Alternating Directions for Method of Multipliers (ADMM)

CS6464

Contents

- DUAL ASCENT & DUAL DECOMPOSITION
- METHOD OF MULTIPLIERS
- ALTERNATING DIRECTION METHOD OF MULTIPLIERS (ADMM)
- EXAMPLE
- CONCLUSION

Goals of ADMM

- Solve problems with very large number of features or training examples
- ADMM suitable for distributed convex optimization problems or large scale ML problems
- ADMM : decomposition-coordination procedure
- Blend benefits of dual decomposition and augmented lagrangian methods (method of multipliers) for constrained optimization problem

Let us say we are solving the following constrained problem:

$$\min f(\mathbf{x})$$

subject to

$$c_i(\mathbf{x}) = 0 \, \forall i \in I.$$

This problem can be solved as a series of unconstrained minimization problems. For reference, we first list the penalty method approach:

$$\min \Phi_k(\mathbf{x}) = f(\mathbf{x}) + \mu_k \sum_{i \in I} c_i(\mathbf{x})^2$$

The penalty method solves this problem, then at the next iteration it re-solves the problem using a larger value of μ_k (and using the old solution as the initial guess or "warm-start").

The augmented Lagrangian method uses the following unconstrained objective:

$$\min \Phi_k(\mathbf{x}) = f(\mathbf{x}) + \frac{\mu_k}{2} \sum_{i \in I} c_i(\mathbf{x})^2 - \sum_{i \in I} \lambda_i c_i(\mathbf{x})$$

and after each iteration, in addition to updating μ_k, the variable λ is also updated according to the rule

$$\lambda_i \leftarrow \lambda_i - \mu_k c_i(\mathbf{x}_k)$$

where $\mathbf{x}_k$ is the solution to the unconstrained problem at the *k*th step, i.e. $\mathbf{x}_k = \operatorname{argmin} \Phi_k(\mathbf{x})$

The variable λ is an estimate of the Lagrange multiplier, and the accuracy of this estimate improves at every step.

The strong Lagrangian principle: Lagrange duality [edit]

Given a nonlinear programming problem in standard form

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \leq 0,\ i \in \{1,\dots,m\} \\ &\qquad\qquad h_i(x) = 0,\ i \in \{1,\dots,p\} \end{aligned}$$

with the domain $\mathcal{D} \subset \mathbb{R}^n$ having non-empty interior, the *Lagrangian function* $\Lambda : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is defined as

$$\Lambda(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x).$$

The vectors λ and ν are called the *dual variables* or *Lagrange multiplier vectors* associated with the problem. The *Lagrange dual function* $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is defined as

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} \Lambda(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right).$$

The dual function g is concave, even when the initial problem is not convex, because it is a point-wise infimum of affine functions. The dual function yields lower bounds on the optimal value p^* of the initial problem; for any $\lambda \geq 0$ and any ν we have $g(\lambda, \nu) \leq p^*$.

If a constraint qualification such as Slater's condition holds and the original problem is convex, then we have strong duality, i.e. $d^* = \max_{\lambda \geq 0, \nu} g(\lambda, \nu) = \inf f_0 = p^*$

Convex problems [edit]

For a convex minimization problem with inequality constraints,

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m \end{aligned}$$

the Lagrangian dual problem is

$$\begin{aligned} \underset{u}{\text{maximize}} \quad & \inf_x \left(f(x) + \sum_{j=1}^{m} u_j g_j(x) \right) \\ \text{subject to} \quad & u_i \geq 0, \quad i = 1, \ldots, m \end{aligned}$$

where the objective function is the Lagrange dual function. Provided that the functions f and $g_1, \cdots, g_m$ are continuously differentiable, the infimum occurs where the gradient is equal to zero. The problem

$$\begin{aligned} \underset{x,u}{\text{maximize}} \quad & f(x) + \sum_{j=1}^{m} u_j g_j(x) \\ \text{subject to} \quad & \nabla f(x) + \sum_{j=1}^{m} u_j \nabla g_j(x) = 0 \\ & u_i \geq 0, \quad i = 1, \ldots, m \end{aligned}$$

is called the Wolfe dual problem. This problem may be difficult to deal with computationally, because the objective function is not concave in the joint variables (u, x). Also, the equality constraint $\nabla f(x) + \sum_{j=1}^{m} u_j \nabla g_j(x)$ is nonlinear in general, so the Wolfe dual problem is typically a nonconvex optimization problem. In any case, weak duality holds.[17]

Alternating direction method of multipliers [edit]

The alternating direction method of multipliers (ADMM) is a variant of the augmented Lagrangian scheme that uses partial updates for the dual variables. This method is often applied to solve problems such as

$$\min_x f(x) + g(x).$$

This is equivalent to the constrained problem

$$\min_{x,y} f(x) + g(y), \quad \text{subject to} \quad x = y.$$

Though this change may seem trivial, the problem can now be attacked using methods of constrained optimization (in particular, the augmented Lagrangian method), and the objective function is separable in *x* and *y*. The dual update requires solving a proximity function in *x* and *y* at the same time; the ADMM technique allows this problem to be solved approximately by first solving for *x* with *y* fixed, and then solving for *y* with *x* fixed. Rather than iterate until convergence (like the Jacobi method), the algorithm proceeds directly to updating the dual variable and then repeating the process. This is not equivalent to the exact minimization, but surprisingly, it can still be shown that this method converges to the right answer (under some assumptions). Because of this approximation, the algorithm is distinct from the pure augmented Lagrangian method.

Dual problem

- Consider a convex equality constrained optimization problem

$$\begin{aligned} minimize \quad & f(x) \\ subject\ to \quad & Ax = b \end{aligned}$$

- Lagrangian function: $L(x, y) = f(x) + y^T(Ax - b)$

  where x: primal variable

  y: the lagrangian variable/dual variable

- Dual function : $g(y) = inf_x\ L(x, y)$
- Dual problem : $maximize\ \ g(y)$

- recover $x^* = argmin_x\ L(x, y^*)$

  y is the optimal dual variable*

DUAL ASCENT

- Gradient ascent for the dual problem :

$$y^{k+1} = y^k + \alpha^k \Delta g(y^k)$$

k : iteration no, α :step size

- $\Delta g(y^k) = A\hat{x} - b$, where $\hat{x} = argmin_x\, L(x, y^k)$

- i.e, the dual ascent method consists of the following steps

$x^{k+1} = argmin_x\, L(x, y^k)$ // x-minimization step

$y^{k+1} = y^k + \alpha^k(Ax^{k+1} - b)$ //dual update step

DUAL DECOMPOSITION

- If f is separable to N subfunctions,

$$f(x) = \sum_{i=1}^{N} f_i(x_i)$$

where $x = (x_1, \ldots, x_N)$ and the variables $x_i \in R^{n_i}$ are subvectors of x.

Partitioning the matrix A, of size M x K (K=$\sum_{i=1}^{N} n_i$), conformably as $A = [A_1 \cdots A_N]$, where each A_i is a matrix of size M x n_i. So $Ax = \sum_{i=1}^{N} A_i x_i$,

- then L is separable :

$$L(x, y) = \sum_{i=1}^{N} (f_i(x_i) + y^T (A_i x_i - \frac{b}{N}))$$

- Dual decomposition has N x-minimization steps for each iteration

Dual Decomposition

- Dual descent is applied to N sub-functions and N separate x-minimizations will result

$$x_i^{k+1} = argmin_{x_i} L(x_i, y^k) \quad i=1,2..\ N$$

// x-minimization step

$$y^{k+1} = y^k + \alpha^k \left(\sum_{i=1}^{N} A_i x_i^{k+1} - b \right)$$ //dual update step

- scatter, update in parallel, gather
- solve a large problem by iteratively solving smaller subproblems in parallel.
- Dual variable update step provides coordination
- Works with a lot of assumptions and often slow

Augmented Lagrangian & Method of Multipliers

- A method to make dual ascent more robust
- Augmented Lagrangian:

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + \rho/2||Ax - b||_2^2$$

ρ is the penalty coefficient

- Method of Multipliers

$x^{k+1} = argmin_x\, L_\rho(x, y^k)$ // x-minimization step

$y^{k+1} = y^k + \rho(Ax^{k+1} - b)$ //dual update step,

// step length ρ

- Disadvantage: Can't do decomposition in x-minimization term because of the quadratic penalty term

Alternating direction method of multipliers (ADMM)

- A method with

  - good robustness of method of multipliers

  - which can support decomposition

- ADMM problem form

$$\begin{aligned} &minimize \quad f(x) + g(z) \\ &subject\ to \quad Ax + Bz = c \end{aligned}$$

- Augmented Lagrangian form:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\left||Ax + Bz - c|\right|_2^2$$

ADMM updates

- $$x^{k+1} = argmin_x \left(L_\rho \left(x, z^k, y^k \right) \right)$$

// x-minimization step

$$z^{k+1} = argmin_z \left(L_\rho \left(x^{k+1}, z, y^k \right) \right)$$

//z-minimization step

$$y^{k+1} = y^k + \rho \left(A x^{k+1} + B z^{k+1} - c \right)$$

//dual update step

Example

- Consider the optimization problem

$$minimize\ f(x) = \sum_{i=1}^{n} f_i(x)$$

- The problem is not separable because x is shared among all convex and closed functions $f_i(x)$
- To make the problem separable, introduce a set of local variables x_i and a global variable z such that the constraint becomes that all local variables should agree

$$minimize \sum_{i=1}^{n} f_i(x_i)$$
$$subject\ to\ \ x_i - z = 0,\ \ i = 1, \ldots . n$$

Augmented Lagrangian and ADMM updates

- Augmented Lagrangian:

$$L_\rho(x_1, \dots x_n, z, y_1, \dots y_n) = \sum_{i=1}^{n} (f_i(x_i) + y_i(x_i - z) + \rho/2||x_i - z||_2^2)$$

- ADMM updates

$$x_i^{k+1} := argmin_{x_i} \left[f_i(x_i) + y_i^k . \left(x_i - z^k\right) + \frac{\rho}{2} \left|\left| x_i - z^k \right|\right|_2^2 \right], \forall i$$

$$z^{k+1} := \frac{1}{n} \sum_{i=1}^{n} \left(x_i^{k+1} + \frac{1}{\rho} y_i^k \right)$$

$$y_i^{k+1} := y_i^k + \rho\left(x_i^{k+1} - z^{k+1}\right)$$

y_i: Lagrange variable
k: iteration number
ρ: step size

The *alternating direction method of multipliers* (ADMM) is a convex optimization algorithm dating back to the early 1980's [10, 11]; it has attracted renewed attention recently due to its applicability to various machine learning and image processing problems. In particular,

- It appears to perform reasonably well on these relatively recent applications, better than when adapted to traditional operations research problems such as minimum-cost network flows.
- The ADMM can take advantage of the structure of these problems, which involve optimizing sums of fairly simple but sometimes nonsmooth convex functions.
- Extremely high accuracy is not usually a requirement for these applications, reducing the impact of the ADMM's tendency toward slow "tail convergence".
- Depending on the application, it is often is relatively easy to implement the ADMM in a distributed-memory, parallel manner. This property is important for "big data" problems in which the entire problem dataset may not fit readily into the memory of a single processor.

The recent survey article [3] describes the ADMM from the perspective of machine learning applications; another, older survey is contained in the doctoral thesis [5].

References

- Boyd, Stephen, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers." *Foundations and Trends® in Machine Learning* 3.1 (2011): 1-122.

[10] M. Fortin and R. Glowinski. On decomposition-coordination methods using an augmented Lagrangian. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems.* North-Holland: Amsterdam, 1983.

[11] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems.* North-Holland: Amsterdam, 1983.

[5] J. Eckstein. *Splitting methods for monotone operators with applications to parallel optimization.* PhD thesis, MIT, 1989.