Disciplined Convex Optimization with CVXR

Anqi Fu    Bala Narasimhan    Stephen Boyd
EE & Statistics Departments
Stanford University

useR! Conference 2018
Convex Optimization

CVXR

Examples

Future Work
Outline

Convex Optimization

CVXR

Examples

Future Work
Convex Optimization

minimize \( f_0(x) \)
subject to \( f_i(x) \leq 0, \quad i = 1, \ldots, M \)
\( Ax = b \)

with variable \( x \in \mathbb{R}^n \)

- Objective and inequality constraints \( f_0, \ldots, f_M \) are convex
- Equality constraints are linear
Convex Optimization

minimize \( f_0(x) \)
subject to \( f_i(x) \leq 0, \quad i = 1, \ldots, M \)
\( Ax = b \)

with variable \( x \in \mathbb{R}^n \)

- Objective and inequality constraints \( f_0, \ldots, f_M \) are convex
- Equality constraints are linear

Why?

- We can solve convex optimization problems
- There are many applications in many fields, including machine learning and statistics
Convex Problems in Statistics

- Least squares, nonnegative least squares
- Ridge and lasso regression
- Isotonic regression
- Huber (robust) regression
- Logistic regression
- Support vector machine
- Sparse inverse covariance
- Maximum entropy and related problems
- . . . and new methods being invented every year!
Domain Specific Languages for Convex Optimization

- Special languages/packages for general convex optimization
- CVX, CVXPY, YALMIP, Convex.jl
- Slower than custom code, but extremely flexible and enables fast prototyping
Domain Specific Languages for Convex Optimization

- Special languages/packages for general convex optimization
- CVX, CVXPY, YALMIP, Convex.jl
- Slower than custom code, but extremely flexible and enables fast prototyping

```python
from cvxpy import *
beta = Variable(n)
cost = norm(X * beta - y)
prob = Problem(Minimize(cost))
prob.solve()
beta.value
```
Outline

Convex Optimization

CVXR

Examples

Future Work
CVXR

A modeling language in R for convex optimization

- Connects to many solvers: ECOS, SCS, MOSEK, etc
- Mixes easily with general R code and other libraries
- Uses disciplined convex programming to verify convexity
Ordinary Least Squares (OLS)

- minimize $||X\beta - y||^2_2$
- $\beta \in \mathbb{R}^n$ is variable, $X \in \mathbb{R}^{m \times n}$ and $y \in \mathbb{R}^m$ are constants

Equation:

```r
library(CVXR)
beta <- Variable(n)
obj <- sum_squares(y - X %*% beta)
prob <- Problem(Minimize(obj))
result <- solve(prob)
result$value
result$getValue(beta)
```

Examples
Ordinary Least Squares (OLS)

- minimize $\|X\beta - y\|^2_2$
- $\beta \in \mathbb{R}^n$ is variable, $X \in \mathbb{R}^{m \times n}$ and $y \in \mathbb{R}^m$ are constants

```r
library(CVXR)
beta <- Variable(n)
obj <- sum_squares(y - X %*% beta)
prob <- Problem(Minimize(obj))
result <- solve(prob)
result$value
result$getValue(beta)
```

- $X$ and $y$ are constants; $\beta$, obj, and prob are S4 objects
- `solve` method returns a list that includes optimal $\beta$ and objective value
Non-Negative Least Squares (NNLS)

- minimize $||X\beta - y||^2_2$ subject to $\beta \geq 0$
Non-Negative Least Squares (NNLS)

- minimize $||X\beta - y||_2^2$ subject to $\beta \geq 0$

```r
constr <- list(beta >= 0)
prob2 <- Problem(Minimize(obj), constr)
result2 <- solve(prob2)
result2$value
result2$getValue(beta)
```

- Construct new problem with list `constr` of constraints formed from constants and variables
- Variables, parameters, expressions, and constraints exist outside of any problem
True vs. Estimated Coefficients

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>NNLS</th>
<th>True</th>
<th>OLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td></td>
<td></td>
<td>1.5</td>
</tr>
<tr>
<td>b2</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b3</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b4</td>
<td>-0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b7</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b9</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Examples
True vs. Estimated Coefficients

Examples
Sparse Inverse Covariance Estimation

- Samples $x_i \in \mathbb{R}^n$ drawn i.i.d. from $N(0, \Sigma)$
- Know covariance $\Sigma \in \mathbb{S}^n_+$ has \textit{sparse} inverse $S = \Sigma^{-1}$
Sparse Inverse Covariance Estimation

- Samples $x_i \in \mathbb{R}^n$ drawn i.i.d. from $N(0, \Sigma)$
- Know covariance $\Sigma \in \mathbb{S}_+^n$ has sparse inverse $S = \Sigma^{-1}$
- One way to estimate $S$ is by maximizing the log-likelihood with a sparsity constraint:

$$\begin{align*}
\text{maximize} & \quad \log \det(S) - \text{tr}(SQ) \\
\text{subject to} & \quad S \in \mathbb{S}_+^n, \quad \sum_{i=1}^n \sum_{j=1}^n |S_{ij}| \leq \alpha
\end{align*}$$

- $Q = \frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x})(x_i - \bar{x})^\top$ is sample covariance
- $\alpha \geq 0$ is a parameter controlling the degree of sparsity
Sparse Inverse Covariance Estimation

\[
S \leftarrow \text{Semidef}(n)
\]
\[
\text{obj} \leftarrow \log\_\text{det}(S) - \text{matrix\_trace}(S \circledast Q)
\]
\[
\text{constr} \leftarrow \text{list}(\text{sum}(\text{abs}(S)) \leq \alpha)
\]
\[
\text{prob} \leftarrow \text{Problem}(\text{Maximize(\text{obj}), constr})
\]
\[
\text{result} \leftarrow \text{solve(prob)}
\]
\[
\text{result}\$\text{getValue}(S)
\]

- Semidef restricts variable to positive semidefinite cone
- Must use \(\log\_\text{det}(S)\) instead of \(\log(\text{det}(S))\) since \text{det} is not a supported atom
- \(\text{result}\$\text{getValue}(S)\) returns an R matrix
True vs. Estimated Sparsity of Inverse

True Inverse

Estimate ($\alpha = 1$)
True vs. Estimated Sparsity of Inverse

True Inverse

Estimate ($\alpha = 6$)
True vs. Estimated Sparsity of Inverse

**True Inverse**

**Estimate (\(\alpha = 10\))**
Outline

Convex Optimization

CVXR

Examples

Future Work
Future Work

- Flesh out convex functions in library
- Develop more applications and examples
- Make connecting new solvers easier
- Add warm start support
- Further speed improvements

Official site: cvxr.rbind.io
CRAN page: CRAN.R-project.org/package=CVXR