A brief introduction to inference using Lagrangian relaxation

Vivek Srikumar

9 April, 2012

1 Introduction

This note briefly summarizes the use of Lagrangian relaxation for inference. Suppose we want to find the solution to the following problem **P**:

$$\max_{\mathbf{x}} \quad f(\mathbf{x}) \tag{1}$$
$$\text{st.} \quad \mathbf{x} \in X, \tag{2}$$
$$\mathbf{c}_i^T \mathbf{x} = b_i; \quad \forall i = 1 \cdots n \tag{3}$$

Further, suppose that solving the following maximization problem $\mathbf{P}'$ is computationally easier:

$$\max_{\mathbf{x} \in X} f(\mathbf{x}) \tag{4}$$

Lagrangian relaxation is a technique that allows us to use the computationally easier P' as a sub-routine to solve **P**.

2 The algorithm

Let λ_i be the Langrange multipliers corresponding to the constraints 3. Then, the Lagrangian is

$$L(\lambda, \mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i \left(\mathbf{c}_i^T \mathbf{x} - b_i\right) \tag{5}$$

This gives us the following dual objective for the problem **P**:

$$\max_{\mathbf{x} \in X} L(\lambda, \mathbf{x}) \equiv \Theta(\lambda) \tag{6}$$

This objective function is a convex function in λ. Note that some of the constraints ($\mathbf{x} \in X$) are not moved into the Lagrangian, while the “difficult” constraints are associated with dual variables. The dual objective gives us an optimization problem equivalent to $\mathbf{P}$. Denote the following dual problem as $\mathbf{D}$:

$$\min_{\lambda \in \Re^n} \Theta(\lambda) \tag{7}$$

To solve the problem $\mathbf{P}$, we solve the dual $\mathbf{D}$. Suppose $f(\mathbf{x})$ is linear in $\mathbf{x}$, denoted by $\mathbf{a}^T\mathbf{x}$. So, we have

$$\begin{aligned}
L(\lambda, \mathbf{x}) &= \mathbf{a}^T\mathbf{x} + \sum_{i=1}^{n} \lambda_i \left(\mathbf{c}_i^T\mathbf{x} - b_i\right) && (8)\\
&= \left(\mathbf{a}^T + \sum_i \lambda_i \mathbf{c}_i^T\right)\mathbf{x} - \sum_i \lambda_i b_i && (9)\\
&\equiv \hat{\mathbf{a}}_\lambda^T x - \hat{b} && (10)
\end{aligned}$$

Here, we use the notation $\hat{\mathbf{a}}$ to denote $\mathbf{a} + \sum_i \lambda_i \mathbf{c}_i$.

The dual objective Θ is defined as the maximum of the Lagrangian over all $\mathbf{x} \in X$. That is, we have

$$\begin{aligned}
\Theta(\lambda) &= \max_{x \in X} L(\lambda, \mathbf{x}) && (11)\\
&= \max_{\mathbf{x} \in X} \hat{\mathbf{a}}_\lambda^T \mathbf{x} - \hat{b} && (12)\\
&= \max_{\mathbf{x} \in X} \hat{\mathbf{a}}_\lambda^T \mathbf{x} && (13)
\end{aligned}$$

The last step has the same functional form as the problem $\mathbf{P'}$, which is computationally efficient to solve. Thus, we can compute the dual objective efficiently.

To solve the actual dual $\mathbf{D}$, we employ sub-gradient descent over Θ. The partial derivative of the function $\Theta(\lambda)$ with respect to λ_i is given by

$$\frac{\partial \Theta}{\partial \lambda_i} = \mathbf{c}_i^T \mathbf{x}^* - b_i, \tag{14}$$

$$\text{where, } \mathbf{x}^* = \arg\max_{\mathbf{x} \in X} \hat{\mathbf{a}}_\lambda^T \mathbf{x} \tag{15}$$

Using this gradient, we can define the algorithm (Algorithm 1) that optimizes the problem $\mathbf{P}$ using the solver for $\mathbf{P'}$ as a sub-routine.

In this algorithm, the λ's are updated only if at least one constraint is violated. Otherwise, the algorithm has found the optimal solution for $\mathbf{P}$ and returns the value.

Notes:

1. It is possible that even after T iterations of the gradient descent, the solution has not yet been found. In such a case, can something be said about the quality of the solution?

Algorithm 1 Using Lagrangian relaxation for solving a "hard" inference problem $\mathbf{P}$ using the solver for an "easier" problem $\mathbf{P'}$ as a sub-routine

$\lambda^{(0)} \leftarrow \mathbf{0}$
for $t = 1 \cdots T$ **do**
  $\mathbf{x}^* \leftarrow \arg\max_{\mathbf{x} \in X} \left(\mathbf{a} + \sum_i \lambda_i \mathbf{c}_i\right)^T \mathbf{x}$
  if $\mathbf{x}^*$ satisfies all the n constraints **then**
    return $\mathbf{x}^*$
  else
    for $i = 1 \cdots n$ **do**
      $\lambda_i^{(t)} \leftarrow \lambda_i^{(t-1)} - \alpha^{(t)} \left(\mathbf{c}_i^T \mathbf{x}^* - b_i\right)$
    end for
  end if
end for

2. The overall problem $\mathbf{P}$ asks for the solution in the space that is the intersection of the feasible space for $\mathbf{P'}$ and the additional constraints. The problem $\mathbf{P'}$ could hide the "integer" constraints over the inference variables. That is, the solution to $\mathbf{P'}$ could be a dynamic program such as maximum flow.

3. Even though this note considers equality constraints of the form $\mathbf{c}_i^T \mathbf{x} = b_i$, it is easy to extend these to inequality constraints. Doing so will introduce additional box constraints in the dual which can be dealt with in the gradient descent using a projection step.

References

[CC11] Y. Chang and M. Collins. Exact decoding of phrase-based translation models through lagrangian relaxation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 26–37, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[EG72] J. P. Evans and F. J. Gould. A Generalized Lagrange Multiplier Algorithm for Optimum or Near Optimum Production Scheduling. *Management Science*, 18(5):pp. 299–311, 1972.

[EI63] H. Everett III. Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources. *Operations Research*, 11(3):pp. 399–417, 1963.

[Geo10] A. M. Geoffrion. Lagrangian Relaxation for Integer Programming. In M. Junger, T. M. Leibling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50*

Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art. Springer-Verlag Berlin Heidelberg, 2010.

[Lem01] C. Lemaréchal. Lagrangian Relaxation. In *Computational Combinatorial Optimization*, pages 112–156, 2001.

[RC11] A.M. Rush and M. Collins. Exact decoding of syntactic translation models through lagrangian relaxation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 72–82, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.