Fast-Lipschitz Optimization

DREAM Seminar Series
University of California at Berkeley
September 11, 2012

Carlo Fischione

ACCESS Linnaeus Center, Electrical Engineering
KTH Royal Institute of Technology
Stockholm, Sweden
web: http://www.ee.kth.se/~carlofi
e-mail: carlofi@kth.se
Optimization is pervasive over networks

- Parallel computing
- Environmental monitoring
- Intelligent transportation systems
- Smart grids
- Smart buildings
- Industrial control
Optimization over networks

- Optimization needs fast solver algorithms of low complexity
  - Time-varying networks, little time to compute solution
  - Distributed computations
  - E.g., networks of parallel processors, cross layer networking, distributed detection, estimation, content distribution, ...

- Parallel and distributed computation
  - Fundamental theory for optimization over networks
  - Drawback over energy-constrained wireless networks: the cost for communication not considered

- An alternative theory is needed
  - In a number of cases, Fast-Lipschitz optimization
Outline

• Motivating example: distributed detection

• Definition of Fast-Lipschitz optimization

• Computation of the optimal solution

• Problems in canonical form

• Examples

• Conclusions
Distributed binary detection

\[ \Gamma_i(s) = w_i(s) \quad \text{if} \quad H_0 \]
\[ \Gamma_i(s) = E + w_i(s) \quad \text{if} \quad H_1 \]

measurements at node \( i \)

\[ T_i = \frac{1}{S} \sum_{s=1}^{S} \Gamma_i(s) \leq x_i \]

hypothesis testing with \( S \) measurements and threshold \( x_i \)

\[ P_{fa}^{(i)}(x_i) = \Pr[T_i > x_i | H_0] \quad \text{probability of false alarm} \]
\[ P_{md}^{(i)}(x_i) = \Pr[T_i \leq x_i | H_1] \quad \text{probability of misdetection} \]

- A threshold minimizing the prob. of false alarm maximizes the prob. of misdetection.
- How to choose optimally the thresholds when nodes exchange opinions?
Threshold optimization in distributed detection

\[
\min_{x} \sum_{i=1}^{n} P_{fa}^{(i)}(x_i)
\]

s.t. \( \sum_{j=1}^{n} b_{i,j} P_{md}^{(j)}(x_j) \leq c_i, \quad i = 1, \ldots, n \),

\[ 0 \leq x \leq E1. \]

- How to solve the problem by distributed operations among the nodes?
- The problem is convex
  - Lagrangian methods (interior point) can be applied
  - Drawback: too many message passing (Lagrangian multipliers) among nodes to compute iteratively the optimal solution
- An alternative method: Fast-Lipschitz optimization

Outline

• Motivating example: distributed detection

• Definition of Fast-Lipschitz optimization

• Computation of the optimal solution

• Problems in canonical form

• Examples

• Conclusions
The Fast-Lipschitz optimization

\[ \max_{\mathbf{x}} \quad f_0(\mathbf{x}) \]

s.t. \[ x_i \leq f_i(\mathbf{x}), \quad i = 1, \ldots, l \]

\[ x_i = h_i(\mathbf{x}), \quad i = l + 1, \ldots, n \]

\[ \mathbf{x} \in \mathcal{D}, \]

\[ f_0(\mathbf{x}) : \mathcal{D} \to \mathbb{R}, \]

\[ f_i(\mathbf{x}) : \mathcal{D} \to \mathbb{R}, \quad i = 1, \ldots, l \]

\[ h_i(\mathbf{x}) : \mathcal{D} \to \mathbb{R}, \quad i = l + 1, \ldots, n \]

\[ \mathcal{D} \subset \mathbb{R}^n \quad \text{nonempty compact set containing the vertexes of the constraints} \]
Computation of the solution

\[
\begin{align*}
\max_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\
\text{s.t.} \quad & x_i \leq f_i(\mathbf{x}), \quad i = 1, \ldots, l \\
& x_i = h_i(\mathbf{x}), \quad i = l + 1, \ldots, n \\
& \mathbf{x} \in \mathcal{D},
\end{align*}
\]

- Centralized optimization
  - Problem solved by a central processor

- Distributed optimization
  - Decision variables and constraints are associated to nodes that cooperate to compute the solution in parallel
Pareto Optimal Solution

Definition: Consider the following set

$$\mathcal{A} = \{ x \in \mathcal{D} : x_i \leq f_i(x), \ i = 1, \ldots, l, \quad x_i = h_i(x), \ i = l + 1, \ldots, n \},$$

and let $\mathcal{B} \in \mathbb{R}^l$ be the image set of $f_0(x)$, namely $f_0(x) : \mathcal{A} \rightarrow \mathcal{B}$. Then, we make the natural assumption that the set $\mathcal{B}$ is partially ordered in a natural way, namely if $x, y \in \mathcal{B}$ then $x \succeq y$ if $x_i \geq y_i \ \forall i$ (e.g., $\mathbb{R}^l_+$ is the ordering cone).

Definition (Pareto Optimal): A vector $x^*$ is called a Pareto optimal (or an Edgeworth-Pareto optimal) point if there is no $x \in \mathcal{A}$ such that $f_0(x) \succeq f_0(x^*)$ (i.e., if $f_0(x^*)$ is the maximal element of the set $\mathcal{B}$ with respect to the natural partial ordering defined by the cone $\mathbb{R}^l_+$).
Notation

$$F(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_n(x) \end{bmatrix}$$

$$F_i(x) : \mathbb{R}^n \to \mathbb{R} \ \forall i$$

$$\nabla F(x) = \begin{bmatrix} \frac{dF_1(x)}{dx_1} & \frac{dF_2(x)}{dx_1} & \ldots & \frac{dF_n(x)}{dx_1} \\ \frac{dF_1(x)}{dx_2} & \frac{dF_2(x)}{dx_2} & \ldots & \frac{dF_n(x)}{dx_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dF_1(x)}{dx_n} & \frac{dF_2(x)}{dx_n} & \ldots & \frac{dF_n(x)}{dx_n} \end{bmatrix}$$

$$|\nabla F(x)|_\infty = \max_j \sum_{i=1}^n \left| \frac{dF_i(x)}{dx_j} \right|$$

Norm infinity: sum along a row

$$|\nabla F(x)|_1 = \max_j \sum_{i=1}^n \left| \frac{dF_i(x)}{dx_i} \right|$$

Norm 1: sum along a column
Now that we have introduced basic notation and concepts, we give some conditions for which a problem is Fast-Lipschitz.
Qualifying conditions

1.a \( \nabla f_0(x) \succ 0 \), i.e., \( f_0(x) \) is strictly increasing,
1.b \( |\nabla F(x)|_\infty < 1 \),

and either

2.a \( \nabla_j F_i(x) \geq 0 \) \( \forall i, j \),
or
3.a \( \nabla_i f_0(x) = \nabla_j f_0(x) \),
3.b \( \nabla_j F_i(x) \leq 0 \) \( \forall i, j \),
3.c \( |\nabla F(x)|_1 < 1 \),
or
4.a \( f_0(x) \in \mathbb{R} \),
4.b \( |\nabla F(x)|_1 \leq \frac{\delta}{\delta + \Delta} \),

\[ \delta = \min_{i, x \in \mathcal{D}} \nabla_i f_0(x), \]
\[ \Delta = \max_{i, x \in \mathcal{D}} \nabla_i f_0(x). \]

\[ \max_x f_0(x) \]
\[ \text{s.t.} \ x_i \leq f_i(x), \ i = 1, \ldots, l \]
\[ x_i = h_i(x), \ i = l + 1, \ldots, n \]
\[ x \in \mathcal{D}, \]

\( f(x) = [f_1(x), f_2(x), \ldots, f_l(x)]^T \)
\( h(x) = [h_{l+1}(x), h_{l+2}(x), \ldots, h_n(x)]^T \)
\( F(x) = [F_i(x)] = [f(x)^T \ h(x)^T]^T \)

Functions may be non-convex
Outline

• Motivating example: distributed detection

• Definition of Fast-Lipschitz optimization

• Computation of the optimal solution

• Problems in canonical form

• Examples

• Conclusions
Optimal Solution

Theorem: Let an F-Lipschitz optimization problem be feasible. Then, the problem admits a unique optimum $x^* \in \mathcal{D}$ given by the solution of the set of equations

$$\begin{align*}
x_i^* &= f_i(x^*) \quad i = 1, \ldots, l \\
x_i^* &= h_i(x^*) \quad i = l + 1, \ldots, n.
\end{align*}$$

- The Pareto optimal solution is just given by a set of (in general non-linear) equations.
- Solving a set of equations is much easier than solving an optimization problem by traditional Lagrangian methods!
Lagrangian methods

- Let’s have a closer look at the Lagrangian methods, which are normally used to solve optimization problems.

- Lagrangian methods are the essential to solve, for example, convex problems.
Lagrangian methods

G. L. Lagrange, 1736-1813

"The methods I set forth require neither constructions nor geometric or mechanical
c onsiderations. They require only algebraic operations subject to a systematic and
uniform course"
Theorem: Consider a feasible F-Lipschitz problem. Then, the KKT conditions are necessary and sufficient.

- KKT conditions:

\[ x_i - f_i(x^*) \leq 0 \quad i = 1, \ldots, l \]
\[ x_i - h_i(x^*) = 0 \quad i = l + 1, \ldots, n \]
\[ \lambda_i^* \geq 0 \quad i = 1, \ldots, n \]
\[ \lambda_i^* f_i(x^*) = 0 \quad i = 1, \ldots, n \]
\[ \nabla_x L(x^*, \lambda^*) = 0, \]

\[ L(x, \lambda) = -f_0(x) + \sum_{i=1}^{l} \lambda_i (x_i - f_i(x)) + \sum_{i=l+1}^{n} \lambda_i (x_i - h_i(x)) \]

\[ x(k+1) = x(k) - \beta \nabla_x L(x(k), \lambda(k)) \]
\[ \lambda(k+1) = \lambda(k) - \beta \nabla_{\lambda} L(x(k), \lambda(k)) \]

Lagrangian methods to compute the solution
Lagrangian methods

\[ L(x, \lambda) = -f_0(x) + \sum_{i=1}^{l} \lambda_i (x_i - f_i(x)) + \sum_{i=l+1}^{n} \lambda_i (x_i - h_i(x)) \]

\[
\begin{align*}
x(k + 1) &= x(k) - \beta \nabla_x L(x(k), \lambda(k)) \\
\lambda(k + 1) &= \lambda(k) - \beta \nabla_{\lambda} L(x(k), \lambda(k))
\end{align*}
\]

- Lagrangian methods need
  1. a central computation of the Lagrangian function
  2. an endless collect-and-broadcast iterative message passing of primal and dual variables

- Fast-Lipschitz methods avoid the central computation and substantially reduce the collect-and-broadcast procedure

\[
\begin{align*}
\max_x & f_0(x) \\
\text{s.t.} & \quad x_i \leq f_i(x), \quad i = 1, \ldots, l \\
& \quad x_i = h_i(x), \quad i = l + 1, \ldots, n \\
& \quad x \in \mathcal{D}.
\end{align*}
\]
Fast-Lipschitz optimization problems can be convex, geometric, quadratic, interference-function,...
Let us see how a Fast-Lipschitz problem is solved without Lagrangian methods
Centralized optimization

- The optimal solution is given by iterative methods to solve systems of non-linear equations (e.g., Newton methods)

\[
x(k+1) = x(k) - \beta (x(k) - F(x(k)))
\]

\[
f(x) = [f_1(x), f_2(x), \ldots, f_l(x)]^T
\]

\[
h(x) = [h_{l+1}(x), h_{l+2}(x), \ldots, h_n(x)]^T
\]

\[
F(x) = [f(x)^T h(x)^T]^T
\]

\[\beta\] is a matrix to ensure and maximize convergence speed

- Many other methods are available, e.g., second-order methods
Distributed optimization

\[
\begin{align*}
\max_x & \quad f_0(x) \\
\text{s.t.} & \quad x_i \leq f_i(x), \quad i = 1, \ldots, l \\
& \quad x_i = h_i(x), \quad i = l + 1, \ldots, n \\
& \quad x \in \mathcal{D}.
\end{align*}
\]

**Proposition:** Let \( x(0) \in \mathcal{X} \) be an initial guess of the optimal solution to a feasible F-Lipschitz problem. Let \( x^i(k) = [x_1(\tau^i_1(k)), x_2(\tau^i_2(k)), \ldots, x_n^i(\tau^i_n(k))] \) the vector of decision variables available at node \( i \) at time \( k \in \mathbb{N}_+ \), where \( \tau^i_j(k) \) is the delay with which the decision variable of node \( j \) is communicated to node \( i \). Then, the following iterative algorithm converges to the optimal solution:

\[
\begin{align*}
x^i(k+1) & = [f_i(x^i(k))]^\mathcal{D} \quad i = 1, \ldots, l \\
x^i(k+1) & = h_i(x^i(k)) \quad i = l + 1, \ldots, n
\end{align*}
\]

where \( k \in \mathbb{N}_+ \) is an integer associated to the iterations.
Outline

• Motivating example: distributed detection

• Definition of Fast-Lipschitz optimization

• Computation of the optimal solution

• Problems in canonical form

• Examples

• Conclusions
Problems in canonical form

Canonical form
Bertsekas, *Non Linear Programming*, 2004

\[
\begin{align*}
\min_x & \quad g_0(x) \\
\text{s.t.} & \quad g_i(x) \leq 0, \quad i = 1, \ldots, l \\
& \quad p_i(x) = 0, \quad i = l + 1, \ldots, n \\
& \quad x \in \mathcal{D},
\end{align*}
\]

\[
\begin{align*}
\max_x & \quad f_0(x) \\
\text{s.t.} & \quad x_i \leq f_i(x), \quad i = 1, \ldots, l, \\
& \quad x_i = h_i(x), \quad i = l + 1, \ldots, n, \\
& \quad x \in \mathcal{D},
\end{align*}
\]

Fast-Lipschitz form

\[
\begin{align*}
f_0(x) &= -g_0(x), \\
f_i(x) &= x_i - \gamma_i g_i(x), \quad \gamma_i > 0 \\
h_i(x) &= x_i - \mu_i p_i(x), \quad \mu_i \in \mathbb{R}
\end{align*}
\]
Theorem: Consider the optimization problem in canonical form. Suppose that $\forall \ x \in \mathcal{D}$

1.a $\nabla g_0(x) < 0$,

1.b $\nabla_i G_i(x) > 0 \ \forall i$,

and either

2.a $\nabla_j G_i(x) \leq 0 \ \forall j \neq i$,

2.b $\nabla_i G_i(x) > \sum_{j \neq i} |\nabla_i G_j(x)| \ \forall i$,

or

3.a $g_0(x) = -c1^T x \ \ c \in \mathbb{R}^+$,

3.b $\nabla_j G_i(x) \geq 0 \ \forall j \neq i$,

3.c $\nabla_i G_i(x) > \sum_{j \neq i} |\nabla_i G_j(x)| \ \forall i$,

or

4.a $g_0(x) \in \mathbb{R}$,

4.b $\frac{\delta}{\delta + \Delta} \nabla_i G_i(x) > \sum_{j \neq i} |\nabla_i G_j(x)| \ \forall i$.

Then, the problem is F-Lipschitz.
Fast-Lipschitz Matlab Toolbox

Outline

• Motivating example: distributed detection

• Definition of Fast-Lipschitz optimization

• Computation of the optimal solution

• Problems in canonical form

• Examples

• Conclusions
Example 1: from canonical to Fast-Lipschitz

\[
\min_{x,y} \quad a e^{-x_1} + b e^{-x_2} \quad a > 0, \ b > 0
\]
\[
\text{s.t.} \quad x_1 - 0.5x_2 - 1 \leq 0
\]
\[
- x_1 + 2x_2 \leq 0
\]
\[
x_1 \geq 0, \ x_2 \geq 0,
\]

- The problem is both convex and Fast-Lipschitz:

\[
\nabla_x (x - 0.5y - 1) = 1 > |\nabla_y (x - 0.5y - 1)| = 0.5, \quad \text{Off-diagonal monotonicity}
\]
\[
\nabla_y (-x + 2y) = 2 > |\nabla_x (-x + 2y)| = 1, \quad \text{Diagonal dominance}
\]

- The optimal solution is given by the constraints at the equality, trivially

\[
x_1 - 0.5x_2 - 1 = 0 \quad x_1 = \frac{4}{3}
\]
\[
-x_1 + 2x_2 = 0, \quad x_2 = \frac{2}{3}
\]
Example 2: hidden Fast-Lipschitz

- **Non Fast-Lipschitz**

  \[
  \min_{x,y,z} \quad ae^{-x} + be^{-y} + ce^{z} \\
  \text{s.t.} \quad 2x - 0.5y + z + 3 \leq 0 \\
  \quad -x + 2y - z^{-1} + 1 \leq 0 \\
  \quad -3x - y + z^{-2} + 2 \leq 0 \\
  \quad x_{\min} \leq x \leq x_{\max}, \quad y_{\min} \leq y \leq y_{\max}, \quad z_{\min} \leq z \leq z_{\max},
  \]

- **Simple variable transformation**, \( t = z^{-1} \), gives a Fast-Lipschitz form

  \[
  \max_{x,y,t} \quad -ae^{-x} - be^{-y} - ce^{-t} \\
  \text{s.t.} \quad 2x - 0.5y + t^{-1} + 3 \leq 0 \\
  \quad -0.5x + 2y - t + 1 \leq 0 \\
  \quad -0.5x - y + t^2 + 2 \leq 0 \\
  \quad x_{\min} \leq x \leq x_{\max}, \quad y_{\min} \leq y \leq y_{\max}, \quad 1/z_{\max} \leq t \leq 1/z_{\min}
  \]
Threshold optimization in distributed detection

\[
\min_{\mathbf{x}} \sum_{i=1}^{n} P_{fa}^{(i)}(x_i)
\]
\[
\text{s.t. } \sum_{j=1}^{n} b_{i,j} P_{md}^{(j)}(x_j) \leq c_i, \quad i = 1, \ldots, n,
\]
\[
0 \leq \mathbf{x} \leq E1.
\]

- How to solve the problem by parallel and distributed operations among the nodes?
- The problem is convex
  - Lagrangian methods (interior point methods) could be applied
  - Drowback: too many message passing (Lagrangian multipliers) among nodes to compute iteratively the optimal solution
- An alternative method: F-Lipschitz optimization
Distributed detection: Fast-Lipschitz vs Lagrangian methods

\[
\begin{align*}
\min_x & \quad \sum_{i=1}^{n} P_{fa}^{(i)}(x_i) \\
\text{s.t.} & \quad \sum_{j=1}^{n} b_{i,j} P_{md}^{(j)}(x_j) \leq c_i, \quad i = 1, \ldots, n, \\
& \quad 0 \leq x \leq E1.
\end{align*}
\]

10 nodes network

Number of iterations

Number of function evaluations

Fast-Lipschitz

Lagrangian methods

(interior point)
Fast-Lipschitz optimization: a class of problems for which all the constraints are active at the optimum

Optimum: the solution to the set of equations given by the constraints

No Lagrangian methods, which are computationally expensive, particularly on wireless networks
Conclusions

- Existing methods for optimization over networks are too expensive

- Proposed the Fast-Lipschitz optimization
  - Application to distributed detection, many other cases

- Fast-Lipschitz optimization is a panacea for many cases, but still there is a lack of a theory for fast parallel and distributed computations

- How to generalize it for
  - static optimization?
  - dynamic optimization?
  - stochastic optimization?
  - game theoretical extensions?
Selected bibliography


