**ECE 490: Introduction to Optimization** **Fall 2018**

# Lecture 20

**Review of Covered Materials, Part II**

*Lecturer: Bin Hu, Date:11/8/2018*

Today we review some materials that are relevant to Midterm 2. Here is a list of relevant topics.

1. Newton's method
2. BFGS method
3. Proximal gradient method
4. LASSO, ISTA, and shrinkage operator
5. Lagrangian multipliers
6. Duality
7. Augmented Lagrangian and ADMM

## 20.1 Newton's Method and BFGS Method

Newton's method uses the Hessian information and iterates as

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

When the iterates are closed to a strict local min, the Hessian information is quite helpful.

However, the Hessian computation can be expensive for many applications. This motivates the developments of Quasi-Newton methods which estimate the Hessian $\nabla^2 f(x_k)$ with some simpler matrix $H_k$. The most popular Quasi-Newton method is the BFGS method that iterates as $x_{k+1} = x_k - \alpha_k H_k^{-1} \nabla f(x_k)$ and calculates $H_k^{-1}$ in the following way:

$$H_{k+1}^{-1} = \left( I - \frac{s_k y_k^\mathsf{T}}{y_k^\mathsf{T} s_k} \right) H_k^{-1} \left( I - \frac{y_k s_k^\mathsf{T}}{y_k^\mathsf{T} s_k} \right) + \frac{s_k s_k^\mathsf{T}}{y_k^\mathsf{T} s_k} \tag{20.1}$$

where $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

## 20.2 Proximal Gradient Method and ISTA

When the objective function is convex but not differentiable, the subgradient method can be used. However, the subgradient method is quite slow. When the objective function is a sum of a smooth convex function and a non-smooth (but relatively simple) convex function, the proximal gradient method can be used. When the objective function is $(f+g)$, one can apply the proximal gradient method $x_{k+1} = \text{prox}_{g,\alpha}(x_k - \alpha \nabla f(x_k))$ if $\text{prox}_{g,\alpha}$ can be efficiently computed. If $f$ is $L$-smooth and $m$-strongly convex, and $g$ is convex, the convergence rate of the proximal gradient method is the same as the rate of the gradient method applied to an $L$-smooth $m$-strongly convex function. This can be proved using the dissipation inequality approach. See the second section in Lecture 16 for this proof.

If we apply the proximal gradient method to the LASSO problem (or other $\ell_1$-regularized problems), the proximal step can be done using the shrinkage operator $S_{\mu\alpha}$. See the first section in Lecture 16 for more discussion on the shrinkage operator.

## 20.3 Lagrangian and Duality

We have talked about the necessary condition for optimization with equality constraint in the first section of Lecture 18. What about sufficient conditions? Now we state one useful sufficient condition for a local min of an optimization problem with equality constraints. Suppose the objective function is $f$ and the equality constraint is $h(x) = 0$. We assume $f$ and $h$ are twice continuously differentiable. If there exists $\lambda^*$ and $x^*$ such that $\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) + (\lambda^*)^\mathsf{T} \nabla h(x^*) = 0$, $h(x^*) = 0$, and $d^\mathsf{T} \nabla^2_{xx} L(x^*, \lambda^*) d > 0$ for all $d \neq 0$ satisfying $d^\mathsf{T} \nabla h(x^*) = 0$, then $x^*$ is a strict local min of $f$ subject to the constraint $h(x) = 0$.

**Ex 1.** Consider the following problem

$$\begin{aligned} \text{minimize} \quad & -(x_1x_2 + x_2x_3 + x_1x_3) \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 3 \end{aligned} \tag{20.2}$$

By setting $\nabla_x L(x^*, \lambda^*) = 0$ and $h(x^*) = 0$, we have

$$\begin{aligned} -x_2^* - x_3^* + \lambda^* &= 0 \\ -x_1^* - x_3^* + \lambda^* &= 0 \\ -x_1^* - x_2^* + \lambda^* &= 0 \\ x_1^* + x_2^* + x_3^* &= 3 \end{aligned}$$

We have four variables and four linear equations. The equations have a unique solution $x_1^* = x_2^* = x_3^* = 1$, and $\lambda^* = 2$. Next we compute the Hessian information $\nabla^2_{xx} L(x^*, \lambda^*)$ and obtain

$$\nabla^2_{xx} L(x^*, \lambda^*) = \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}$$

Suppose $d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$. The condition $d^\mathsf{T} \nabla h(x^*) = 0$ just states $d_1 + d_2 + d_3 = 0$. We have

$$d^\mathsf{T} \nabla^2_{xx} L(x^*, \lambda^*) d = -2(d_1 d_2 + d_2 d_3 + d_1 d_3)$$

Since $(d_1 + d_2 + d_3)^2 = 0$, we have $-2(d_1 d_2 + d_2 d_3 + d_1 d_3) = d_1^2 + d_2^3 + d_3^2 > 0$ for any $d \neq 0$. Therefore, the sufficient condition is satisfied, and we have a strict local min here.

It is worth mentioning that the sufficient condition is met if $\nabla^2_{xx} L(x^*, \lambda^*) > 0$. However, in the above example, even though we do not have the positive definiteness of $\nabla^2_{xx} L(x^*, \lambda^*)$, the proposed sufficient condition still works.

The duality theory is also very important. See Lecture 18 for more discussions. One important thing is calculating the dual function for a given Lagrangian. Also see HW 5 for such problems.

## 20.4 Augmented Lagrangian and ADMM

Augmented Lagrangian is introduced to help the convergence of the gradient ascent on the dual variable. ADMM can be viewed as the "decomposable method of multipliers", and addresses the following problem

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) \\ \text{subject to} \quad & Ax + By = c \end{aligned} \tag{20.3}$$

We first formulate the augmented Lagrangian:

$$L_\rho(x, y, \lambda) = f(x) + g(y) + \lambda^\mathsf{T}(Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|^2$$

ADMM alternates the minimization over $x$ and $y$. Specifically, ADMM iterates as

$$\begin{aligned} x_{k+1} &= \arg\min_x L_\rho(x, y_k, \lambda_k) \\ y_{k+1} &= \arg\min_y L_\rho(x_{k+1}, y, \lambda_k) \\ \lambda_{k+1} &= \lambda_k + \rho(Ax_{k+1} + By_{k+1} - c) \end{aligned}$$

Compared with the method of multipliers, the main advantage of ADMM is that sometimes the computation of $\arg\min_x L_\rho(x, y_k, \lambda_k)$ and $\arg\min_y L_\rho(x_{k+1}, y, \lambda_k)$ can still be parallelized even when the computation of $\arg\min_{x,y} L_\rho(x, y, \lambda_k)$ cannot be parallelized. It is important to understand how to write out primal and dual updates of ADMM for a given problem. See the examples in Lecture 19 and HW 5.