Matrix Algebra and Optimization for Statistics and Machine Learning

Yiyuan She

Department of Statistics, Florida State University

- Augmented Lagrangian and ADMM

Primal vs. dual

- Consider a convex problem $\min l(\beta) + P(\beta)$
  - A primal method: proximal gradient descent
- Consensus form: $\min_{\beta,\gamma} l(\beta) + P(\gamma)$ s.t. $\beta = \gamma$
  - Notice the affine constraint
- We can introduce the Lagrangian to design a dual algorithm or a **primal-dual** algorithm such as ADMM

Dual (sub)gradient

- Dual for the general problem $\min_\beta f(\beta)$ s.t. $A\beta = c$:

$$\max_\nu g(\nu) = \max_\nu \inf_\beta L(\beta, \nu) = \max_\nu \inf_\beta f(\beta) + \langle \nu, A\beta - c\rangle$$
$$= \max_\nu -f^*(-A^T\nu) - c^T\nu \text{ (cvx)}$$

- The dual **subgradient** method ($A\beta^{t+1} - c \in \partial g(\nu^t)$):

$$\beta^{t+1} \in \arg\min_\beta L(\beta, \nu^t), \quad \nu^{t+1} = \nu^t + \alpha_t(A\beta^{t+1} - c)$$

- When $g \in \mathcal{C}^{(1)}$, it becomes dual **ascent**. Stepsize?

Strong convexity and strong smoothness

- When ∇g is Lipschitz continuous, or more generally, $\mathbf{\Delta}_g(\nu, \nu') \leq L\mathbf{D}_2(\nu, \nu')$, a universal stepsize can be used
- $h(x)$ is β-strongly smooth with respect to a norm $\|\cdot\|$ iff $\mathbf{\Delta}_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y\rangle \leq \frac{\beta}{2}\|x - y\|^2$
- Assume h is closed and convex. Then h is α-strongly convex wrt $\|\cdot\|$ iff h^* is $1/\alpha$-strongly smooth wrt $\|\cdot\|_*$
- Back to the problem, in either sense, it seems that some kind of 'strong convexity' in $A\beta$ is desired

- We illustrate the main idea assuming differentiability
- Let (x, x^*) be a dual pair ($x^* = \nabla h(x)$, $x = \nabla h^*(x^*)$)
- $g(\delta) := \mathbf{\Delta}_{h^*}(x^* + \delta, x^*) \leq (1/\alpha)\|\delta\|^2 \Leftrightarrow g^*(y) \geq \alpha\|y\|_*^2$ (noticing that $(\lambda\|\cdot\|^2)^* = \lambda\|\cdot/\lambda\|_*^2 = (1/\lambda)\|\cdot\|_*^2$)
- $g^*(y) = \sup_\delta \langle y + \nabla h^*(x^*), \delta\rangle - h^*(x^* + \delta) + h^*(x^*) = \sup_{\delta'} \langle y + \nabla h^*(x^*), \delta'\rangle - h^*(\delta') + h^*(x^*) - \langle y + x, x^*\rangle$
- From Fenchel-Young, $g^*(y) = h(y + x) - h(x) + \langle x, x^*\rangle - \langle y + x, x^*\rangle = \mathbf{\Delta}_h(y + x, x)$

Augmented Lagrangian

- Add a quadratic term in the objective:

$$\min_{\beta} f(\beta) + \frac{\rho}{2}\|A\beta - c\|_2^2 \quad \text{s.t. } A\beta = c$$

  which is obviously equivalent to the original problem
- Its Lagrangian is called the *augmented* Lagrangian:

$$L_{\rho}(\beta, \nu) = f(\beta) + \langle \nu, A\beta - c \rangle + \frac{\rho}{2}\|A\beta - c\|_2^2$$

- Dual ascent now works!

The method of multipliers

- Assume f is convex and $\rho > 0$. The algorithm is

$$\beta^{t+1} \in \arg\min_{\beta} L_\rho(\beta, \nu^t),$$
$$\nu^{t+1} = \nu^t + \rho(A\beta^{t+1} - c)$$

- (Also the *proximal point* algorithm on the dual)
- From the 's-convexity', it is perhaps not surprising to see the penalty parameter is used as the dual stepsize

- With this stepsize, (β^{t+1}, ν^{t+1}) is always dual feasible:

$$\begin{aligned}\nabla f(\beta^{t+1}) + A^T\nu^{t+1} &= \nabla f(\beta^{t+1}) + A^T(\nu^t + \rho(A\beta^{t+1} - c)) \\ &= \nabla_\beta L_\rho(\beta^{t+1}, \nu^t) = 0\end{aligned}$$

- Optimality conditions: $A\beta^\star = c, \nabla f(\beta^\star) + A^T\nu^\star = 0$
- But on the problem $\min l(\beta) + P(\gamma)$ s.t. $\beta = \gamma$ we need

$$\begin{aligned}&(\beta^{t+1}, \gamma^{t+1}) \in \arg\min_{\beta,\gamma} L_\rho(\beta, \gamma, \nu^t) \\ &\nu^{t+1} = \nu^t + \rho(\beta^{t+1} - \gamma^{t+1})\end{aligned}$$

Alternating direction method of multipliers

- Instead of doing the joint optimization wrt (β, γ), ADMM runs **BCD** for just one cycle

$$
\begin{aligned}
\beta^{t+1} &= \arg\min_{\beta} L_\rho(\beta, \gamma^t, \nu^t) \\
\gamma^{t+1} &= \arg\min_{\gamma} L_\rho(\beta^{t+1}, \gamma, \nu^t) \\
\nu^{t+1} &= \nu^t + \rho(\beta^{t+1} - \gamma^{t+1})
\end{aligned}
$$

- From $L_\rho = l(\beta) + P(\gamma) + \nu^T(\beta - \gamma) + \frac{\rho}{2}\|\beta - \gamma\|_2^2$, we can write it in the proximal form $\beta^{t+1} = \text{prox}_{l/\rho}(\gamma^t - \frac{\nu^t}{\rho})$, $\gamma^{t+1} = \text{prox}_{P/\rho}(\beta^{t+1} + \frac{\nu^t}{\rho})$, $\nu^{t+1} = \nu^t + \rho(\beta^{t+1} - \gamma^{t+1})$

Scaled ADMM

- A convenient reparametrization $\nu \leftarrow \nu/\rho$ gives

$$\begin{aligned}
\beta^{t+1} &= \mathrm{prox}_{l/\rho}(\gamma^t - \nu^t) \\
\gamma^{t+1} &= \mathrm{prox}_{P/\rho}(\beta^{t+1} + \nu^t) \\
\nu^{t+1} &= \nu^t + (\beta^{t+1} - \gamma^{t+1})
\end{aligned}$$

- If l and P are both indicators, ρ can be removed!
  - **DP**: ℓ_2 loss, 2 dual variables, 2 updates in each epoch
- The (scaled) augmented Lagrangian can be written as $l(\beta) + P(\gamma) + (\rho/2)\{\|\beta - \gamma + \nu\|_2^2 - \|\nu\|_2^2\}$

A general form

- ADMM is usually described for the general problem of

$$\min_{\beta,\gamma} l(\beta) + P(\gamma), \text{ s.t. } A\beta + B\gamma = c$$

- Examples: features (X), sparsity patterns (T)
- Augmented Lagrangian: $L_\rho(\beta, \gamma, \nu) = l(\beta) + P(\gamma)$ $+ \nu^T(A\beta + B\beta - c) + (\rho/2)\|A\beta + B\gamma - c\|_2^2$
- The (unscaled) ADMM is then given by

$$\begin{cases} \beta^{t+1} & \in \arg\min_\beta L_\rho(\beta, \gamma^t, \nu^t), \\ \gamma^{t+1} & \in \arg\min_\gamma L_\rho(\beta^{t+1}, \gamma, \nu^t), \\ \nu^{t+1} & = \nu^t + \rho(A\beta^{t+1} + B\gamma^{t+1} - c) \end{cases}$$

Linearized ADMM

- Consider the problem of $\min l(\beta) + P(\gamma)$ s.t. $\gamma = T\beta$ where $\text{prox}_{\alpha P}$ is easy to calculate while l is not simple
- The β-optimization step is

$$\beta^{t+1} \in \arg\min_{\beta} l(\beta) + P(\gamma^t) + \langle \nu^t, T\beta - \gamma^t \rangle + \frac{\rho}{2} \|T\beta - \gamma^t\|_2^2$$

- A good idea is to linearize the complex loss $l(\beta)$, or $\|T\beta - \gamma^t\|_2^2$, or both, to give an update of β (no loop)
- Not much sacrifice in convergence in the convex setup
- Nesterov's accelerations can be applied

Convergence

- In general, ADMM is as slow (fast) as GD: $\mathcal{O}(1/T)$
- Convergence of residual/objective/dual variable is easy to show, but not the convergence of primal variables!
- In practice, ADMM may be slow in high dimensions
- But it is useful when modest accuracy suffices

The penalty parameter

- Theoretically, ρ just needs to be positive
- Practically, we might want to set ρ appropriately large to yield a primal optimal solution in time
- Typically, the larger the value of ρ is, the slower the convergence is (for solving the primal)
- See Boyd et al. (2011) for an ad-hoc varying scheme
- In large problems, it can be tricky to pick a good ρ

Other variants

- *Inexact* minimization: β, γ optimization steps do not have be carried out exactly
- The β, γ updates can be performed *multiple* times
- Add an additional *dual*-update step after updating β
- Momentum-based *acceleration* for β, γ, or ν
- Many other related operator splitting methods exist

Example: ℓ_1

- Least absolute deviations: $\min_\beta \|y - X\beta\|_1$
- $\min \|r\|_1$ s.t. $r = y - X\beta$:
  - $L_\rho = \|r\|_1 + \langle \nu, r - y + X\beta \rangle + (\rho/2)\|r - y + X\beta\|_2^2$
  - β: OLS; r: Θ_{soft}
- $\min \iota_{y - X\beta - s = 0} + \|r\|_1$ s.t. $r = s$:
  - $L_\rho = \iota_{y - X\beta - s = 0} + \|r\|_1 + \langle \nu, r - s \rangle + (\rho/2)\|r - s\|_2^2$
  - (β, s): projection (OLS); r: Θ_{soft}

- Quantile lasso: $\min_\beta \|y - X\beta\|_1 + \lambda\|\beta\|_1$
- $\min \|r\|_1 + \lambda\|\beta\|_1$ s.t. $r = X\beta - y$
  - $L_\rho = \|r\|_1 + \lambda\|\beta\|_1 + \langle \nu, r - X\beta + y\rangle + (\rho/2)\|r - X\beta + y\|_2^2$
  - β: lasso; r: Θ_{soft}
- SVM: $\min_\beta \sum(1 - y_i x_i^T \beta)_+ + \lambda\|\beta\|_2^2/2$
- $\min \sum(r_i)_+ + (\lambda/2)\|\beta\|_2^2$ s.t. $r = 1 - y \circ (X\beta)$
  - $L_\rho = 1^T r_+ + \lambda\|\beta\|_2^2/2 + \langle \nu, r - 1 + \text{diag}\{y\}X\beta\rangle + (\rho/2)\|r - 1 + \text{diag}\{y\}X\beta\|_2^2$
  - β: ridge regression; r: Θ_{soft} $(2\|r_+\|_1 = 1^T r + \|r\|_1)$

Example: graph learning

- Gaussian: $\min\langle\hat{\Sigma}, \Omega\rangle - \log\det\Omega + \lambda\|\Omega\|_1$
- Proximal gradient/Newton, (dual) BCD, ADMM
- $\min\,\langle\hat{\Sigma}, \Phi\rangle - \log\det\Phi + \lambda\|\Omega\|_1$ s.t. $\Phi = \Omega$
  - $L_\rho = \langle\hat{\Sigma}, \Phi\rangle - \log\det\Phi + \lambda\|\Omega\|_1 + \langle Z, \Phi - \Omega\rangle + \rho\|\Phi - \Omega\|_F^2$
  - Ω: Θ_{soft}; Φ: analytic form available!
- **Ising** model: $p(x|\Omega) = \exp(x^T\Omega x + b^T x)/Z(\Omega)$, where $x_j = 0, 1$. Large-p: the (normalizing) Z is intractable!
- WLOG, assume $\Omega = \Omega^T$ and $b = 0$ $(\omega_{j,j} \leftarrow \omega_{j,j} + b_j)$. The full neg-log-likelihood is $-\langle X^T X, \Omega\rangle + n\log Z(\Omega)$

Pseudo-likelihood **approximation**

- $p(x_j|x_{-j},\Omega) = \exp(\omega_{jj}x_j + \sum_{k\neq j}\omega_{j,k}x_jx_k - z)$, with the local normalizing 'constant' easy to evaluate ($x_j = 0, 1$) $z(\Omega, x_{-j}) = z(\Omega[j,], x_{-j}) = \log(1+\exp(\omega_{j,j}+\sum_{k\neq j}\omega_{j,k}x_{i,k}))$
  - Node-wise logistic regression (neighborhood approach)
- Pseudo-likelihood Ising graph learning ($X = [x_{i,j}]_{n\times p}$):

$$\min_{\Omega=[\omega_{j,k}]} \|\lambda\circ\Omega\|_1 + \sum_{i=1}^{n}\sum_{j=1}^{p}\{-\sum_{k=1}^{p} x_{i,j}x_{i,k}\omega_{j,k} + z(\Omega, X[i,-j])\}$$

$$= -\langle X^TX, \Omega\rangle + \sum_{i=1}^{n}\sum_{j=1}^{p} z(\Omega, X[i,-j]) + \|\lambda\circ\Omega\|_1$$

- ADMM (or proximal methods) can be similarly applied

- Latent variable graphical model

$$\min_{S,L} \langle S - L, \hat{\Sigma} \rangle - \log\det(S - L) + \lambda \|S\|_1 + \lambda' tr(L)$$
$$\text{s.t. } S - L \succ 0, L \succeq 0$$

- $\min_{S,L} \langle R, \hat{\Sigma} \rangle - \log\det(R) + \lambda \|S\|_1 + \lambda' tr(L) + \iota_{L \succeq 0}$ s.t. $R = S - L$ (notice the log-det barrier)
- R: analytic form available; S: Θ_{soft}; L: SVD truncation

Example: nonnegative matrix factorization

- NMF has wide applications in machine learning and can achieve *parts*-based representation
- Approximate $X \geq 0$ by WH with $W, H \geq 0$

$$\min_{W \in \mathbb{R}^{n \times r}, H \in \mathbb{R}^{r \times m}} \|X - WH\|_F^2 \text{ s.t. } W_{ij} \geq 0, H_{ij} \geq 0$$

- This is a nonconvex but bilinear problem
- $\min \|X - Z\|_F^2 + \iota_{W \geq 0} + \iota_{H \geq 0}$ s.t. $Z = WH$:
  - $\|X - Z\|_F^2 + \iota_{W \geq 0} + \iota_{H \geq 0} + \langle \nu, Z - WH \rangle + \rho \|Z - WH\|_F^2$
  - (Z, W): OLS + NLS; H: NLS

Example: Fantope

- Recall Fantope for sparse PCA

$$\max_{P \in \mathcal{F}^r} \langle \Sigma, P \rangle - \lambda \| \operatorname{vec}(P) \|_1$$

  where $\mathcal{F}^r = \{P : 0 \preceq P \preceq I, tr(P) = r\}$
- $\min_P -\langle \Sigma, P \rangle + \lambda \| \operatorname{vec}(P) \|_1 + \iota_{\mathcal{F}^r}(Q)$ s.t. $P = Q$
  - $-\langle \Sigma, P \rangle + \lambda \| \operatorname{vec}(P) \|_1 + \iota_{\mathcal{F}^r}(Q) + \langle \nu, P - Q \rangle + \rho \| P - Q \|_F^2$
  - P: Θ_{soft}; Q: Fantope projection
  - Suffices to solve $\min_t \|d - t\|_2^2$ s.t. $0 \leq t_i \leq 1$, $\sum t_i = r$
- Alternatively, we can use $\min_P -\langle \Sigma, P \rangle + \lambda \| \operatorname{vec}(P) \|_1$ $+ \iota_{tr(Q)=r} + \iota_{0 \preceq R \preceq I}$ s.t. $P = Q = R$