Fully Distributed Algorithms for Convex Optimization Problems

Damon Mosk-Aoyama[1], Tim Roughgarden[1], and Devavrat Shah[2]

[1] Department of Computer Science, Stanford University
[2] Department of Electrical Engineering and Computer Science, MIT

Abstract. We describe a distributed algorithm for convex constrained optimization in networks without any consistent naming infrastructure. The algorithm produces an approximately feasible and near-optimal solution in time polynomial in the network size, the inverse of the permitted error, and a measure of curvature variation in the dual optimization problem. It blends, in a novel way, gossip-based information spreading, iterative gradient ascent, and the barrier method from the design of interior-point algorithms.

1 Problem Description

We consider an undirected graph $G = (V, E)$, with $V = \{1, \ldots, n\}$, in which each node i has a non-negative decision variable x_i. Our goal is to solve convex optimization problems of the following form.

$$\begin{aligned} \text{minimize} \quad & f(x) = \sum_{i=1}^{n} f_i(x_i) \qquad (1) \\ \text{subject to} \quad & Ax = b \\ & x_i \geq 0, \quad i = 1, \ldots, n \end{aligned}$$

We assume that each function f_i is twice differentiable and strictly convex, with $\lim_{x_i \downarrow 0} f_i'(x_i) < \infty$ and $\lim_{x_i \uparrow \infty} f_i'(x_i) = \infty$. The elements of the $m \times n$ matrix A and the vector $b \in \mathbf{R}^m$ are non-negative.

Although we seek a solution to the primal problem (1), instead of directly enforcing the non-negativity constraints, we introduce a logarithmic barrier. For a parameter $\theta > 0$, we replace the objective function in (1) by $\sum_{i=1}^{n}(f_i(x_i) - \theta \ln x_i)$, and we remove the constraints $x_i \geq 0$ to obtain a primal barrier problem. The associated Lagrange dual problem is the following optimization problem, for which a feasible solution is any vector $\lambda \in \mathbf{R}^m$.

$$\text{maximize} \quad g_\theta(\lambda) = -b^T \lambda + \sum_{i=1}^{n} \inf_{x_i > 0} \left(f_i(x_i) - \theta \ln x_i + a_i^T \lambda x_i \right) \qquad (2)$$

We assume that the primal barrier problem is feasible; that is, there exists a vector $x \in \mathbf{R}^n$ with $x_i > 0$ such that $Ax = b$. As a result, the optimal value of the primal barrier problem is finite, and Slater's condition implies that the dual problem (2) has the same optimal value, and there exists a dual solution λ^* that achieves this optimal value. Since (2) is an unconstrained maximization problem with a strictly concave objective function, the optimal solution λ^* is unique.

2 Algorithm Description

Our approach to solving the primal problem (1) is to apply gradient ascent for the dual barrier problem (2). The algorithm generates a sequence of feasible solutions $\lambda^0, \lambda^1, \lambda^2, \ldots$ for (2), where λ^0 is an initial vector provided as input. To update λ^{k-1} to λ^k in an iteration k, the algorithm uses the gradient $\nabla g_\theta\left(\lambda^{k-1}\right)$ to determine the direction of the difference $\lambda^k - \lambda^{k-1}$.

For a dual solution λ, the gradient $\nabla g_\theta(\lambda)$ is given by $\nabla g_\theta(\lambda) = Ax(\lambda) - b$, where the vector $x(\lambda) \in \mathbf{R}^n$ is defined by $x_i(\lambda) = h_i^{-1}\left(-a_i^T \lambda\right)$. Here, the function h_i is defined as $h_i(x_i) = f_i'(x_i) - \theta/x_i$, and a_i is the ith column of the matrix A. To compute a component $j \in \{1, \ldots, m\}$ of the gradient, the nodes estimate the sum $\sum_{i=1}^n A_{ji} x_i(\lambda)$ using a distributed summation algorithm by Mosk-Aoyama and Shah (PODC, 2006). This is a randomized algorithm that takes an error parameter ϵ_1 as input, and is said to succeed if the output value it produces is within a factor of $1 \pm \epsilon_1$ of the actual sum.

Based on the formula above for $\nabla g_\theta(\lambda)$, the norm of the gradient measures how far the vector $x(\lambda)$ is from satisfying the equality constraints in (1). The nodes continue to execute iterations of gradient ascent until the ℓ_2-norm $\|\nabla g_\theta(\lambda)\|$ goes below a threshold determined by an input error parameter, where λ is the current dual solution. At this point, the algorithm terminates with the vector $x(\lambda)$ as the output. The number of iterations required for the algorithm to terminate can be bounded in terms of the following quantity.

$$R = \frac{\left(\max_{i=1,\ldots,n} \max_{\lambda \in B(\lambda^*, \|\lambda^0 - \lambda^*\|)} \left(h_i^{-1}\right)' \left(-a_i^T \lambda\right)\right) \sigma_{\max}\left(A^T\right)^2}{\left(\min_{i=1,\ldots,n} \min_{\lambda \in B(\lambda^*, \|\lambda^0 - \lambda^*\|)} \left(h_i^{-1}\right)' \left(-a_i^T \lambda\right)\right) \sigma_{\min}\left(A^T\right)^2} \tag{3}$$

Here, $B(\lambda^*, r) = \{\lambda \mid \|\lambda - \lambda^*\| \le r\}$, and $\sigma_{\min}\left(A^T\right)$ and $\sigma_{\max}\left(A^T\right)$ denote the smallest and largest singular values, respectively, of the matrix A^T.

Given an error parameter ϵ as input, the nodes set the parameters used in the gradient ascent so that, provided that every invocation of the summation subroutine succeeds, the number of iterations executed is bounded as follows.

Theorem 1. *After $O\left(R^2 \log\left(\|Ax\left(\lambda^0\right) - b\| / (\epsilon \|b\|)\right)\right)$ iterations, the gradient ascent terminates with a solution $x(\lambda)$ such that $\|Ax(\lambda) - b\| \le \epsilon \|b\|$. The objective function value of the solution satisfies $f(x(\lambda)) \le \mathrm{OPT} + \epsilon \|b\| \|\lambda\| + n\theta$, where OPT is the optimal value of the primal program (1).*

Some of the parameters in the gradient ascent are set by the nodes using quantities, such as R, whose values would be unknown to the nodes. A complete algorithm for computing an approximate solution to the primal problem (1) with high probability can be obtained by adding an outer loop to the algorithm that executes gradient ascent for different possible values of these quantities.