Lagrangian Relaxation

Today we will consider a method for producing a bound on the optimal value of problems of the form:

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \in X.
\end{align*}
\]  

(1)

Intuitively, we think about the constraint \( x \in X \) as “easy” and the constraints \( Ax \geq b \) as “hard”. As an example, the set \( X \) may be \( \{x \geq 0 : x \text{ integer}, Cx = d \} \), where \( C \) is a graph incidence matrix. Such a set can be quickly optimized over using the network simplex algorithm discussed in a previous recitation.

Now, we will relax the “hard” constraints \( Ax \geq b \) by removing them and inserting a penalty for violations. Let \( \lambda \) be a vector in \( \mathbb{R}^m \), and consider the new problem:

\[
\begin{align*}
\min & \quad c^T x + \lambda^T (Ax - b) \\
\text{s.t.} & \quad x \in X.
\end{align*}
\]  

(2)

Let \( L(\lambda) \) be the optimal objective value of this program. Interestingly, we can use any \( \lambda \) to get a lower bound on the optimal value of our original problem.

**Proposition 1** Let \( Z^* \) be the optimal objective of the original problem (1). Then, for any \( \lambda \in \mathbb{R}^m \) we have that \( L(\lambda) \leq Z^* \).

**Proof:** Using the definitions,

\[
\begin{align*}
Z^* &= \min_{x \in X} \{c^T x \mid Ax = b\} \\
&= \min_{x \in X} \{c^T x + \lambda^T (Ax - b) \mid Ax = b\} \\
&\geq \min_{x \in X} \{c^T x + \lambda^T (Ax - b)\} = L(\lambda).
\end{align*}
\]

\( \square \)

The Lagrangian Dual

To get the best possible lower bound, consider the problem:

\[
L^* = \max_{\lambda} L(\lambda) = \max_{\lambda} \min_{x \in X} c^T x + \lambda^T (Ax - b)
\]
We call this the Lagrangian dual problem. Note that thanks to Proposition 1 we immediately get weak Duality, i.e. \( L^* \leq Z^* \). In general, no strong duality results hold, however, lower bounds are still very useful in practice.

Note that for a fixed \( x \), \( c^T x + \lambda^T (Ax - b) \) defines a hyperplane in \( \mathbb{R}^m \). Taking the point-wise minimum of a set of hyperplanes yields a concave, piecewise-linear function (as in Figure).

![Figure 1: Example of point-wise minimum (blue) of hyperplanes.](image)

Now, the natural question is **how to solve the Lagrangian dual?**

**Finite case**

If \( X = \{x^1, \ldots, x^k\} \) is a finite set, then we can compute the value \( L^* \) by the following LP:

\[
\max_{(q, \lambda)} \quad q \\
\text{s.t.} \quad q \leq c^T x^i + \lambda^T (Ax^i - b) \quad i = 1, \ldots, k
\]

Note that when \( X \) is large, this is inefficient. However, taking the dual of this LP, we get:

\[
\min \quad \sum_j y_j (c^T x^j) \\
\text{s.t.} \quad \sum_j y_j (A_i x^j - b_i) = 0 \quad \forall i = 1, \ldots, m \\
\sum_j y_j = 1 \\
y_j \geq 0
\]

If we rearrange the equations, and use the fact that \( \sum_j y_j = 1 \), we get an equivalent representation:

\[
\min \quad c^T \left( \sum_j y_j x^j \right) \\
\text{s.t.} \quad A \left( \sum_j y_j x^j \right) = b \\
\sum_j y_j = 1 \\
y_j \geq 0
\]
Letting $\text{conv}(X)$ be the convex hull of $X$, note that $x \in \text{conv}(X)$ iff $x = \sum_j \alpha_j x^j$, $\sum_j \alpha_j = 1$, $\alpha \geq 0$, $x^j \in X$. Hence, this LP is exactly the same as optimizing over the convex hull of $X$. Hence, this can be written as:

$$\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \in \text{conv}(X)
\end{align*}$$

Hence, $L^*$ can be computed by solving this LP. This is why it is call Lagrangian relaxation, we are relaxing the restriction $x \in X$ to $x \in \text{conv}(X)$.

**Infinite case**

We could be tempted to use a Gradient Descent algorithm to solve this problem, the only issue with this is that the function that we are trying to optimize is not smooth every where. In fact, let $\lambda \in \mathbb{R}^m$ be fixed, if $x^*$ is the unique minimizer of

$$\min_{x \in X} \{c^T x + \lambda^T(Ax - b)\}$$

then, $\nabla L(\lambda) = Ax^* - b$. However, if there are multiple optimal $x^*_1, \ldots$ (i.e. many hyperplanes are intersecting that point) then we only get super gradients. Recall that a subgradient of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ at $x \in \mathbb{R}^n$ is a vector $g \in \mathbb{R}^n$ such that

$$f(y) \geq f(x) + g^T(y - x) \quad \text{for all } y \in \mathbb{R}^n.$$ 

Analogously, for a concave function a supergradient should satisfy, $f(y) \leq f(x) + g^T(y - x)$.

**Proposition 2** Let $x^*_1$ be one of the minimizers defined before, then, $Ax^*_1 - b$ is a supergradient of $L$ at $\lambda$.

**Proof:** Pick any $\mu \in \mathbb{R}^m$, we’d like to show $L(\mu) \leq L(\lambda) + (Ax^*_1 - b)^T(\mu - \lambda)$. Note that $L(\mu) \leq c^T x^*_1 + \mu^T(Ax^*_1 - b)$ by definition. Then, summing and subtracting $\lambda^T(Ax^*_1 - b)$ we get

$$L(\mu) \leq c^T x^*_1 + \mu^T(Ax^*_1 - b) + (Ax^*_1 - b)^T(\mu - \lambda)$$

$$L(\lambda) + (Ax^*_1 - b)^T(\mu - \lambda).$$

We could use this fact to derive a supergradient method to maximize $L(\cdot)$. Consider the following algorithm

1. Choose starting $\lambda^0 \in \mathbb{R}^m$.

2. Repeat:
   
   (a) Solve $x^* := \min_{x \in X} \{c^T x + (\lambda^k)^T(Ax - b)\}$ (This should be fast since we assumed that $x \in X$ is an easy constraint).

   (b) If $Ax^* - b = 0$ stop (you’ve reached an optimum).

   Otherwise, set $\lambda^{k+1} := \lambda^k + t^k(Ax^* - b)$.

In practice, people use a more relaxed stopping criteria, such as $\|Ax^*b\| \leq \varepsilon$ for a small $\varepsilon > 0$. This method is guaranteed to converge (very slowly) if the sequence of step sizes $\{t^k\}_k$ satisfies that $t^k \to 0$ as $k$ goes to infinity and $\sum_{k=0}^{\infty} t^k = \infty$. 7-3