Epigraph Projections for Fast General Convex Programming

Po-Wei Wang [1], Matt Wytock [2], J. Zico Kolter [1]

[1]Machine Learning Department and School of Computer Science,
Carnegie Mellon University

[2]Electrical Engineering Department, Stanford University

June 21, 2016

Who here has used `cvx` or `cvxpy` ?

CVX is great!

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|_2^2 + \lambda \|x\|_1$$

$$\Updownarrow$$

```
from cvxpy import *

x = Variable(n)
f = sum_squares(A*x-b) + lam * norm1(x)

prob = Problem(Minimize(f), [])
prob.solve()
```

CVX is great!

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|_2^2 + \lambda\|x\|_1 + \underline{\lambda\|x\|_2}$$

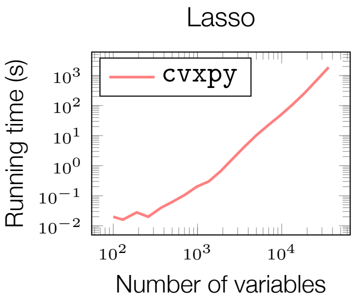
$$\Updownarrow$$

```
from cvxpy import *

x = Variable(n)
f = sum_squares(A*x-b) + lam * norm1(x) + lam * norm2(x)

prob = Problem(Minimize(f), [])
prob.solve()
```

But it is slow when it scales



100 variables	0.01 seconds
1000 variables	0.9 seconds
10000 variables	800 seconds

Well . . .

Why is `cvx(py)` slow?

Convert problem to conic form

- linear, quadratic, or semidefinite programs
- In this case, Lasso is transformed to a quadratic program

Solve with conic solver

- Primal-dual interior point method implemented in SCS and ECOS

No one would solve Lasso that way!

Contributions

Make `cvx(py)` as good as specialized solvers (almost)!

	Before		After
100 variables	0.01 seconds		0.01 seconds
1000 variables	0.9 seconds	$\Longrightarrow$	0.05 seconds
10000 variables	800 seconds		2 seconds

Same exact modeling language with flexibility (i.e., can still add additional regularizers and constraints with one line of code)

But faster!

Outline and Contribution

Solving `cvx(py)` problems without conic transform

- By proximal operators and epigraph projections

Collection of epigraph projection algorithms

- Cover many common convex functions

Experiments

- Order of magnitude faster than other solvers

Full framework available at `http://epopt.io`

Background: Proximal Methods

Proximal operator is defined by

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

A generalization of projection

Many method, e.g. proximal gradient method, use these operators

Key feature:

- Simple closed-form expression for many functions

Example: proximal operator of L1 norm

$$\mathrm{prox}_{\lambda\|\cdot\|_1} = \begin{cases} 0 & \text{if } x_i \leq \lambda \\ \mathtt{sign}(x_i)(|x_i| - \lambda) & \text{otherwise} \end{cases}$$

Why hard . . .

Why is it hard to solve `cvx(py)` problem with "fast" methods? (e.g. proximal methods)

`cvx(py)` express very general composition of function and constraints using a framework called **Disciplined Convex Programming**

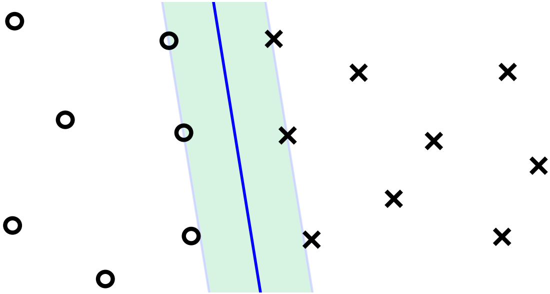
DCP involves composition of "simple" convex functions, called atoms, to create complex functions

- E.g. $\|x\|_1, \|x\|_2^2, \max(0, x)$ are all atoms.

No "easy" proximal operators for general DCP functions

Example: Robust SVM

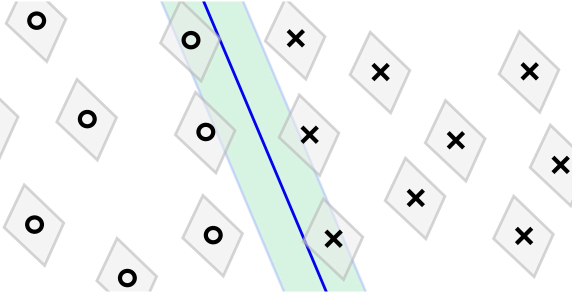
Robust SVM separates points with uncertainty regression



$$\underset{\theta}{\text{minimize}} \quad \frac{\lambda}{2}\|\theta\|_2^2 + \sum_{i=1}^{m} \max\{0, 1 - y_i \cdot \theta^T z_i\}$$

Example: Robust SVM

Robust SVM separates points with uncertainty regression



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

We have proximal operators for $\max(0, \cdot)$, $\|\cdot\|_1$, and linear functions, but we don't have the proximal for $\max\{0, 1 - y_i \cdot \theta^T z_i + \|P^T\theta\|_1\}$

Our approach

We present an algorithm that can convert any DCP into the form

$$f(x) \equiv \sum_{i=1}^{n} g_i(x), \quad \text{subject to} \quad Ax = b,$$

where $g_i(x)$ is a DCP atom

$$g_i(x) = \text{Simple functions},$$

or the indicator of an epigraph set of a DCP atom

$$g_i(x) = \mathcal{I}\{h(x) \leq x_1\}.$$

We will be able to solve the DCP problem given the proximal of $g_i(x)$

Example: Robust SVM to proximal and epigraph

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

The problem is equivalent to

$$\begin{aligned} \underset{x=\{\theta,t,p,q\}}{\text{minimize}} \quad & g_1(x) + g_2(x), \\ \text{subject to} \quad & g_1(x) = \frac{\lambda}{2}\|\theta\|_2^2 \\ & g_2(x) = \sum_{i=1}^{m} \max\{0, 1 - y_i \cdot \theta^T x_i + \|P^T\theta\|_1\} \end{aligned}$$

Example: Robust SVM to proximal and epigraph

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

The problem is equivalent to

$$\begin{aligned} \underset{x=\{\theta,t,p,q\}}{\text{minimize}} \quad & g_1(x) + g_2(x), \\ \text{subject to} \quad & g_1(x) = \frac{\lambda}{2}\|\theta\|_2^2 \\ & g_2(x) = \sum_{i=1}^{m} \max\{0, 1 - y_i \cdot \theta^T x_i + \|P^T\theta\|_1\} \end{aligned}$$

Example: Robust SVM to proximal and epigraph

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

The problem is equivalent to

$$\begin{aligned}
\underset{x=\{\theta,t,p,q\}}{\text{minimize}} \quad & g_1(x) + g_2(x) + g_3(x), \\
\text{subject to} \quad & g_1(x) = \frac{\lambda}{2}\|\theta\|_2^2 \\
& g_2(x) = \sum_{i=1}^{m} \max\{0, 1 - y_i \cdot \theta^T x_i + \underline{q}\} \\
& g_3(x) = \mathcal{I}(\underline{\|P^T\theta\|_1 \le q})
\end{aligned}$$

Example: Robust SVM to proximal and epigraph

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

The problem is equivalent to

$$\begin{aligned} \underset{x=\{\theta,t,p,q\}}{\text{minimize}} \quad & g_1(x) + g_2(x) + g_3(x), \\ \text{subject to} \quad & g_1(x) = \frac{\lambda}{2}\|\theta\|_2^2 \\ & g_2(x) = \sum_{i=1}^{m} \max\{0, 1 - y_i \cdot \theta^T x_i + q\} \\ & g_3(x) = \mathcal{I}(\|P^T\theta\|_1 \leq q) \end{aligned}$$

Example: Robust SVM to proximal and epigraph

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

The problem is equivalent to

$$\begin{aligned}
\underset{x=\{\theta,t,p,q\}}{\text{minimize}} \quad & g_1(x) + g_2(x) + g_3(x), \\
\text{subject to} \quad & g_1(x) = \frac{\lambda}{2}\|\theta\|_2^2 \\
& g_2(x) = \sum_{i=1}^{m} \max\{0, t_i\} \qquad \underline{t_i = 1 - y_i \cdot \theta^T x_i + q} \\
& g_3(x) = \mathcal{I}(\|p\|_1 \le q) \qquad \underline{p = P^T\theta}
\end{aligned}$$

Consist of only DCP atoms, epigraph indicators, and linear equalities.

Solving DCP

We proved that any DCP problem can be converted to the form

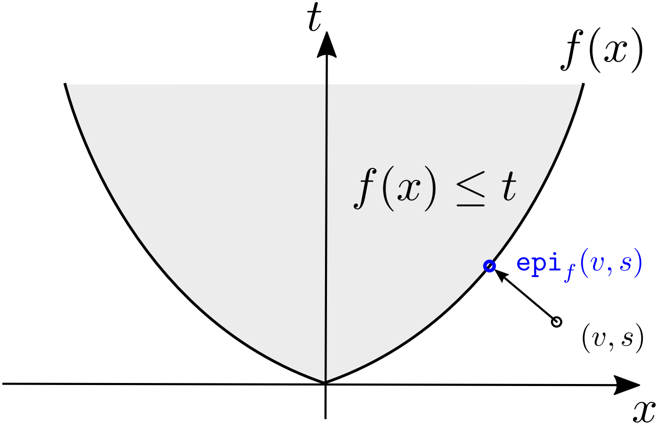
$$\underset{x}{\text{minimize}}\, f(x) \equiv \underset{x}{\text{minimize}} \sum_{i=1}^{n} g_i(x), \;\; \text{subject to } Ax = b,$$

which can be solved by operator splitting (e.g. ADMM, DR), as long as we have proximal operators for the DCP atoms and epigraph indicators

$$\begin{aligned} x_i^{k+1} &\leftarrow \mathrm{prox}_{g_i}(u_i^k - z^k) \\ z^{k+1} &\leftarrow \begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N}(x_i^{k+1} + u^k) \\ b \end{bmatrix} \\ u_i^{k+1} &\leftarrow u_i^k + x_i^{k+1} - z^{k+1} \end{aligned}$$

Epigraph projection

Proximal of epigraph indicator $=$ projection on the epigraph set



$$\underset{x,t}{\text{minimize}} \;\; \|(x,t) - (v,s)\|^2, \;\; \text{subject to} \;\; f(x) \leq t$$

Contribution: Epigraph projection Algorithms

In this work, we design a wide class of epigraph projection algorithms for DCP atoms, for example,

$$f(x) = \sum_i x_i^2$$ by exact method

$$f(x) = \log \sum_i \exp(x_i)$$ by primal-dual Newton method

$$f(x) = -\sum_i \log(x_i)$$ by implicit Newton method

$$f(x) = \sum_i |x_i|$$ by sum of max solver

$$\ldots\ldots$$

Solving Epigraph Projections

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

$$\Updownarrow$$

$$\underset{\lambda \ge 0}{\text{minimize}}\, D(\lambda),$$

in which $D(\lambda)$ can be construct by the proximal operator

A 1D optimization problem, can always be solved via bisection, but is time-consuming (each iteration of bisection requires a new prox operation)

Epigraph Projection: Closed-form method

For example, for the atom

$$f(x) = \|x\|_2^2,$$

the dual epigraph projection problem has a closed-form solution satisfying

$$\frac{dD(\lambda)}{d\lambda} = (\frac{1}{2}\lambda)(1 + 2\lambda)^2 - \|v\|_2^2 = 0$$

This equation can be solved by cubic formula or Kerner method

While many proximal operators have closed-form solution, not many epigraph projections have them

Epigraph Projection: Primal-dual Newton method

When the domain of the atom is unconstrained, e.g.,

$$f(x) = \log \sum_i \exp(x_i),$$

we can exploit the KKT system of the epigraph problem

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

and derive the Newton direction for the system

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

The Hessian $\nabla_x^2 f(x)$ is often structured and can be inversed in $O(n)$

Epigraph Projection: Implicit dual Newton

When the domain of the atom is constrained, e.g.,

$$f(x) = -\sum_i \log(x_i), \quad \Longrightarrow \quad x_i > 0,$$

primal-dual Newton method cannot be applied directly

However, we can write dual function in terms of proximal operator

$$\begin{aligned} D(\lambda) &\equiv \min_{x,t} L(x,t,\lambda) \\ &= L(\mathrm{prox}_{\lambda f}(v), \lambda + s, \lambda) \end{aligned}$$

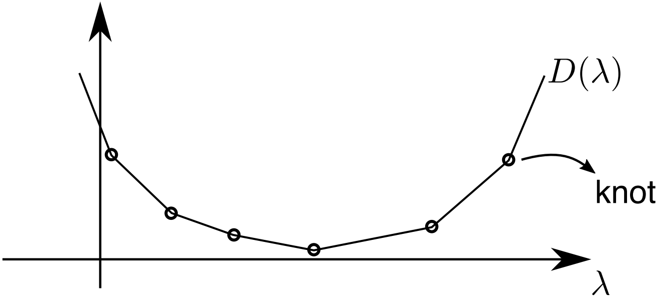
Because $\mathrm{prox}_{\lambda f}(v)$ is a function of λ, we can apply implicit function theorem to obtain $\frac{dD(\lambda)}{d\lambda}$ and $\frac{d^2 D(\lambda)}{d\lambda^2}$

Epigraph Projection: Sum of Max Method

When the atom is piece-wise linear, e.g.,

$$f(x) = \sum_i |x_i|,$$

the dual epigraph problem is also piece-wise linear, with at most $O(n)$ knots



We can enumerate all the knot point and perform quick-select on the gradient direction to find the optimal λ

Contribution Summary

In conclusion,

we present an algorithm that can solve any DCP using only proximal and epigraph projection operators

We designed a wide class of epigraph projection algortihms that enables the proximal method to work in general DCP problems

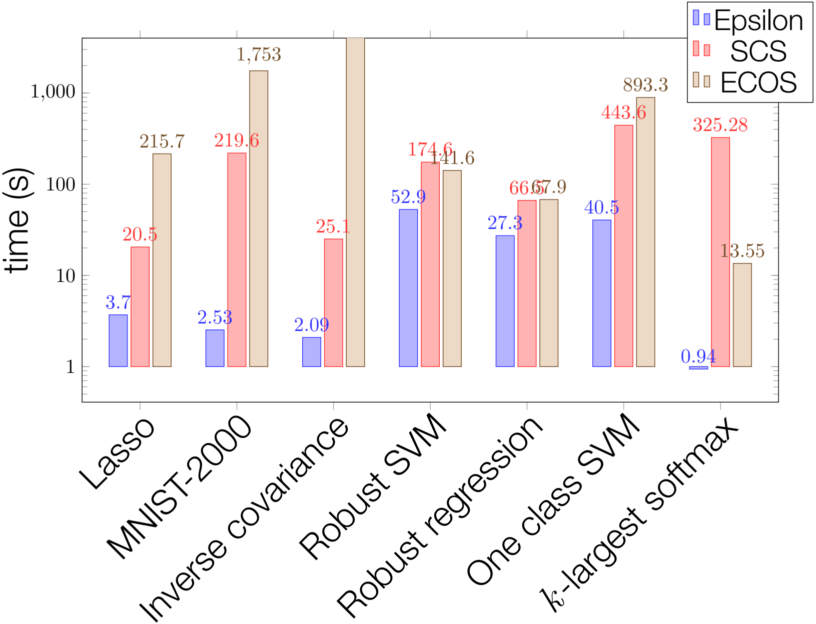
The Epsilon Framework

All the algorithms and experiments are integrated in the

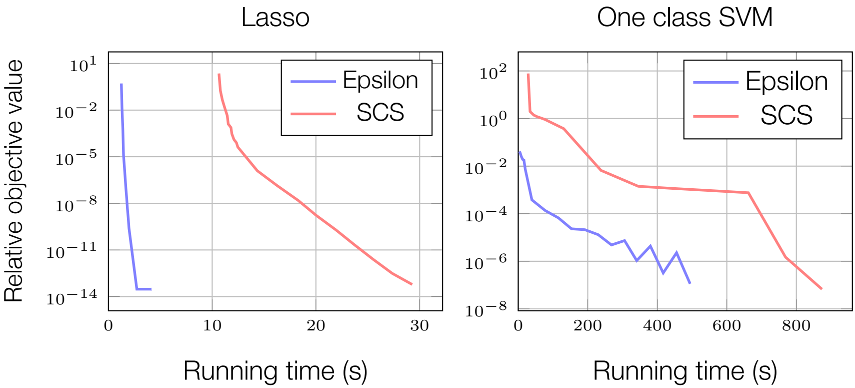
Epsilon framework (Epigraph Proximal Solver),

which is downloadable at `http://epopt.io`

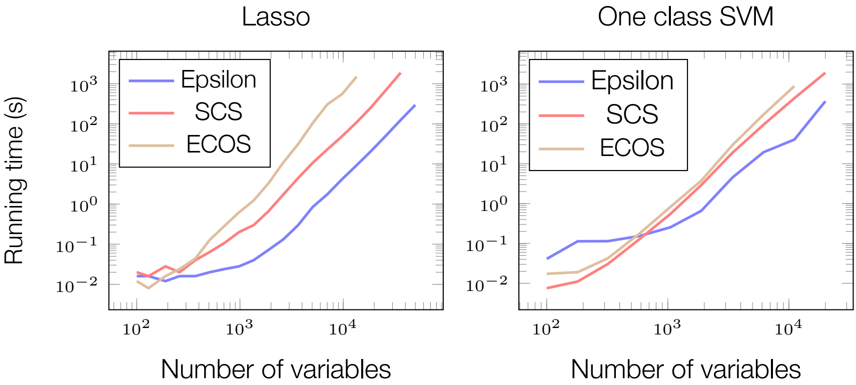
Experiment: time to same objective values



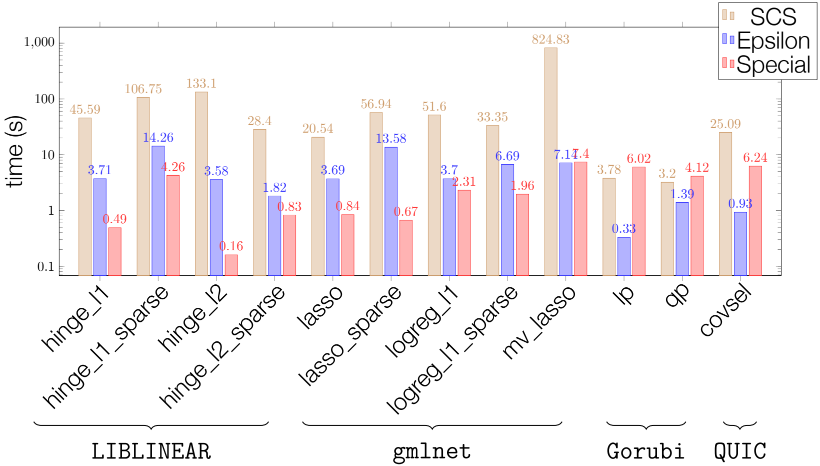
Experiment: Time vs Objective



Experiment: Scaling



Expriment: Compared with Specialized Solvers



Conclusion

Solving `cvx(py)` problems without conic transform

- By proximal operators and epigraph projections

Collection of epigraph projection algorithms

- Covers most DCP atoms

Experiments

- Order of magnitude faster than other solver

Full framework available at `http://epopt.io`