 + |\bar{a}_i^T x - b_i| \right). \tag{37}$$

	Epsilon		SCS		ECOS	
	Time	Objective	Time	Objective	Time	Objective
Lasso	3.7s	3.21×10^1	20.5s	3.21×10^1	215.7s	3.21×10^1
MNIST (2000 examples)	2.53s	1.72×10^3	219.6s	1.72×10^3	1753.0s	1.72×10^3
Sparse inverse covariance	2.09s	3.73×10^2	25.1s	3.73×10^2	-	-
Robust SVM	52.9s	2.46×10^1	174.6s	2.48×10^3	141.6s	2.44×10^1
Robust regression	27.3s	3.95×10^{-1}	66.5s	3.95×10^{-1}	67.9s	3.95×10^{-1}
SV data description	40.5s	2.23×10^2	443.6s	2.23×10^2	893.3s	2.23×10^2
Sum-k-largest softmax	0.94s	2.14×10^1	325.28s	2.14×10^1	13.55s	2.14×10^1

Table 1. Comparison of running time and objective value for Epsilon, SCS and ECOS, "-" indicates no result after 1 hour.

Sum-k-largest softmax We consider a form of "worst-case" softmax regression, where we minimize the loss suffered by only the top k elements in the data set. This can be written as the optimization problem

$$\underset{\Theta}{\text{minimize}} \quad \sum_{i=1}^{k} z_{[i]} + \lambda \|\Theta\|_2^2, \tag{38}$$

where $z_{[i]}$ is the i-largest element of vector z, and $z_i = -\log \text{softmax}(x_i, y_i, \Theta)$ is the multiclass softmax loss.

Support vector data description (SVDD) Given a set of points, $x_1, \ldots, x_m \in \mathbb{R}^n$, SVDD (Tax & Duin, 2004; Chang et al., 2007) describes those points with an n-dimensional Euclidean ball by solving

$$\underset{\rho, a}{\text{minimize}} \quad \sum_{i=1}^{m} [\|x_i - a\|_2^2 - \rho)]_+ + \lambda[\rho]_+ \tag{39}$$

with optimization variables $\rho \in \mathbb{R}$ and $a \in \mathbb{R}^n$.

The appeal of general convex programming is that models such as these can be rapidly developed and prototyped; however, as with simpler problems, existing solvers often have difficultly scaling to larger problem sizes. Table 1 shows that for these problems Epsilon is at least 2-3x faster than existing approaches and in some cases reduces running times by 5-10x. For support vector data description, Figure 2 shows that for larger problems Epsilon is significantly faster than SCS and ECOS, e.g. for a dataset $x_1, \ldots, x_{20000} \in \mathbb{R}^{2000}$, Epsilon requires roughly 6.5 minutes, SCS 32 minutes and ECOS more than 1 hour.

In Figure 3 we also highlight, for the Lasso and SVDD problems, the evolution of primal objective versus time. We see that in both cases, Epsilon consistently has a lower objective value after the same amount of computation time; this holds for all the examples in Table 1, though we do not include all plots for brevity. One element to note, however, is that unlike interior points methods, ADMM-based methods cannot easily produce guaranteed bounds on the suboptimality of a solution; although we do not pursue this issue here, and simply show that the resulting solutions are far

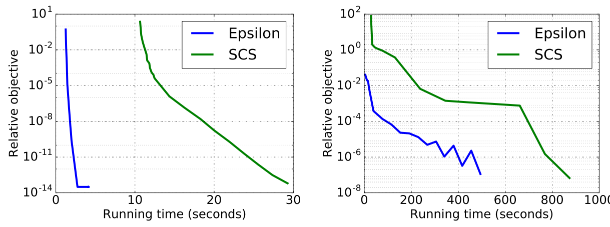

Figure 3. Comparison on time and primal objective suboptimality for Lasso problem (top) and SVDD problem (bottom).

superior given reasonable running times, determining if we can produce guaranteed bounds in our setting is an important question for future work. Finally, although it is not the focus of this paper, we also compare Epsilon to several specialized solvers developed for specific classes of problems, and find that in many cases it is quite competitive, despite being a fully generic approach; these results are presented in Appendix B.

6. Conclusion

This paper has focused on the development of fast optimization methods for general convex programs, here specified using a DCP modeling framework. We show that by implementing just proximal and epigraph projection operators for a set of atomic convex functions, we can solve any DCP built from those atoms. We have then implemented a wide set of such operators and we show that the resulting solution methods can substantially outperform existing approaches.

Acknowledgements This work was supported in part by the National Science Foundation under award IIS-1320402, and by support from Google. Matt Wytock was partly funded by a scholarship from the Siebel Foundation.

References

Banerjee, Onureena, El Ghaoui, Laurent, and d'Aspremont, Alexandre. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

Becker, Stephen R, Candès, Emmanuel J, and Grant, Michael C. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

Boyd, Stephen and Vandenberghe, Lieven. *Convex optimization*. Cambridge university press, 2004.

Boyd, Stephen, Parikh, Neal, Chu, Eric, Peleato, Borja, and Eckstein, Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Chang, Chien-Chung, Tsai, Hsi-Chen, and Lee, Yuh-Jye. A minimum enclosing balls labeling method. 2007.

Chang, Chih-Chung and Lin, Chih-Jen. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chierchia, Giovanni, Pustelnik, Nelly, Pesquet, Jean-Christophe, and Pesquet-Popescu, Béatrice. Epigraphical Projection and Proximal Tools for Solving Constrained Convex Optimization Problems: Part I. Technical report, October 2012.

Cormen, Thomas H. *Introduction to algorithms*. MIT press, 2009.

Diamond, Steven and Boyd, Stephen. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 2015. To appear.

Domahidi, Alexander, Chu, Eric, and Boyd, Stephen. Ecos: An socp solver for embedded systems. In *Control Conference (ECC), 2013 European*, pp. 3071–3076. IEEE, 2013.

Donoho, David L. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627, 1995.

Dontchev, Asen L and Rockafellar, R Tyrrell. *Implicit functions and solution mappings*. Springer, 2009.

Douglas, Jim and Rachford, Henry H. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, pp. 421–439, 1956.

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Rob. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33 (1):1, 2010.

Gabay, Daniel and Mercier, Bertrand. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

Grant, Michael and Boyd, Stephen. Graph implementations for nonsmooth convex programs. In Blondel, V., Boyd, S., and Kimura, H. (eds.), *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag Limited, 2008.

Grant, Michael, Boyd, Stephen, and Ye, Yinyu. Disciplined convex programming. In Liberti, Leo and Maculan, Nelson (eds.), *Global Optimization*, volume 84 of *Nonconvex Optimization and Its Applications*, pp. 155–210. Springer US, 2006. ISBN 978-0-387-28260-2. doi: 10.1007/0-387-30528-9_7.

Harizanov, Stanislav, Pesquet, Jean-Christophe, and Steidl, Gabriele. *Scale Space and Variational Methods in Computer Vision: 4th International Conference, SSVM 2013, Schloss Seggau, Leibnitz, Austria, June 2-6, 2013. Proceedings*, chapter Epigraphical Projection for Solving Least Squares Anscombe Transformed Constrained Optimization Problems, pp. 125–136. Springer Berlin Heidelberg, 2013.

Hsieh, Cho-Jui, Sustik, Mátyás A., Dhillon, Inderjit S., and Ravikumar, Pradeep D. Sparse inverse covariance matrix estimation using quadratic approximation. *CoRR*, abs/1306.3212, 2013.

Hsieh, Cho-Jui, Sustik, Mátyás A, Dhillon, Inderjit S, and Ravikumar, Pradeep. Quic: quadratic approximation for sparse inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1):2911–2947, 2014.

Joachims, Thorsten. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133–142. ACM, 2002.

Lanckriet, Gert RG, Ghaoui, Laurent El, Bhattacharyya, Chiranjib, and Jordan, Michael I. A robust minimax approach to classification. *The Journal of Machine Learning Research*, 3:555–582, 2003.

LeCun, Yann, Cortes, Corinna, and Burges, Christopher JC. The mnist database of handwritten digits, 1998.

O'Donoghue, Brendan, Chu, Eric, Parikh, Neal, and Boyd, Stephen. Operator splitting for conic optimization via homogeneous self-dual embedding. *arXiv preprint arXiv:1312.3039*, 2013.

Optimization, Gurobi et al. Gurobi optimizer reference manual. *URL: http://www. gurobi. com*, 2012.

Parikh, Neal and Boyd, Stephen. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):123–231, 2013.

Platt, John C. Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-19416-3.

Rahimi, Ali and Recht, Benjamin. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2007.

Rockafellar, R Tyrrell. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

Ryu, Ernest K and Boyd, Stephen. Primer on monotone operator methods. *To appear, Appl. Comput. Math.*, 15 (1), 2016.

Shalev-Shwartz, Shai, Singer, Yoram, Srebro, Nathan, and Cotter, Andrew. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

Shivaswamy, Pannagadatta K, Bhattacharyya, Chiranjib, and Smola, Alexander J. Second order cone programming approaches for handling missing and uncertain data. *The Journal of Machine Learning Research*, 7: 1283–1314, 2006.

Tax, David MJ and Duin, Robert PW. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

Tibshirani, Robert. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Tofighi, Mohammad, Bozkurt, Alican, Kose, Kivanc, and Cetin, A Enis. Deconvolution using projections onto the epigraph set of a convex cost function. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pp. 1638–1641. IEEE, 2014.

A. Experiment details

In this section we provide the details of the experiments corresponding to the results presented in Section 5. In each case we describe the underlying problem, the initial optimization problem, and how this translates into proximal form for the Epsilon solver (that is, a sum of objective terms where each term has an efficient proximal operator or epigraph projection, plus a set of linear equality constraints), using the transformations from Section 3. All these examples are included in the Epsilon distribution, available at `http://epopt.io`.

A.1. Lasso

The Lasso problem is

$$\underset{\theta}{\text{minimize}} \;\; (1/2)\|X\theta - y\|_2^2 + \lambda\|\theta\|_1, \tag{40}$$

with input features $X \in \mathbb{R}^{m\times n}$, response variables $y \in \mathbb{R}^m$, and model parameters $\theta \in \mathbb{R}^n$. The regularization parameter $\lambda \geq 0$ controls the tradeoff between data fit and the ℓ_1 regularization term which encourages sparsity in the model parameters. The Lasso is especially useful in the high-dimensional case where $m < n$ as the sparsity induced by ℓ_1 regularization effectively controls the number of free parameters in the model, see Tibshirani (1996) for details. The Lasso problem as written is already in proximal form with

$$\begin{aligned} f_1(\theta) &= (1/2)\|X\theta - y\|_2^2, \\ f_2(\theta) &= \lambda\|\theta\|_1. \end{aligned} \tag{41}$$

In our experiments, we generate $X \in \mathbb{R}^{1500\times 5000}$ from a standard Normal distribution and set $y = X\theta_0 + \epsilon$ with θ_0 having 1% nonzero standard Normal entries and $\epsilon \sim \mathcal{N}(0, 0.05^2)$. We set the regularization parameter $\lambda = 0.5\|X^T y\|_\infty$.

A.2. Sparse inverse covariance

Sparse inverse covariance estimation models a multivariate Gaussian distribution over n variables by solving the optimization problem

$$\underset{\Theta}{\text{minimize}} \;\; -\log|\Theta| + \operatorname{tr} S\Theta + \lambda\|\Theta\|_1 \tag{42}$$

where $S \in \mathbb{S}^n$ is the sample covariance, $\Theta \in \mathbb{S}^n$ is the estimated inverse covariance and the ℓ_1 norm $\|\cdot\|_1$ is applied elementwise. As with the Lasso, the ℓ_1 penalty promotes sparse structure and is especially useful in the high-dimensional case with more variables than samples ($m < n$), see Friedman et al. (2008) for details. With respect to our framework, given a proximal operator for the $-\log|\cdot|$ term, sparse inverse covariance estimation can trivially be put in proximal form with

$$\begin{aligned} f_1(\Theta) &= -\log|\Theta| + \operatorname{tr} S\Theta, \\ f_2(\Theta) &= \lambda\|\Theta\|_1. \end{aligned} \tag{43}$$

where in the construction of f_1, we have exploited the fact that any proximal operator can easily be combined with a linear function (see e.g. Parikh & Boyd (2013) for details). In our experiments, we construct the sample covariance from 100 samples drawn from $\mathcal{N}(0, \Theta_0^{-1})$ where $\Theta_0 \in \mathbb{S}^{200}$ has uniform random entries and is 1% nonzero.

A.3. MNIST (2000 images)

The MNIST dataset (LeCun et al., 1998) consists of handwritten digits, constructed with the goal of building a classifier for automatically recognizing each digit ($\{0,\ldots,9\}$) in each image. As a linear classifier applied directly to the raw pixels performs poorly, we generate random Fourier features (Rahimi & Recht, 2007) and train a classifier using sparse softmax regression

$$\underset{\Theta}{\text{minimize}} \;\; \ell(X, y; \Theta) + \lambda\|\Theta\|_1 \tag{44}$$

where $X \in \mathbb{R}^{m\times n}$ are the image features, $y \in \{0,\ldots,9\}^m$ are the image labels, $\lambda \geq 0$ is the regularization parameter and the softmax loss, parameterized by $\Theta \in \mathbb{R}^{n\times 10}$, is given by

$$\ell(X, y; \Theta) = \sum_{i=1}^m \left(\log\sum_{k=1}^{10}\exp(x_i^T\theta_k) - x_i^T\theta_{y_i}\right). \tag{45}$$

Unlike the least squares loss employed in the Lasso, the softmax loss cannot easily be composed with an arbitrary linear function and this requires the introduction of an additional auxiliary variable, $Z \in \mathbb{R}^{m\times 10}$. With this additional variable, the proximal form for this problem is given by

$$\begin{aligned} f_1(Z) &= \sum_{i=1}^m \left(\log\sum_{k=1}^{10}\exp(Z_{ik})\right) \\ f_2(\Theta) &= \lambda\|\Theta\|_1 - \sum_{i=1}^m x_i^T\theta_{y_i} \end{aligned} \tag{46}$$

with equality constraints

$$Z = X\Theta \tag{47}$$

In our experiments, we train the classifier on 2000 images using 1000 random Fourier features and $\lambda = 0.1$.

A.4. Robust SVM

In our experiments we use an ℓ_∞ variant of the support vector machine, similar to the formulations in (Lanckriet et al., 2003; Shivaswamy et al., 2006), but with an ℓ_∞ uncertainty

Function			Proximal / epigraph operator	
Type	**Atom**	**Definition**	**Method**	**Complexity**
Elementwise $x, y \in \mathbb{R}$	Absolute	$f(x) = \lvert x\rvert \ (\equiv \lVert x\rVert_1)$	soft thresholding sum-of-max	$O(n)$ $O(n)$
	Square	$f(x) = x^2 \ (\equiv \lVert x\rVert_2^2)$	linear equation cubic equation	$O(n)$ $O(n)$
	Hinge	$f(x) = \max\{x, 0\}$	soft thresholding sum-of-max	$O(n)$ $O(n)$
	Deadzone	$f(x) = \max\{\lvert x\rvert - \epsilon, 0\},\ \epsilon \ge 0$	soft thresholding sum-of-max	$O(n)$ $O(n)$
	Quantile	$f(x) = \max\{\alpha x, (\alpha-1)x\},\ 0 \le \alpha \le 1$	soft thresholding sum-of-max	$O(n)$ $O(n)$
	Logistic	$f(x) = \log(1 + \exp(x))$	Newton Primal-dual Newton	$O(n) \cdot$ (# Newton) $O(n) \cdot$ (# Newton)
	Inverse positive	$f(x) = 1/x,\ x \ge 0$	cubic equation Implicit dual Newton	$O(n)$ $O(n) \cdot$ (# Newton)
	Negative log	$f(x) = -\log(x),\ x \ge 0$	quadratic equation Implicit dual Newton	$O(n)$ $O(n) \cdot$ (# Newton)
	Exponential	$f(x) = \exp(x)$	Newton Primal-dual Newton	$O(n) \cdot$ (# Newton) $O(n) \cdot$ (# Newton)
	Negative entropy	$f(x) = x \cdot \log(x),\ x \ge 0$	Projected Newton Implicit dual Newton	$O(n) \cdot$ (# Newton) $O(n) \cdot (\text{\# Newton})^2$
	KL Divergence	$f(x, y) = x \cdot \log(x/y),\ x, y \ge 0$	Projected Newton Implicit dual Newton	$O(n) \cdot$ (# Newton) $O(n) \cdot (\text{\# Newton})^2$
	Quadratic over linear	$f(x, y) = x^2/y,\ y \ge 0$	Projected Newton Implicit dual Newton	$O(n) \cdot$ (# Newton) $O(n) \cdot (\text{\# Newton})^2$
Vector $x \in \mathbb{R}^n$	ℓ_2-norm	$f(x) = \lVert x\rVert_2$	group soft thresholding analytic projection	$O(n)$ $O(n)$
	Maximum	$f(x) = \max_i x_i \ \ (\lVert x\rVert_\infty \equiv \max_i \lvert x_i\rvert)$	sum-of-max sum-of-max	$O(n)$ $O(n)$
	Sum-k-largest	$f(x) = \sum_{i=1}^k x_{[i]} \ \ (x_{[i]} \ge x_{[i+1]})$	sum-of-clip bisection	$O(n)$ $O(n) \cdot$ (# Bisection)
	Log-sum-exp	$f(x) = \log\left(\sum_{i=1}^n \exp(x_i)\right)$	Newton Primal-dual Newton	$O(n) \cdot$ (# Newton) $O(n) \cdot$ (# Newton)
Matrix $X \in \mathbb{R}^{n\times n}$	Negative log det Nuclear norm Spectral norm	$f(X) = -\log\det(X), X \in \mathbb{S}^n$ $f(X) = \lVert\sigma(X)\rVert_1, X \in \mathbb{R}^{m\times n}$ $f(X) = \lVert\sigma(X)\rVert_\infty, X \in \mathbb{R}^{m\times n}$	$-\log$ on $\lambda(X)$ $\lVert\cdot\rVert_1$ on $\sigma(X)$ $\lVert\cdot\rVert_\infty$ on $\sigma(X)$	$O(n^3)$ $O(n^3)$ $O(n^3)$

Table 2. Complete list of proximal and epigraph projection operators implemented for this work. Most proximal operators (except sum-k-largest) have appeared in some form in previous literature, but epigraph projections are typically novel to this work.

ball instead of an ℓ_2 uncertainty ball. In this setting, given input data (x_i, y_i) we wish to train an SVM by minimizing the standard regularized hinge loss,

$$\underset{\theta}{\text{minimize}} \quad \frac{\lambda}{2}\lVert\theta\rVert_2^2 + \sum_{i=1}^m \max\{0, 1 - y_i \cdot \bar{x}_i^T\theta\} \tag{48}$$

but where $\bar{x}_i$ lies in some uncertainty set centered at x_i, $\bar{x}_i = x_i + Pu$ where $\lVert u\rVert_\infty \le 1$. This can be expressed as the optimization problem

$$\underset{\theta}{\text{minimize}} \quad \frac{\lambda}{2}\lVert\theta\rVert_2^2 + \sum_{i=1}^m \sup_{\lVert u_i\rVert_\infty \le 1} \max\{0, 1 - y_i \cdot \theta^T(\bar{x}_i + Pu_i)\} \tag{49}$$

which, using the relation that $\sup_{\lVert u\rVert_\infty \le 1} c^T u = \lVert c\rVert_1$ where, is equivalent to the optimization problem

$$\underset{\theta}{\text{minimize}} \quad \frac{\lambda}{2}\lVert\theta\rVert_2^2 + \sum_{i=1}^m \max\{0, 1 - y_i \cdot \theta^T x_i + \lVert P^T\theta\rVert_1\}. \tag{50}$$

As discussed in Section 3, this is transformed to proximal form

$$\begin{aligned} f_1(\theta) &= \frac{\lambda}{2}\lVert\theta\rVert_2^2 \\ f_2(z_3) &= \sum \max\{z_3, 0\} \\ f_3(z_1, z_2) &= I\{\lVert z_1\rVert_1 \le z_2\} \end{aligned} \tag{51}$$

and equality constraints

$$\begin{aligned} z_1 &= P^T x \\ z_3 &= 1 - \mathrm{diag}(y) X\theta + 1 z_2. \end{aligned} \tag{52}$$

In our experiments, we generated $X \in \mathbb{R}^{2500\times 750}$ random uniform [0,1], $\theta \in \mathbb{R}^{750}$ also random uniform, and set $y = \mathrm{sign}(x_i^T\theta + \mathcal{N}(0, 0.1))$. To create well-separated points, we further added $x_i \leftarrow x_i + 0.7 y_i \cdot \theta$, and chose $P = \mathrm{diag}(\mathrm{N}(0, 750))$.

Support vector data description Given a set of unlabeled points, $x_1, \ldots, x_m \in \mathbb{R}^n$, support vector data description (Tax & Duin, 2004; Chang et al., 2007) describes those points with an n-dimensional Euclidean ball by solving

$$\underset{\rho, a}{\text{minimize}} \quad \sum_{i=1}^{m} [\|x_i - a\|_2^2 - \rho)]_+ + \lambda[\rho]_+ \tag{53}$$

with optimization variables $\rho \in \mathbb{R}$ and $a \in \mathbb{R}^n$. The first term penalizes points outside a ball centered at a with radius $\sqrt{\rho}$ while the second term regularizes the radius with $\lambda \geq 0$ controlling the tradeoff. This problem is transformed to proximal form with operators

$$\begin{aligned} f_1(t, \rho) &= \sum_{i=1}^{m} [t_i]_+ + \lambda[\rho]_+ \\ f_3(a, s) &= \sum_{i=1}^{m} I(\|x_i - a\|_2^2 \leq s_i) \end{aligned} \tag{54}$$

and equality constraint

$$t = s - \rho. \tag{55}$$

In our experiments, we generate 5000 random points uniformly over the 200-dimensional unit hypersphere and then choose 100 outliers at random and add noise, $\epsilon \sim \mathcal{N}(0, I)$; we fit the model with $\lambda = 1$.

Robust Regression Consider a noisy matrix bounded by a unit ball over some known perturbation directions A_i,

$$\mathcal{A} = \{\bar{A} + c_1 A_1 + \ldots + c_p A_p \mid \|p\|_2 \leq 1\}. \tag{56}$$

The maximum error incurred from performing linear regression over the uncertain set can be written as

$$\begin{aligned} &\sup_{\|c\|_2 \leq 1} \|(\bar{A} + c_1 A_1 + \ldots + c_p A_p)x - b\|_\infty \\ &= \max_k \left| \sup_{\|c\|_2 \leq 1} c_k (A_k x) + (\bar{a}_k^T x - b_k) \right| \\ &= \max_k \left| \|A_k x\|_2 + |\bar{a}_k^T x - b_k| \right|. \end{aligned} \tag{57}$$

For robust regression (Boyd & Vandenberghe, 2004, pg. 323), we wish to find a solution x that minimize this worst possible error,

$$\underset{x}{\text{minimize}} \ \max_k \left| \|A_k x\|_2 + |\bar{a}_k^T x - b_k| \right|. \tag{58}$$

With additional variable $t, u, v, p, q \in \mathbb{R}^k$, this problem is transformed to proximal form with operators

$$\begin{aligned} f_1(t) &= \max_{i=1,\ldots,k} (t_i), \\ f_2(p, u) &= \sum_{i=1}^{k} \mathcal{I}(\|p_i\|_2 \leq u_i), \\ f_3(q, v) &= \sum_{i=1}^{k} \mathcal{I}(|q_i| \leq v_i), \end{aligned} \tag{59}$$

with equality constraints

$$\begin{aligned} t &= u + v, \\ p_i &= A_i x, \\ q_i &= \bar{a}_i^T x - b_i, \ \forall i = 1, \ldots, k. \end{aligned} \tag{60}$$

In the experiment, we generate the $\bar{A}$, A_i, and b from uniform distribution, then normalize $\bar{A}$ and A_i to a unit ball. We choose $p = 5000$ and $A, A_i \in \mathbb{R}^{10\times 200}$, for $i = 1, \ldots, p$.

sum-k-largest softmax The softmax loss is a multiclass loss function defined as

$$\text{softmax}(x, y, \Theta) = \frac{\exp(x^T \Theta_{y_i})}{\sum_k \exp(x^T \Theta_k)}, \tag{61}$$

where $\Theta \in \mathbb{R}^{n\times c}$ is the weight for each class. The softmax loss is commonly used in, for example, the regularized logistic regression, which can be formulated by the softmax loss plus a regularization term. Here, we consider to minimize the worst k loss incurred from the regression. I.e., we only minimize

$$\underset{\Theta}{\text{minimize}} \quad \sum_{i=1}^{k} z_{[i]} + \lambda\|\Theta\|_2^2, \tag{62}$$

where $z_{[i]}$ is the i-largest element of vector z, and $z_i = -\log \text{softmax}(x_i, y_i, \Theta)$ is the multiclass softmax loss. With additional variable $u_i \in \mathbb{R}^c$, $\forall i = 1, \ldots, m$, this problem is transformed to proximal form with operators

$$\begin{aligned} f_1(\Theta) &= -\sum_{i=1}^{m} x_i^T \Theta_{y_i} + \lambda\|\Theta\|_2^2, \\ f_2(z) &= \text{sum-}k\text{-largest}(z), \\ f_3(\Theta, z) &= \mathcal{I}(\text{log-sum-exp}(u_i) \leq z_i), \end{aligned} \tag{63}$$

Solver	Problem	Time	
		Epsilon	Solver
liblinear	hinge_l1	3.71s	0.49s
	hinge_l1_sparse	14.26s	4.26s
	hinge_l2	3.58s	0.16s
	hinge_l2_sparse	1.82s	0.83s
glmnet	lasso	3.69s	0.84s
	lasso_sparse	13.58s	0.67s
	logreg_l1	3.70s	2.31s
	logreg_l1_sparse	6.69s	1.96s
	mv_lasso	7.14s	7.40s
Gurobi	lp	0.33s	6.02s
	qp	1.39s	4.12s
QUIC	covsel	0.93s	6.24s

Table 3. Comparison of running times between Epsilon and specialized solvers.

with equality constraints

$$u_i = x_i^T \Theta,\ \forall i = 1, \dots, m. \tag{64}$$

In the experiment we choose $X \in \mathbb{R}^{400\times 10}$, $k = 5$, and the number of classes to be 120. We generate the data from normalized uniform distribution, and assign classes uniformly.

B. Comparison with specialized solvers

In this section, we compare Epsilon to an assortment of specialized solvers which are available for common problems. Before doing so, we emphasize that the general convex programming approach offers many advantages to specialized algorithms in terms of reuse and extensibility. In addition, most convex problems do not have dedicated, mature software packages readily available in common mathematical programming environments (e.g. Matlab, R, Python). Furthermore, even when specialized solvers are available, translating problems to the interface provided by a particular package requires effort to understand and conform to the idiosyncrasies of each implementation. In contrast, general convex programming offers a uniform syntax and interface allowing problems to be easily formulated, extended and solved.

In terms of running times, Table 3 compares Epsilon to four dedicated software packages implementing specialized algorithms: liblinear (Fan et al., 2008), glmnet (Friedman et al., 2010), Gurobi (Optimization et al., 2012) and QUIC (Hsieh et al., 2013). The default stopping criteria is used for each solver, corresponding to moderate accuracy for Epsilon and high accuracy for the specialized solvers. For the most part, Epsilon is competitive, although on a few problems liblinear and glmnet are significantly faster. This is due to the ability of these specialized algorithms to exploit sparsity in the solution which arises due to ℓ_1 regularization (Lasso problems) or a small number of support vectors in the dual SVM formulation (hinge problems). At present, Epsilon does not take advantage of such structure and thus may have a disadvantage on sparse problems.

On the other hand, Table 3 shows that Epsilon is significantly faster than Gurobi and QUIC in solving linear/quadratic programs and sparse inverse covariance estimation, respectively. However, it is important to highlight that the specialized algorithms solve these problems to high accuracy (e.g. tolerances of 10^{-8} or smaller) while Epsilon targets only moderate accuracy (e.g. 10^{-3}). For moderate accuracy, the operator splitting approach can be highly competitive, allowing Epsilon to be significantly faster on some problems.