Projected Gradient Descent Efficiently Solves the Trust Region Problem

Mark Nishimura, Reese Pathak
EE364b: Convex Optimization II Class Project

Introduction

Trust region methods are sequential programming procedures which formulate and solve many instances of the following **trust region problem**

$$\begin{array}{ll} \text{minimize} & (1/2)x^TAx + b^Tx \\ \text{subject to} & \|x\| \le R \end{array} \tag{1}$$

with variable x. The matrix A may be indefinite, in which case the resulting problem is not convex.

Projected Gradient Descent

We investigate the behavior of **projected gradient descent** (PGD) which consists of iterating the following two steps:

$$\begin{aligned} y^{(k+1)} &= x^{(k)} - \eta \nabla f(x^{(k)}) \\ x^{(k+1)} &= \Pi_{\mathcal{B}(R)}(y^{(k+1)}). \end{aligned} \tag{2}$$

Structured group lasso

Another approach is the *structured group lasso*:

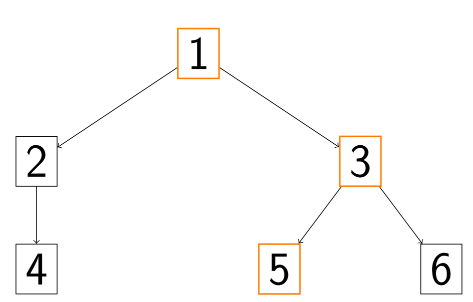
$$\text{minimize} \;\; f(x) + \textstyle\sum_{i=1}^N \lambda_i \|x_{g_i}\|_2$$

where $g_i \subseteq [n]$ and $\mathcal{G} = \{g_1, \ldots, g_N\}$

- 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**:

$$r(x) = \|(\|x_{g_1}\|_{p_1}, \ldots, \|x_{g_N}\|_{p_N})\|_{p_0}$$

Hierarchical selection



- $\mathcal{G} = \{\{4\}, \{5\}, \{6\}, \{2,4\}, \{3,5,6\}, \{1,2,3,4,5,6\}\}$
- 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

Algorithm

We solve this problem using an ADMM lasso implementation:

```
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
```

Line search

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

---

given x^k, λ^{k-1}, and parameter $\beta \in (0,1)$.
Let $\lambda := \lambda^{k-1}$.
repeat
1. Let $z := \mathbf{prox}_{\lambda g}(x^k - \lambda \nabla f(x^k))$.
2. **break if** $f(z) \le \hat{f}_\lambda(z, x^k)$.
3. Update $\lambda := \beta\lambda$.

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

---

typical value of β is $1/2$, and

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

Convergence proof

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Numerical example

Consider a numerical example with $f(x) = \|Ax - b\|_2^2$ with $A \in \mathbf{R}^{10\times 100}$ and $b \in \mathbf{R}^{10}$. Entries of A and b are generated as independent samples from a standard normal distribution. Here, we have chosen λ using cross validation.

Results

On this numerical example, the ADMM method converges quickly. We give two realizations corresponding to different parameters A and b.

Conclusion

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Acknowledgements

This material is based upon work supported by the X Fellowship.