EE 381V: Large Scale Optimization **Fall 2012**

Lecture 25 — November 29

Lecturer: Caramanis & Sanghavi *Scribe: Jeff Mahler and Zhichao Shu*

25.1 Introduction

Recall the concept of duality from previous lectures. Given a constained optimization problem:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & h(x) \leq 0 \\ & Ax = b. \end{aligned}$$

The Lagrangian for this problem is:

$$L(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu(Ax - b),$$

from which we define the dual objective:

$$g(\lambda, \mu) = \min_x \ L(x, \lambda, \mu).$$

When we have strong duality between the primal and dual problems, we can recover the primal optimum from the dual. Letting $\lambda^*, \mu^* = \text{argmax } g(\lambda, \mu)$, the optimum of the primal is

$$x^* = \operatorname*{argmin}_x \ L(x, \lambda^*, \mu^*).$$

In this lecture we introduce the Primal and Dual Decomposition, motivated by using the above relationship between the primal and dual problem to obtain a faster or parallel solution to an optimzation problem. This lecture will cover four topics:

1. Coupled variables (e.g. SVM)
2. Coupled constraints (e.g. Network Rate Control)
3. Augmented Lagrangian
4. Alternating Direction Method of Multipliers

25.2 Coupled Variables

Consider the unconstrained optimization problem:

$$\min_{x_1,x_2,y} \quad f_1(x_1,y) + f_2(x_2,y).$$

We call y the coupling variable for the problem and x_1, x_2 the local variables for the subproblems. Note that for a fixed y this problem is separable since f_1 does not depend on x_2 and f_2 does not depend on x_1. Therefore we can solve for the optimal values of f_1 and f_2 in parallel.

25.2.1 Primal Decomposition

Fixing the value of y, we have the two subproblems:

$$\begin{aligned}\text{subproblem 1} &: \min_{x_1} f_1(x_1,y)\\ \text{subproblem 2} &: \min_{x_2} f_2(x_2,y),\end{aligned}$$

with optimal values $\phi_1(y), \phi_2(y)$, respectively. Thus the original problem is equivalent to the master problem

$$\min_y \phi_1(y) + \phi_2(y)$$

called the primal decomposition of the original objective. We can solve the master problem using subgradient descent or Newton's method (if ϕ_i are differentiable), solving each of the two subproblems on every iteration. If the subproblems are solved in parallel or happen to be sufficiently easier to solve than the original problem, then this method may be faster.

25.2.2 Dual Decomposition

We can also introduce local versions y_1, y_2 of the coupled variable y to get the constrained optimization problem:

$$\begin{aligned}\min_{x_1,x_2,y_1,y_2} \quad & f_1(x_1,y_1) + f_2(x_2,y_2)\\ \text{subject to} \quad & y_1 = y_2.\end{aligned}$$

The Lagrangian for this problem is:

$$L(x_1,x_2,y_1,y_2,\nu) = f_1(x_1,y_1) + f_2(x_2,y_2) + \nu^T(y_1 - y_2).$$

This equation is separable, so we can break it up into two subproblems minimizing (x_1, y_1) and (x_2, y_2) separately:

$$\begin{aligned}\text{subproblem 1} &: \min_{x_1,y_1} f_1(x_1,y_1) + \nu^T y_1\\ \text{subproblem 2} &: \min_{x_2,y_2} f_2(x_2,y_2) - \nu^T y_2,\end{aligned}$$

with optimal values $g_1(\nu), g_2(\nu)$, respectively. The master dual problem is

$$\max_{\nu} g(\nu) = g_1(\nu) + g_2(\nu),$$

called the dual decomposition of the original objective. We can solve this master problem using subrgradient ascent using the subgradient $y_2 - y_1$ of $-g(\nu)$, giving the following algorithm:

Choose initial ν_0.
Repeat:
1. Solve subproblems 1 and 2 to obtain y_1, y_2.
2. Update $\nu_{k+1} = \nu_k - \alpha_k(y_2 - y_1)$.

One useful interpretation of the dual decomposition, borrowed from Prof. Stephen S. Boyd at Stanford University, is to consider y_1 the resources consumed by the first subproblem, y_2 the resources supplied by the second subproblem, and ν the price of resources. With this interpretation, the master algorithm is adjusting the prices at each iteration to ensure that supply equals demand ($y_1 = y_2$) rather than allocating resources directly.

Example 1 - SVM. Consider the problem of classifying a huge number of data points using a Support Vector Machine (SVM):

$$\min_{w} \quad \frac{1}{N}\sum_{i=1}^{N} l(x_i, y_i, w) + r(w),$$

where $l(x, y, w)$ is the loss function and $r(w)$ constrains the separation of the hyperplane and the support vectors. We can rewrite this unconstrained problem as a separable constrained optimization problem by breaking up the constraints into the sets S_1 and S_2:

$$\begin{aligned}
&[\min_{w_1} && \frac{1}{N}\sum_{i\in S_1} l(x_i, y_i, w_1) + \frac{1}{2}r(w_1)] \\
&&& + \\
&[\min_{w_2} && \frac{1}{N}\sum_{i\in S_2} l(x_i, y_i, w_2) + \frac{1}{2}r(w_2)] \\
&\text{subject to} && w_1 = w_2.
\end{aligned}$$

Using the dual decomposition and algorithm as described above, we can iteratively classify the sets S_1 and S_2 in parallel and update the dual variable ν on each iteration.

25.3 Coupled Constraints

Consider the constrained optimization problem with coupled constraints:

$$\begin{aligned}
&\min_{x,y} && f(x) + g(y) \\
&\text{subject to} && Ax + By = b.
\end{aligned}$$

25.3.1 Primal Decompostion

We can once again decompose the primal into two subproblems by choosing a value $t \in \mathbb{R}^m$:

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{subject to} & Ax = t \\ \min\limits_{y} & g(y) \\ \text{subject to} & By = b - t, \end{array}$$

which have optimal values $\phi_1(t)$ and $\phi_2(t)$, respectively. Once again we can form a master problem $\phi_1(t) + \phi_2(t)$ and minimize with respect to t. Repeatedly solving the subproblems in parallel and updating t will solve the original objective.

25.3.2 Dual Decompostion

Taking the Lagrangian of the original optimization problem we have

$$L(x, y, \nu) = f(x) + g(y) + \nu^T(Ax + By - b),$$

which can be separated in x and y to form the subproblems:

$$\begin{aligned} \text{subproblem 1} &: \min_x \ f(x) + \nu^T Ax \\ \text{subproblem 2} &: \min_y \ g(y) + \nu^T By. \end{aligned}$$

This can be solved using the same dual decomposition algorithm used earlier with the subgradient $Ax + By - b$, giving the update $\nu_{k+1} = \nu_k - \alpha_k(Ax + By - b)$.

Example 2 - Network Rate Control. Consider the problem of allocating rates of traffic flow x in a network:

$$\begin{array}{ll} \text{maximize}\limits_{x} & \sum\limits_j U_j(x_j) \\ \text{subject to} & Rx \preceq C, \end{array}$$

where each U_j is concave, R is a matrix where $R_{i,j} = 1$ if flow j passes over link i and $R_{i,j} = 0$ otherwise and C is the capacity of each link. The Lagrangian for this problem is:

$$\begin{aligned} L(x, \nu) &= \sum_j U_j(x_j) + \nu^T(Rx - C) \\ &= \sum_j (U_j(x_j) + \sum_{l \in R_j} \nu_l x_j) - \nu^T C. \end{aligned}$$

Thus the Lagrangian is a sum of j decoupled subproblems. Using the dual decomposition algorithm, given an initial ν we can solve the optimal x_j for each subproblem in parallel and update ν as $\nu_{k+1} = \nu_k + \alpha_k(C - Rx)$ on each iteration.

25.3.3 Limitations

Though the dual decomposition may yield a faster or parallel solution to optimization problems, it is limited by the fact that for a convex optimization problem:

$$\begin{aligned} \min_{x_1,x_2,y_1,y_2} \quad & f_1(x_1,y_1) + f_2(x_2,y_2) \\ \text{subject to} \quad & y_1 = y_2, \end{aligned}$$

we cannot guarantee that $y_1^{(k)} - y_2^{(k)} \to 0$ as $k \to \infty$ in general, so the solution to the dual decomposition may not be the solution to our original problem. This is due to the fact that a general convex optimization problem may have multiple optimal solutions. However, if the objective is strictly convex, then it has an unique minimizer and therefore $y_1^{(k)} - y_2^{(k)} \to 0$ as $k \to \infty$. This motivates the Augmented Lagrangian method.

25.4 Augmented Lagrangian

Consider the following constrained problem:

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & Ax = b. \end{aligned}$$

The Lagrangian of the problem is:

$$L(x,\lambda) = f(x) + \lambda^T(Ax - b).$$

The Augumented Lagrangian of the problem is:

$$AugL(x,\lambda,\rho) = f(x) + \lambda^T(Ax - b) + \tfrac{\rho}{2}||Ax - b||_2^2.$$

25.4.1 Algorithm

$$\begin{aligned} x^{(k+1)} &= \arg\min_x L(x,\lambda^{(k)}), \\ \lambda^{(k+1)} &= \lambda^{(k)} + \rho(Ax^{(k+1)} - b) \end{aligned}$$

This method is not vanilla gradient ascent on $L(x,\lambda)$ on Augmented Lagrangian, but these updates will get to the optimum of the original $L(x,\lambda)$. In order to prove the convergence of this method, it is useful to first introduce the Proximal Point Algorithm.

25.4.2 Proximal Point Algorithm

Consider the unconstrained optimization problem:

$$\min \quad f(x)$$

Update x using the following equation:

$$x^{(k)} = \arg\min_u [f(u) + \tfrac{1}{2t^{(k)}}||u - x^{(k-1)}||_2^2]$$

We can assume that for large $t^{(k)}$, the problem is "hard" to solve; for small $t^{(k)}$, the problem is "easy".

Next we will show the convergence of the proximal point algorithm, the procedure is very similar to what we did for proximal gradient algorithm.

It is easy to see that $x^{(k)} = \text{Prox}_{t^{(k)}f}(x^{(k-1)})$, for simplicity, we write this as: $x^+ = \text{Prox}_{tf}(x)$. Define $G_t(x) = \frac{1}{t}(x - \text{Prox}_{tf}(x))$, then we have $x^+ = x - tG_t(x)$.

Claim 25.1. $G_t(x) \in \partial f(x - tG_t(x))$

Proof: Notice that $x - tG_t(x) = x^+ = \text{Prox}_{tf}(x)$; and recall a basic property of the proximal mapping, which is immediate from the definition:

$$u \in \text{Prox}_h(x) \Leftrightarrow x - u \in \partial h(u)$$

Thus we have:

$$\begin{aligned} x - tG_t(x) - x &\in \partial f(x - tG_t(x)) \\ \Rightarrow G_t(x) &\in \partial f(x - tG_t(x)) \end{aligned}$$

□

Claim 25.2.

$$f(x^+) \leq f(z) + G_t(x)^T(x - z) - \frac{t}{2}||G_t(x)||_2^2 \tag{25.1}$$

holds for all z.

Proof: Recall that $G_t(x) \in \partial f(x - tG_t(x)) = \partial f(x^+)$, from the definition of subgradient, we have:

$$\begin{aligned} f(x^+) &\leq f(z) + G_t(x)^T(x^+ - z) \\ &= f(z) + G_t(x)^T(x - z) + G_t(x)^T(x^+ - x) \\ &= f(z) + G_t(x)^T(x - z) - t||G_t(x)||_2^2 \\ &\leq f(z) + G_t(x)^T(x - z) - \frac{t}{2}||G_t(x)||_2^2 \end{aligned}$$

□

Theorem 25.1. $f(x_{best}^{(k)}) - f^* \leq \frac{||x^{(0)}-x^*||_2^2}{2\sum_{i=1}^k t^{(i)}}$

Proof: Put $z = x^*$ in (25.1), we can get:

$$\begin{aligned} f(x^+) - f^* &\leq G_t(x)^T(x - x^*) - \frac{t}{2}||G_t(x)||_2^2 \\ &= \frac{1}{2t}[||x - x^*||_2^2 - ||x - x^* - tG_t(x)||_2^2] \\ &= \frac{1}{2t}[||x - x^*||_2^2 - ||x^+ - x^*||_2^2] \\ t^{(i)}(f(x^{(i)}) - f^*) &\leq \frac{1}{2}[||x^{(i-1)} - x^*||_2^2 - ||x^{(i)} - x^*||_2^2] \end{aligned} \tag{25.2}$$

Sum (25.2) over all $i \leq k$, we have:

$$\begin{aligned} &\sum_{i=1}^k t^{(i)}(f(x^{(i)}) - f^*) \leq \frac{1}{2}[||x^{(0)} - x^*||_2^2 - ||x^{(k)} - x^*||_2^2] \\ \Rightarrow &\sum_{i=1}^k t^{(i)}(f(x^{(i)}) - f^*) \leq \frac{1}{2}[||x^{(0)} - x^*||_2^2 \\ \Rightarrow &\sum_{i=1}^k t^{(i)}(f(x_{best}^{(k)}) - f^*) \leq \frac{1}{2}[||x^{(0)} - x^*||_2^2 \\ \Rightarrow &f(x_{best}^{(k)}) - f^* \leq \frac{||x^{(0)} - x^*||_2^2}{2\sum_{i=1}^k t^{(i)}} \end{aligned}$$

□

Back to Augmented Lagrangian, it can be shown that Augmented Lagrangian is just proximal point algorithm on $-q(\lambda)$:

$$\lambda^{(k+1)} = \arg\min_u[-q(u) + \tfrac{1}{2\rho}||u - \lambda^{(k)}||_2^2].$$

Thus we have:

$$q(\lambda^*) - q(\lambda^{(k)}) \leq \tfrac{||\lambda^{(0)}-\lambda^*||}{2k\rho}.$$

25.4.3 Augmented Lagrangian VS. simple Dual decomposition

Plus: Augmented Lagrangian has faster convergence rate with strict convexity.
Minus: Augmented Lagrangian destroys decoupling:

$$f(x) + g(y) + \lambda^T(Ax + By - b) + \tfrac{\rho}{2}||Ax + By - b||_2^2.$$

We have the cross product of x and y in $||Ax + By - b||_2^2$ and the above expression can no longer be decoupled.

25.5 Alternating Direction Method of Multipliers (ADMM)

$$x^{(k+1)} = \arg\min_x L_\rho(x, y^{(k)}, \lambda^{(k)}),$$
$$y^{(k+1)} = \arg\min_y L_\rho(x^{(k+1)}, y, \lambda^{(k)}),$$
$$\lambda^{(k+1)} = \arg\max_\lambda L_\rho(x^{(k+1)}, y^{(k+1)}, \lambda)$$

Thus method can restore simple problems for x, y minimization via alternation.

25.6 Homework problem

Sparse + Low Rank matrix decomposition

$$M = L^* + S^*,$$

where L^* is a low rank matrix and S^* is a sparse matrix.
We can decompose the matrix by solving:

$$\begin{aligned} \min \quad & ||L||_* + r||s||_1 \\ s.t. \quad & L + S = M. \end{aligned}$$