Outline

First section

Second section
Bulleted list

- XXX
  - XXX
  - XXX
  - XXX

- XXX
  - XXX
  - XXX
  - XXX

- XXX
Pictures with tikz
convex envelope of (nonconvex) \( f \) is the largest convex underestimator \( g \)

\( i.e., \) the best convex lower bound to a function

\( \text{example: } \ell_1 \) is the envelope of \text{card} (on unit \( \ell_\infty \) ball)

\( \text{example: } \| \cdot \|_* \) is the envelope of \text{rank} (on unit spectral norm ball)

\( \text{example: } \) various characterizations: \( e.g., f^{**} \) or convex hull of epigraph
Outline

First section

Second section

Second section
Group lasso

(e.g., Yuan & Lin; Meier, van de Geer, Bühlmann; Jacob, Obozinski, Vert)

▶ problem:

\[
\text{minimize } f(x) + \lambda \sum_{i=1}^{N} \|x_i\|_2
\]

i.e., like lasso, but require groups of variables to be zero or not

▶ also called $\ell_{1,2}$ mixed norm regularization
Structured group lasso

(Jacob, Obozinski, Vert; Bach et al.; Zhao, Rocha, Yu; . . .)

- problem:
  \[
  \text{minimize } f(x) + \sum_{i=1}^{N} \lambda_i \| x_{g_i} \|_2
  \]
  where \( g_i \subseteq [n] \) and \( G = \{g_1, \ldots, g_N\} \)

- like group lasso, but the groups can overlap arbitrarily

- particular choices of groups can impose ‘structured’ sparsity

- e.g., topic models, selecting interaction terms for (graphical) models, tree structure of gene networks, fMRI data

- generalizes to the composite absolute penalties family:
  \[
  r(x) = \| (\| x_{g_1} \|_{p_1}, \ldots, \| x_{g_N} \|_{p_N}) \|_{p_0}
  \]
Structured group lasso
(Jacob, Obozinski, Vert; Bach et al.; Zhao, Rocha, Yu; ...)

Hierarchical selection:

\[ G = \{\{4\}, \{5\}, \{6\}, \{2, 4\}, \{3, 5, 6\}, \{1, 2, 3, 4, 5, 6\}\} \]

- nonzero variables form a rooted and connected subtree
  - if node is selected, so are its ancestors
  - if node is not selected, neither are its descendants
Sample ADMM implementation: lasso

```matlab
prox_f = @(v,rho) (rho/(1 + rho))*(v - b) + b;
prox_g = @(v,rho) (max(0, v - 1/rho) - max(0, -v - 1/rho));

AA = A*A';
L = chol(eye(m) + AA);

for iter = 1:MAX_ITER
    xx = prox_g(xz - xt, rho);
yx = prox_f(yz - yt, rho);

    yz = L \ (L' \ (A*(xx + xt) + AA*(yx + yt)));
xz = xx + xt + A'*(yx + yt - yz);

    xt = xt + xx - xz;
yt = yt + yx - yz;
end
```

Second section
Figure

The figure shows the comparison of different cases and realizations of a function over iterations. The y-axis represents the difference between the best function value $f_{\text{best}}(k)$ and the optimal function value $f^*$, scaled on a logarithmic scale. The x-axis represents the iteration count $k$. The graph includes:

- The noise-free case, represented by a dotted line.
- Realization 1, represented by a dashed blue line.
- Realization 2, represented by a dashed red line.

This visualization helps in understanding the convergence and performance of the algorithms under different conditions.
Algorithm

if $L$ is not known (usually the case), can use the following line search:

given $x^k$, $\lambda^{k-1}$, and parameter $\beta \in (0, 1)$.
Let $\lambda := \lambda^{k-1}$.

repeat
1. Let $z := \text{prox}_{\lambda g}(x^k - \lambda \nabla f(x^k))$.
2. break if $f(z) \leq \hat{f}_\lambda(z, x^k)$.
3. Update $\lambda := \beta \lambda$.

return $\lambda^k := \lambda$, $x^{k+1} := z$.

typical value of $\beta$ is $1/2$, and

$$\hat{f}_\lambda(x, y) = f(y) + \nabla f(y)^T(x - y) + (1/2\lambda)\|x - y\|_2^2$$