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

Lecture 19

Alternating Direction Method of Multipliers (ADMM)

Lecturer: Bin Hu, Date:11/6/2018

In this lecture, we focus on optimization with linear equality constraints. We will talk about the augmented Lagrangian function, the method of multipliers, and the alternating direction method of multipliers (ADMM).

19.1 Augmented Lagrangian Function

In the last lecture, we have talked about the dual ascent method and the primal-dual gradient dynamics. These methods apply gradient ascent to the dual variable. The shape of the dual function determines how well the gradient ascent on the dual variable works. Suppose the objective function is $f(x)$ and the constraint is $Ax = b$. Recall that the dual function is defined as

$$D(\lambda) = \min_x \{f(x) + \lambda^\mathsf{T}(Ax - b)\}$$

In many situations, $D(\lambda)$ is not even differentiable. Just imagine f is a linear function and the dual function can just become $-\infty$. To overcome this issue, the augmented Lagrangian function is introduced. The idea of the augmented Lagrangian is based on reformulating the original constrained minimization problem. Suppose the original problem is

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h(x) = 0 \end{aligned} \tag{19.1}$$

The above problem is actually equivalent to

$$\begin{aligned} \text{minimize} \quad & f(x) + \tfrac{\rho}{2}\|h(x)\|^2 \\ \text{subject to} \quad & h(x) = 0 \end{aligned} \tag{19.2}$$

where $\rho > 0$ is a hyperparameter. The two problems are completely equivalent due to the fact that eventually the optimal point has to satisfy the equality constraint $h(x) = 0$. Verify this by yourself! Now we can write out the Lagrangian for (19.2) as

$$L_\rho(x, \lambda) = f(x) + \lambda^\mathsf{T} h(x) + \frac{\rho}{2}\|h(x)\|^2 \tag{19.3}$$

which is exactly the augmented Lagrangian for the original problem (19.1). Now the differentiability of the function $\min_x L_\rho(x, \lambda)$ is improved, and this helps the gradient ascent step on the dual variable. The definition of augmented Lagrangian is general and covers the case where h is nonlinear. Next, we focus on linear equality constraints and introduce algorithms that are developed based on the augmented Lagrangian. From now on, we assume $h(x) = Ax - b$.

19.2 Method of Multipliers

The method of multipliers can be viewed as an extension of the dual ascent method. The difference is that the method of multipliers uses the augmented Lagrangian. Consider the constrained minimization problem with an equality constraint $Ax = b$. Notice $\nabla_\lambda L_\rho(x, \lambda) = Ax - b$. Therefore, the method of multipliers iterates as

$$x_{k+1} = \arg\min_x L_\rho(x, \lambda_k) \tag{19.4}$$

$$\lambda_{k+1} = \lambda_k + \rho(Ax_{k+1} - b) \tag{19.5}$$

Notice the stepsize for the dual variable update is exactly ρ. This stepsize is used to ensure $\nabla f(x_{k+1}) + A^\mathsf{T}\lambda_{k+1} = 0$. Recall that we need $\nabla_x L(x, \lambda) = 0$ and $\nabla_\lambda L(x, \lambda) = 0$. So the stepsize ρ just ensures $\nabla_x L(x_{k+1}, \lambda_{k+1}) = 0$ for all $k > 1$.

The method of multipliers has much better convergence properties compared with the dual ascent method. However, it also has some disadvantages as we will discuss in the next section.

19.3 Decomposition Issues for Method of Multipliers

For separable f, it is beneficial if we can parallelize the computation. Assume $x = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^p \end{bmatrix}$ where x^i is a scalar. If f is separable, i.e. $f(x) = f_1(x^1) + \ldots + f_p(x^p)$, the Lagrangian is also separable. We rewrite A as $\begin{bmatrix} A_1 \ldots A_p \end{bmatrix}$. When applying the dual ascent method, we have

$$x_{k+1} = \arg\min_x L(x, \lambda_k) = \arg\min_x \sum_{i=1}^{p} \left\{ f_i(x^i) + \lambda_k^\mathsf{T}(A_i x^i - b) \right\}$$

So the update for x_{k+1} can be parallelized as

$$x_{k+1}^i = \arg\min_{x^i} \{ f_i(x^i) + \lambda_k^\mathsf{T} A_i x^i \}$$

The dual variable update provides the required coordination:

$$\lambda_{k+1} = \lambda_k + \rho \left(\sum_{i=1}^{p} A_i x_{k+1}^i - b \right)$$

So after the parallel computing of x_{k+1}^i, one computer will gather all the information for x_{k+1}^i and then compute λ_{k+1}.

The key in the parallel implementation of the dual ascent method is that $\lambda^\mathsf{T} Ax$ is a separable function of x. However, if we apply the method of multipliers, the term $\|Ax - b\|^2$

is not separable in general and destroys the splitting of the primal variable update. The method of multipliers can not be parallelized for general A due to this decomposition issue. When A is an identity matrix, the term $\|Ax-b\|^2$ becomes $\|x-b\|^2$ and is still separable. The method of multipliers can still be parallelized in this case. What if $A=\begin{bmatrix}D & I\end{bmatrix}$? In these cases, we can at least partially fix the decomposition issue using ADMM. Therefore, ADMM can be viewed as the "decomposable method of multipliers."

19.4 ADMM

ADMM addresses the following problem

$$\begin{aligned} \text{minimize} \quad & f(x)+g(y) \\ \text{subject to} \quad & Ax+By=c \end{aligned} \tag{19.6}$$

Again, we 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

$$x_{k+1}=\arg\min_x L_\rho(x,y_k,\lambda_k) \tag{19.7}$$

$$y_{k+1}=\arg\min_y L_\rho(x_{k+1},y,\lambda_k) \tag{19.8}$$

$$\lambda_{k+1}=\lambda_k+\rho(Ax_{k+1}+By_{k+1}-c) \tag{19.9}$$

If we apply the method of multipliers to (19.6), the following iteration is adopted:

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

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.

Ex1: LASSO. Consider the LASSO problem $\min_x \frac{1}{2}\|Ax-b\|^2+\mu\|x\|_1$. This problem can be equivalently rewritten as

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\|Ax-b\|^2+\mu\|y\|_1 \\ \text{subject to} \quad & x-y=0 \end{aligned}$$

For the above problem, the augmented Lagrangian is given as

$$L_\rho(x, y, \lambda) = \frac{1}{2}\|Ax - b\|^2 + \mu\|y\|_1 + \lambda^\mathsf{T}(x - y) + \frac{\rho}{2}\|x - y\|^2$$

It is not straightforward to obtain $\arg\min_{x,y} L_\rho(x, y, \lambda_k)$ for the above augmented Lagrangian. Hence it is not a good idea to directly apply the method of multipliers. However, the computation of $\arg\min_x L_\rho(x, y_k, \lambda_k)$ and $\arg\min_y L_\rho(x_{k+1}, y, \lambda_k)$ are quite straightforward, and hence we should be able to apply ADMM for the above problem. We have

$$x_{k+1} = \arg\min_x \left\{\frac{1}{2}\|Ax - b\|^2 + \lambda_k^\mathsf{T} x + \frac{\rho}{2}\|x - y_k\|^2\right\}$$

By the optimality condition of strongly-convex functions, we have $A^\mathsf{T}(Ax_{k+1} - b) + \lambda_k + \rho(x_{k+1} - y_k) = 0$ and hence $x_{k+1} = (A^\mathsf{T}A + \rho I)^{-1}(A^\mathsf{T}b - \lambda_k + \rho y_k)$. Notice $(A^\mathsf{T}A + \rho I)^{-1}$ can be efficiently computed in the Fourier domain for a lot of imaging applications. This makes ADMM an attractive approach for these applications. Similarly, we have

$$y_{k+1} = \arg\min_y \left\{\mu\|y\|_1 - \lambda_k^\mathsf{T} y + \frac{\rho}{2}\|x_{k+1} - y\|^2\right\}$$

This step is similar to the proximal operator update in ISTA, and can be efficiently parallelized using the shrinkage operator. Specifically, one can get $y_{k+1} = S_{\mu/\rho}(x_{k+1} + \lambda_k/\rho)$ where $S_{\mu/\rho}$ is the shrinkage operator that shrinks any value between $-\mu/\rho$ and μ/ρ to 0. Similar to the proximal step in ISTA, the shrinkage operation can be easily parallelized for ADMM. The update for λ_{k+1} is still trivially $\lambda_{k+1} = \lambda_k + \rho(x_{k+1} - y_{k+1})$. Putting all the pieces together, the ADMM iteration for the LASSO problem is as follows

$$\begin{aligned}
x_{k+1} &= (A^\mathsf{T}A + \rho I)^{-1}(A^\mathsf{T}b - \lambda_k + \rho y_k) \\
y_{k+1} &= S_{\mu/\rho}(x_{k+1} + \lambda_k/\rho) \\
\lambda_{k+1} &= \lambda_k + \rho(x_{k+1} - y_{k+1})
\end{aligned}$$

For many imaging applications, the computation for x_{k+1} and y_{k+1} can be efficiently parallelized. This makes ADMM even more efficient than FISTA for these applications.

Ex2: Consensus optimization. Consider the empirical risk minimization $\sum_{i=1}^n f_i(x)$. This problem can be equivalently rewritten as

$$\begin{aligned}
\text{minimize} \quad & \tfrac{1}{2}\textstyle\sum_{i=1}^n f_i(x^i) \\
\text{subject to} \quad & x^i - y = 0
\end{aligned}$$

where x^i is a vector having the same dimension as x. It is straightforward to write the augmented Lagrangian in a sum form:

$$L_\rho = \sum_{i=1}^n \left(f_i(x^i) + (\lambda^i)^\mathsf{T}(x^i - y) + \frac{\rho}{2}\|x^i - y\|^2\right)$$

where λ^i has the same dimension as x^i. This formulation leads to a natural parallelization of the update for x^i_{k+1}. Eventually the ADMM update rule for this problem is

$$
\begin{aligned}
x^i_{k+1} &= \arg\min_{x^i} \left\{ f_i(x^i) + (\lambda^i_k)^\mathsf{T}(x^i - y_k) + \frac{\rho}{2}\|x^i - y_k\|^2 \right\} \\
y_{k+1} &= \frac{1}{n}\sum_{i=1}^{n} \left(x^i_{k+1} + \frac{\lambda^i_k}{\rho} \right) \\
\lambda^i_{k+1} &= \lambda^i_k + \rho(x^i_{k+1} - y_{k+1})
\end{aligned}
$$