Sparse and Regularized Optimization

In many applications, we seek not an exact minimizer of the underlying objective, but rather an approximate minimizer that satisfies certain desirable properties:

- sparsity (few nonzeros);
- specific nonzero patterns (e.g. tree structure);
- low-rank (if a matrix);
- low "total-variation";
- generalizability. (Vapnik: "...tradeoff between the quality of the approximation of the given data and the complexity of the approximating function.")

A common way to obtain structured solutions is to modify the objective f by adding a regularizer $\tau\psi(x)$, for some parameter $\tau > 0$.

$$\min_x f(x) + \tau\psi(x),$$

where ψ induces the desired structure in x.

Applications I

LASSO for variable selection. Originally stated as

$$\min_x \frac{1}{2}\|Ax-b\|_2^2 \text{ such that } \|x\|_1 \le T,$$

for parameter $T > 0$. Equivalent to an "ℓ_2-ℓ_1" formulation:

$$\min_x \frac{1}{2}\|Ax-b\|_2^2 + \tau\|x\|_1, \qquad \text{for some } \tau > 0.$$

Group LASSO to select disjoint groups of variables:

$$\min_x \frac{1}{2}\|Ax-b\|_2^2 + \sum_{g \in G} \|x_{[g]}\|_2,$$

with each $[g]$ a subset of indices $\{1, 2, \ldots, n\}$.

- Easy to shrink with disjoint groups.
- Still easy when $\|\cdot\|_2$ is replaced by $\|\cdot\|_\infty$.

Applications II

Overlapping Groups. There are sometimes complex relationships between the variables, and we want the set of variables selected (the nonzeros) to respect these relationships. Can sometimes design groups regularizers that induce this structure. Examples:

- Each group is the set of descendants of a node in a directed graph;
- When coefficients form a tree (e.g. wavelet representations), each group could be (a) the set of ancestors of a node; (b) parent-child pairs; (c) all subtrees.

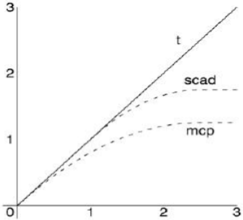
There's much recent work on shrink-based algorithms by F. Bach and the WILLOW and SIERRA project teams.

- For hierarchical groups ($[g] \cap [h] \neq \emptyset$ implies either $[g] \subset [h]$ or $[h] \subset [g]$), shrink operator can be computed efficiently by ordering groups appropriately.
- For non-hierarchical groups with $\| \cdot \|_\infty$, use a network flow technique to shrink (Mairal et al, NIPS, 2010).

Applications III

Nonconvex Regularizers. Nonconvex element-wise penalties have become popular for variable selection in statistics.

- SCAD (smoothed clipped absolute deviation) (Fan and Li, 2001)
- MCP (Zhang, 2010).



Properties: unbiased estimates, sparse estimates, solution path continuous in regulariation parameter τ.

Code: SparseNet (Mazumder, Friedman, Hastie, 2011): coordinate desc.

Applications IV

Compressed Sensing. Sparse signal recovery from noisy measurements. Given matrix A (with more columns than rows) and observation vector y, seek a sparse x (i.e. few nonzeros) such that $Ax \approx y$. Solve

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \tau\|x\|_1.$$

- Under "restricted isometry" properties on A ("tall" column submatrices are nearly orthonormal), $\|x\|_1$ is a good surrogate for $\text{card}(x)$.
- Assume that A is not stored explicitly, but matrix-vector multiplications are available. Hence can compute f and ∇f.

Applications V: Classification and Regression

Support Vector Machines. See above. Can use $\|w\|_1$ in the linear SVM, to get a sparse weight vector.

ℓ_1-**Regularized Logistic Regression.** See above.

Applications VI: Matrix Completion

Seek a matrix $X \in \mathbb{R}^{m \times n}$ with low rank that matches certain observations, possibly noisy.

$$\min_X \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2 + \tau\psi(X),$$

where $\mathcal{A}(X)$ is a linear mapping of the components of X (e.g. element-wise observations).

Can have ψ as the nuclear norm — see discussion above for solution of subproblems via SVD.

Alternatively: X is the sum of sparse matrix and a low-rank matrix. The element-wise 1-norm $\|X\|_1$ is useful in inducing sparsity.

Basics of Shrinking

Regularizer ψ is often nonsmooth but "simple." Often the problem is easy to solve when f is replaced by a quadratic with diagonal Hessian:

$$\min_z g^T(z-x) + \frac{1}{2\alpha}\|z-x\|_2^2 + \tau\psi(z).$$

Equivalently,

$$\min_z \frac{1}{2\alpha}\|z-(x-\alpha g)\|_2^2 + \tau\psi(z).$$

Define the shrink operator as the arg min:

$$S_\tau(y,\alpha) := \arg\min_z \frac{1}{2\alpha}\|z-y\|_2^2 + \tau\psi(z).$$

Typical algorithm:

$$x_{k+1} = S_\tau(x_k - \alpha_k g_k, \alpha_k),$$

with for example $g_k = \nabla f(x_k)$.

Interesting Regularizers and their Shrinks: I

Cases for which the subproblem is simple:

- $\psi(z) = \|z\|_1$. Thus $S_\tau(y, \alpha) = \text{sign}(y) \max(|y| - \alpha\tau, 0)$. When y complex, have

$$S_\tau(y, \alpha) = \frac{\max(|y| - \tau\alpha, 0)}{\max(|y| - \tau\alpha, 0) + \tau\alpha} y.$$

- $\psi(z) = \sum_{g \in G} \|z_{[g]}\|_2$ or $\psi(z) = \sum_{g \in G} \|z_{[g]}\|_\infty$, where $z_{[g]}$, $g \in G$ are non-overlapping subvectors of z. Here

$$S_\tau(y, \alpha)_{[g]} = \frac{\max(|y_{[g]}| - \tau\alpha, 0)}{\max(|y_{[g]}| - \tau\alpha, 0) + \tau\alpha} y_{[g]}.$$

- $\psi(x) = I_\Omega(x)$: Indicator function for a closed convex set Ω. Then $S_\tau(y, \alpha)$ is the projection of y onto Ω.

Interesting Regularizers and their Shrinks: II

- Z is a matrix and $\psi(Z) = \|Z\|_*$ is the nuclear norm of Z: the sum of singular values. Threshold operator is

$$S_\tau(Y, \alpha) := \arg\min_Z \frac{1}{2\alpha} \|Z - Y\|_F^2 + \tau \|Z\|_*$$

with solution obtained from the SVD $Y = U \Sigma V^T$ with U, V orthonormal and $\Sigma = \text{diag}(\sigma_i)_{i=1,2,\dots,m}$. Setting $\tilde{\Sigma} = \text{diag}(\max(\sigma_i - \tau\alpha, 0)_{i=1,2,\dots,m})$, the solution is

$$S_\tau(Y, \alpha) = U \tilde{\Sigma} V^T.$$

Basic Prox-Linear Algorithm

(Fukushima and Mine, 1981) for solving $\min_x f(x) + \tau\psi(x)$.

0: Choose x_0

k: Choose $\alpha_k > 0$ and set

$$\begin{aligned} x_{k+1} &= S_\tau(x_k - \alpha_k \nabla f(x_k); \alpha_k) \\ &= \arg\min_z \nabla f(x_k)^T(z - x_k) + \frac{1}{2\alpha_k}\|z - x_k\|_2^2 + \tau\psi(z). \end{aligned}$$

This approach goes by many names, including "forward-backward splitting," "shrinking / thresholding."

Straightforward, but can be fast when the regularization is strong (i.e. solution is "highly constrained") and the reduced problem is well conditioned.

Can show convergence for steps $\alpha_k \in (0, 2/L)$, where L is the bound on $\nabla^2 f$. (Like a short-step gradient method.)

Enhancements

Alternatively, since α_k plays the role of a steplength, can adjust it to get better performance *and* guaranteed convergence.

- "Backtracking:" decrease α_k until sufficient decrease condition holds.
- Use Barzilai-Borwein strategies to get nonmonotonic methods. By enforcing sufficient decrease every 10 iterations (say), still get global convergence.

The approach can be accelerated using optimal gradient techniques. See earlier discussion of **FISTA**, where we solve the shrinking problem with $\alpha_k = 1/L$ in place of a step along $-\nabla f$ with this steplength.

Note that these methods reduce ultimately to gradient methods on a reduced space: the optimal manifold defined by the regularizer ψ. Acceleration or higher-order information can help improve performance.

Continuation in τ

Performance of basic shrinking methods is quite sensitive to τ.

Typically higher $\tau \Rightarrow$ stronger regularization $\Rightarrow$ optimal manifold has lower dimension. Hence, it's easier to identify the optimal manifold, and basic shrinking methods can sometimes do so quickly.

For smaller τ, a simple "continuation" strategy can help:

```
0: Given target value $\tau_f$, choose initial $\tau_0 > \tau_f$, starting point $\bar{x}$ and
   factor $\sigma \in (0, 1)$.
k: Find approx solution $x(\tau_k)$ of $\min_x f(x) + \tau\psi(x)$, starting from $\bar{x}$;
   **if** $\tau_k = \tau_f$ **then STOP**;
   Set $\tau_{k+1} \leftarrow \max(\tau_f, \sigma\tau_k)$ and $\bar{x} \leftarrow x(\tau_k)$;
```

Recent report by Xiao and Zhang (2012) analyzes this strategy.

(Solution $x(\tau)$ is often desired on a range of τ values anyway.)

Stochastic Gradient + Regularization: Dual Averaging

Solve the regularized problem, but have only *estimates* of $\nabla f(x_k)$.

We can combine dual averaging, stochastic gradient, and shrinking: see Xiao (2010) who extends Nesterov (2009).

$$\min_x \phi_\tau(x) := E_\xi f(x;\xi) + \tau\psi(x)$$

At iteration k choose ξ_k randomly and i.i.d from the ξ distribution, and choose $g_k \in \partial f(x_k;\xi_k)$. Use these to define the averaged subgradient $\bar{g}_k = \sum_{i=1}^k g_i/(k+1)$, and solve the subproblem

$$x_{k+1} = \arg\min_x \bar{g}_k^T x + \tau\psi(x) + \frac{\gamma}{\sqrt{k}}\|x - x_0\|^2.$$

Same as earlier, but with regularizer ψ included explicitly.

Can prove convergence results for averaged iterates $\bar{x}_k$: roughly

$$E\phi_\tau(\bar{x}_k) - \phi_\tau^* \leq \frac{C}{\sqrt{k}},$$

where expectation is over the random number stream $\xi_0, \xi_1, \dots, \xi_{k-1}$.

Stochastic Gradient + Regularization: FOBOS

An obvious extension of the prox-linear approach to the stochastic gradient setting: replace ∇f by an approximation e.g. $g_k \in \partial f(x_k; \xi_k)$:

$$x_{k+1} = S_\tau(x_k - \alpha_k g_k; \alpha_k).$$

(Duchi and Singer, 2009, p.9-10).

COMID: Generalization to mirror descent.

Also SSG (Lin et al, 2011), and Truncated Gradient (Langford et al, 2009).

Identifying Optimal Manifolds

Identification of the manifold of the regularizer ψ on which x^* lies can improve algorithm performance, by focusing attention on a reduced space. We can thus evaluate *partial* gradients and Hessians, restricted to just this space.

For nonsmooth regularizer ψ, the optimal manifold is a smooth surface passing through x^* along which the restriction of ψ is smooth.

Example: for $\psi(x) = \|x\|_1$, have manifold consisting of z with

$$z_i \begin{cases} \geq 0 & \text{if } x_i^* > 0 \\ \leq 0 & \text{if } x_i^* < 0 \\ = 0 & \text{if } x_i^* = 0. \end{cases}$$

If we know the optimal nonzero components, we know the manifold. We could restrict the search to just this set of nonzeros.

Identification in Stochastic Gradient / Dual Averaging

In the stochastic setting, dual averaging has identification properties most like shrink algorithms with full gradient.

In the strongly convex setting, as the non-averaged iterates x_k (mostly) converge to x^*, the gradient estimate $\bar{g}_k$ (mostly) converges to $\nabla f(x^*)$. Eventually, the subsequence of iterates that lie on the optimal manifold becomes dense. (Lee and Wright, 2012)

This property is not shared by algorithms that

- average the primal iterates;
- use only the latest gradient estimate g_k in the step computation (that is, they *don't* average the dual iterates.

In particular, under ℓ_1 regularization, algorithms with these features don't usually produce sparse solutions.

Algorithmic implication: Once the manifold settles down, can switch to a different algorithm, better suited to a small space.

Further Reading

1. *Optimization for Machine Learning*, edited volume with 18 Chapters, MIT Press, NIPS Workshop Series, 2011.
2. F. Bach, R. Jenatton, J. Mairal, G. Obozinski, “Convex optimization with sparsity-inducing norms,” in [1], 2011.
3. M. Fukushima and H. Mine. “A generalized proximal point algorithm for certain non-convex minimization problems.” International Journal of Systems Science, 12, pp. 989–1000, 1981.
4. P. L. Combettes and V. R. Wajs. “Signal recovery by proximal forward-backward splitting.” Multiscale Modeling and Simulation, 4, pp. 1168–1200, 2005.
5. S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. “Sparse reconstruction by separable approximation.” IEEE Transactions on Signal Processing, 57, pp. 2479–2493, 2009.
6. L. Xiao. “Dual averaging methods for regularized stochastic learning and online optimization.” Journal of Machine Learning Research, 11, pp 2543–2596, 2010.
7. L. Xiao and T. Zhang, “A Proximal-Gradient Homotopy methods for the Sparse Least-Squares Problem,” Technical Report, March 2012. arXiv:1203:2003v1.
8. A. Lewis and S. Wright, “A Proximal Method for Composite Minimization,” 2008.
9. S. Lee and S. J. Wright, “Manifold identification for dual averaging in regularized stochastic online learning,” JMLR, 2012.

V. Decomposition / Coordinate Relaxation

For $\min\ f(x)$, at iteration k, choose a subset $\mathcal{G}_k \subset \{1, 2, \ldots, n\}$ and take a step d_k only in these components. i.e. fix $d_k(i) = 0$ for $i \notin \mathcal{G}_k$.

Gives more manageable subproblem sizes, in practice.

Can

- take a reduced gradient step in the $\mathcal{G}_k$ components;
- take multiple "inner iterations"
- actually solve the reduced subproblem in the space defined by $\mathcal{G}_k$.

Constraints and Regularizers Complicate Things

For $\min_{x \in \Omega} f(x)$, need to put enough components into $\mathcal{G}_k$ to stay feasible, as well as make progress.

Example: $\min f(x_1, x_2)$ with $x_1 + x_2 = 1$. Relaxation with $\mathcal{G}_k = \{1\}$ or $\mathcal{G}_k = \{2\}$ won't work.

For separable regularizer (e.g. Group LASSO) with

$$\psi(x) = \sum_{i \in G} \psi_i(x_{[i]}),$$

need to ensure that $\mathcal{G}_k$ is a union of the some index subsets $[g]$. i.e. the relaxation components must be consonant with the partitioning.

Decomposition and Dual SVM

Decomposition has long been popular for solving the dual (QP) formulation of SVM, since the number of variables (= number of training examples) is sometimes very large.

SMO: Each $\mathcal{G}_k$ has two components.

LIBSVM: SMO approach (still $|\mathcal{G}_k| = 2$), with different heuristic for choosing $\mathcal{G}_k$.

LASVM: Again $|\mathcal{G}_k| = 2$, with focus on online setting.

SVM-light: Small $|\mathcal{G}_k|$ (default 10).

GPDT: Larger $|\mathcal{G}_k|$ (default 400) with gradient projection solver as inner loop.

Choice of $\mathcal{G}_k$ and Convergence Results

Some methods (e.g. Tseng and Yun, 2010) require $\mathcal{G}_k$ to be chosen so that *the improvement in subproblem objective obtained over the subset* $\mathcal{G}_k$ *is at least a fixed fraction of the improvement available over the whole space*. Undesirable, since to check it, usually need to evaluate the **full gradient** $\nabla f(x_k)$.

Alternative is a *generalized Gauss-Seidel* requirement, where each coordinate is "touched" at least once every T iterations:

$$\mathcal{G}_k \cup \mathcal{G}_{k+1} \cup \ldots \cup \mathcal{G}_{k+T-1} = \{1, 2, \ldots, n\}.$$

Can show global convergence (e.g. Tseng and Yun, 2009; Wright, 2010).

There are also results on

- global linear convergence rates
- optimal manifold identification
- fast local convergence for an algorithm that takes reduced steps on the estimated optimal manifold.

All are *deterministic* analyses.

Naive Stochastic Coordinate Descent

Analysis tools of stochastic gradient may be useful.

for $\min_x f(x)$, take steps of the form $x_{k+1} = x_k - \alpha_k g_k$, where

$$g_k(i) = \begin{cases} [\nabla f(x_k)]_i & \text{if } i \in \mathcal{G}_k \\ 0 & \text{otherwise,} \end{cases}$$

With suitable random selection of $\mathcal{G}_k$ can ensure that g_k (appropriately scaled) is an unbiased estimate of $\nabla f(x_k)$. Hence can apply SGD techniqes discussed earlier, to choose α_k and obtain convergence.

Stochastic Coordinate Descent

(Richtarik and Takac, 2012; see also Nesterov, 2012)

$$\min_{x} \phi(x) := f(x) + \psi(x).$$

Describe an approach that

- Partitions the components of x into subvectors, consonant with the regularizer ψ;
- Makes a random selection of one partition to update at each iteration;
- Exploits knowledge of the partial Lipschitz constant for each partition in choosing the step.
- Allows parallel implementation.

Essentially, picks one partition and does the basic short-step, prox-linear method for that component, shrunk with the regularizer for that component.

Richtarik and Takac call it RCDC: Randomized Coordinate Descent for Composite Functions.

RCDC Details

Partition components $\{1, 2, \ldots, n\}$ into m blocks with block $[i]$ with corresponding columns from the $n \times n$ identity matrix denoted by U_i.

Denote by L_i the partial Lipschitz constant on partition $[i]$:

$$\|\nabla_{[i]} f(x + U_i t) - \nabla_{[i]} f(x)\| \le L_i \|t\|.$$

Separate the regularizer ψ as above:

$$\psi(x) = \sum_{i=1}^{m} \psi_i(x_{[i]}).$$

For overall structure, define a weighted norm, for weights $w_1, w_2, \ldots, w_m$:

$$\|x\|_W := \left(\sum_{i=1}^{m} w_i \|x_{[i]}\|^2 \right)^{1/2}$$

Weighted measure of level set size:

$$\mathcal{R}_W(x) := \max_{y} \max_{x^* \in X^*} \{ \|y - x^*\|_W \, : \, \phi(y) \le \phi(x) \}.$$

RCDC Algorithm

The key subproblem, for a given partition i:

$$P(x,i): \quad \min_d \; d^T \nabla_{[i]} f(x) + \frac{L_i}{2}\|d\|^2 + \psi_i(x_{[i]} + d).$$

Fix probabilities of choosing each partition: p_i, $i = 1, 2, \dots, m$.

Iteration k:

- Choose partition $i_k \in \{1, 2, \dots, m\}$ with probability p_i;
- Solve $P(x_k, i_k)$ to obtain step $d_{k,i}$;
- Set $x_{k+1} = x_k + U_{i_k} d_{k,i}$.

For convex f and ψ, and uniform weights $p_i = 1/m$, can prove high probability convergence of f to within a specified threshold of $f(x^*)$ in $1/k$ iterations.

RCDC Analysis

The basic analysis requires three steps.

I. Given random varaible sequence $\xi_0, \xi_1, \xi_2, \cdots$ with

$$E[\xi_{k+1} \,|\, \xi_k] \le \left(1 - \frac{1}{c_2}\right) \xi_k, \;\; \text{whenever } \xi_k \ge \epsilon,$$

where $\epsilon > 0$ is a specified threshold and $c_2 \in (0,1)$ is some constant, we have for $k \ge K$ with

$$K := c_2 \log \frac{\xi_0}{\epsilon \rho},$$

that $P(\xi_k \le \epsilon) \ge 1 - \rho$.

II. Expected improvement in ϕ at step k is the average of the m possible partition-wise improvements. For each partition, because of the short-step strategy, the actual improvement is at least equal to the improvement predicted by the subproblem $P(x_k, i)$.

RCDC Analysis, continued

$$
\begin{aligned}
E[\phi(x_{k+1}) \,|\, x_k] &\le \phi(x_k) + \frac{1}{m} \sum_{i=1}^{m} \Big[d_{k,i}^T \nabla_{[i]} f(x_k) + \frac{L_i}{2} \|d_{k,i}\|^2 \\
&\qquad + \psi_{[i]}((x_k)_{[i]} + d_{k,i}) - \psi_i((x_k)_{[i]}) \Big] \\
&= \phi(x_k) + J(x_k).
\end{aligned}
$$

(Each term in the sum $J(x_k)$ is the improvement predicted if partition i is selected at iteration k.)

III. The predicted optimality gap after step k is a substantial improvement over the gap $\phi(x_k) - \phi^*$ at iteration k.

$$
\phi(x) + J(x) - \phi^* \le \max\left(\frac{1}{2}, 1 - \frac{\phi(x) - \phi^*}{2\|x - x^*\|_L^2} \right) (\phi(x) - \phi^*).
$$

RCDC Analysis, continued

Put these pieces together, defining

$$\xi_k := \phi(x_k) - \phi^*,$$

and assuming that the defined threshold ϵ has

$$\epsilon < \min(\mathcal{R}_L^2(x_0), \phi(x_0) - \phi^*)$$

and defining

$$K := \frac{2n\mathcal{R}_L^2(x_0)}{\epsilon} \log \frac{\phi(x_0) - \phi^*}{\epsilon\rho},$$

we have for $k \geq K$ that

$$P(\phi(x_k) - \phi^* \leq \epsilon) \geq 1 - \rho.$$

RCDC, Strongly Convex Case

If ϕ is strongly convex with respect to the weighted norm $\|\cdot\|_L$, we have expected convergence at a linear rate. Require

$$\phi(x) \geq \phi(y) + (x-y)^T \nabla\phi(y) + \frac{\mu}{2}\|x-y\|_L^2,$$

where $\nabla\phi$ denotes any subgradient of ϕ. Defining

$$\gamma_\mu := \begin{cases} 1-\mu/4 & \text{if } \mu \leq 2 \\ 1/\mu & \text{otherwise} \end{cases},$$

we have

$$E[\phi(x_k) - \phi^*] \leq \left(1 - \frac{1-\gamma_\mu}{n}\right)^k (\phi(x_0) - \phi^*).$$

Further Reading

1. P. Tseng and S. Yun, "A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training." Computational Optimization and Applications, 47, pp. 179–206, 2010.
2. P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization." Mathematical Programming, Series B, 117. pp. 387–423, 2009.
3. P. Richtarik and M. Takac, "Iteration complexity of randomized block-coordinate gradient descent methods for minimizing a composite function," Technical Report, Revised, July 2012.
4. S. J. Wright, "Accelerated block-coordinate relaxation for regularized optimization." SIAM J. Optimization 22 (2012), pp. 159–186.
5. Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems." SIAM J. Optimization 22 (2012), pp. 341–362.

Augmented Lagrangian Methods and Splitting

Consider linearly constrained problem:

$$\min \ f(x) \ \text{ s.t. } \ Ax = b.$$

Augmented Lagrangian is

$$\mathcal{L}(x, \lambda; \rho) := f(x) + \lambda^T (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2,$$

where $\rho > 0$. Basic augmented Lagrangian / method of multipliers is

$$\begin{aligned} x_k &= \arg\min_x \mathcal{L}(x, \lambda_{k-1}; \rho_k); \\ \lambda_k &= \lambda_{k-1} + \rho_k (Ax_k - b); \\ &\text{(choose } \rho_{k+1}\text{).} \end{aligned}$$

Extends in a fairly straightforward way to inequality constraints, nonlinear constraints.

Quick History of Augmented Lagrangian

- Dates from 1969: Hestenes, Powell.
- Developments in 1970s, early 1980s by Rockafellar, Bertsekas, and others.
- Lancelot code for nonlinear programming: Conn, Gould, Toint, around 1990.
- Largely lost favor as an approach for general nonlinear programming during the next 15 years.
- Recent revival in the context of sparse optimization and its many applications, in conjunction with splitting / coordinate descent.

Separable Objectives: ADMM

Alternating Directions Method of Multipliers (ADMM) arises when the objective in the basic linearly constrained problem is separable:

$$\min_{(x,z)} f(x) + h(z) \text{ subject to } Ax + Bz = c,$$

for which

$$\mathcal{L}(x, z, \lambda; \rho) := f(x) + h(z) + \lambda^T (Ax + Bz - c) + \frac{\rho}{2} \|Ax - Bz - c\|_2^2.$$

Standard augmented Lagrangian would minimize $\mathcal{L}(x, z, \lambda; \rho)$ over (x, z) jointly — but these are coupled through the quadratic term, so the advantage of separability is lost.

Instead, minimize over x and z separately and sequentially:

$$\begin{aligned} x_k &= \arg\min_x \mathcal{L}(x, z_{k-1}, \lambda_{k-1}; \rho_k); \\ z_k &= \arg\min_z \mathcal{L}(x_k, z, \lambda_{k-1}; \rho_k); \\ \lambda_k &= \lambda_{k-1} + \rho_k (Ax_k + Bz_k - c). \end{aligned}$$

ADMM

- Basically, does a round of block-coordinate descent in (x, z).
- The minimizations over x and z add only a quadratic term to f and h, respectictly. This does not alter the cost much.
- Can perform these minimizations inexactly.
- Convergence is often slow, but sufficient for many applications.
- Many recent applications to compressed sensing, image processing, matrix completion, sparse principal components analysis.

ADMM has a rich collection of antecendents. For an excellent recent survey, including a diverse collection of machine learning applications, see (Boyd et al, 2011).

ADMM for Consensus Optimization

Given

$$\min \sum_{i=1}^{m} f_i(x),$$

form m copies of the x, with the original x as a "master" variable:

$$\min_{x,x^1,x^2,\dots,x^m} \sum_{i=1}^{m} f_i(x^i) \text{ subject to } x^i = x, \; i = 1, 2, \dots, m.$$

Apply ADMM, with $z = (x^1, x^2, \dots, x^m)$, get

$$x_k^i = \arg\min_{x^i} f_i(x^i) + (\lambda_{k-1}^i)^T (x^i - x_{k-1}) + \frac{\rho_k}{2} \|x^i - x_{k-1}\|_2^2, \; \forall_i,$$

$$x_k = \frac{1}{m} \sum_{i=1}^{m} \left(x_k^i + \frac{1}{\rho_k} \lambda_{k-1}^i \right),$$

$$\lambda_k^i = \lambda_{k-1}^i + \rho_k (x_k^i - x_k), \; \forall_i$$

Obvious parallel possibilities in the x^i updates. Synchronize for x update.

ADMM for Awkward Intersections

The feasible set is sometimes an intersection of two or more convex sets that are easy to handle separately (e.g. projections are easily computable), but whose intersection is more difficult to work with.

Example: Optimization over the cone of doubly nonnegative matrices:

$$\min_X f(X) \ \text{s.t.}\ X \succeq 0, \ \ X \geq 0.$$

General form:

$$\min \ f(x) \ \text{s.t.}\ x \in \Omega_i, \ \ i = 1, 2, \dots, m$$

Just consensus optimization, with indicator functions for the sets.

$$\begin{aligned}
x_k &= \arg\min_x \ f(x) + \sum_{i=1}^m (\lambda^i_{k-1})^T (x - x^i_{k-1}) + \frac{\rho_k}{2} \|x - x^i_{k-1}\|_2^2, \\
x^i_k &= \arg\min_{x_i \in \Omega_i} (\lambda^i_{k-1})^T (x_k - x^i) + \frac{\rho_k}{2} \|x_k - x^i\|_2^2, \ \ \forall_i \\
\lambda^i_k &= \lambda^i_{k-1} + \rho_k (x_k - x^i_k), \ \ \forall_i.
\end{aligned}$$

ADMM and Prox-Linear

Given

$$\min_x \; f(x) + \tau\psi(x),$$

reformulate as the equality constrained problem:

$$\min_{x,z} \; f(x) + \tau\psi(z) \;\text{ subject to }\; x = z.$$

ADMM form:

$$\begin{aligned}
x_k &:= \min_x \; f(x) + \tau\psi(z_{k-1}) + (\lambda_k)^T(x - z_{k-1}) + \frac{\mu_k}{2}\|z_{k-1} - x\|_2^2, \\
z_k &:= \min_z \; f(x_k) + \tau\psi(z) + (\lambda_k)^T(x_k - z) + \frac{\mu_k}{2}\|z - x_k\|_2^2, \\
\lambda_{k+1} &:= \lambda_k + \mu_k(x_k - z_k).
\end{aligned}$$

- Minimization over z is the shrink operator — often inexpensive.
- Minimization over x can be performed approximately using an algorithm suited to the form of f.

Further Reading

- S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction methods of multipliers," Foundations and Trends in Machine Learning, 3, pp. 1-122, 2011.
- S. Boyd, "Alternating Direction Method of Multipliers," Talk at NIPS Workshop on Optimization and Machine Learning, December 2011: `videolectures.net/nipsworkshops2011_boyd_multipliers/`
- J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," Mathematical Programming, 55, pp. 293-318, 1992.