Convex optimization

Javier Peña
Carnegie Mellon University

Universidad de los Andes
Bogotá, Colombia
September 2014

Convex optimization

Problem of the form

$$\min_x \quad f(x)$$
$$x \in Q,$$

where

- $Q \subseteq \mathbb{R}^n$ convex set:

$$x, y \in Q, \ \lambda \in [0,1] \Rightarrow \lambda x + (1-\lambda) y \in Q,$$

- $f : Q \to \mathbb{R}$ convex function:

$$\text{epigraph}(f) = \{(x,t) \in \mathbb{R}^{n+1} : x \in Q, t \geq f(x)\} \ \text{ convex set.}$$

Special cases

Linear programming

$$\begin{aligned} \min_y \quad & \langle c, y \rangle \\ & \langle a_i, y \rangle - b_i \geq 0,\ i = 1, \dots, n. \end{aligned}$$

Semidefinite programming

$$\begin{aligned} \min_y \quad & \langle c, y \rangle \\ & \textstyle\sum_{j=1}^m A_j y_j - B \succeq 0. \end{aligned}$$

Second-order cone programming

$$\begin{aligned} \min_y \quad & \langle c, y \rangle \\ & \langle a_i, y \rangle - b_i \geq \| A_i y - d_i \|_2,\ i = 1, \dots, r. \end{aligned}$$

Agenda

- Applications
- Algorithms
- Open problems

Applications

Classification

Classification data

$\mathcal{D} = \{(x_1, \ell_1), \ldots, (x_n, \ell_n)\}$, with $x_i \in \mathbb{R}^d, \ \ell_i \in \{-1, 1\}$.

Linear classification

Find $(\beta_0, \beta) \in \mathbb{R}^{d+1}$ such that for $i = 1, \ldots, n$

$$\operatorname{sgn}(\beta_0 + \langle \beta, x_i \rangle) = \ell_i \Leftrightarrow \ell_i(\beta_0 + \langle x_i, \beta \rangle) > 0.$$

Support vector machines

Find linear classifier with largest margin

$$\begin{aligned} \min_{\beta_0, \beta} \quad & \|\beta\|_2 \\ & \ell_i(\langle x_i, \beta \rangle + \beta_0) \geq 1, \ i = 1, \ldots, n \end{aligned}$$

Regression

Regression data

$\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, with $x_i \in \mathbb{R}^d,\ y_i \in \mathbb{R}$.

Linear regression

Find $\beta \in \mathbb{R}^d$ that minimizes *training error:*

$$\min_\beta \sum_{i=1}^n (\beta_0 + \langle \beta, x_i \rangle - y_i)^2 \ \Leftrightarrow\ \min_\beta \|X\beta - y\|_2^2$$

$$X := \begin{bmatrix} \mathbf{1} & x_1^\top \\ \vdots & \vdots \\ 1 & x_n^\top \end{bmatrix},\ y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Sparse regression

High dimensional regression

$\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, large d, e.g., $d > n$.
Want $\beta \in \mathbb{R}^d$ *sparse*:

$$\min_{\beta} \left(\|X\beta - y\|_2^2 + \lambda \cdot \|\beta\|_0 \right)$$

$\|\beta\|_0 := |\{i : \beta_i \neq 0\}|$.

Lasso regression (Tibshirani, 1996)

The above problem is computationally intractable. Use instead

$$\min_{\beta} \left(\|X\beta - y\|_2^2 + \lambda \cdot \|\beta\|_1 \right).$$

Extensions

Group lasso, fused lasso, and others.

Compressive sensing

Raw 3MB jpeg versus a compressed 0.3MB version.

Question

If an image is compressible, can it be acquired efficiently?

Compressive sensing

Compressibility corresponds to sparsity in a suitable representation.

Restatement of the above question:

Question

Can we recover a sparse vector $\bar{x} \in \mathbb{R}^n$ from $m \ll n$ linear measurements

$$b_k = \langle a_k, \bar{x} \rangle,\ k = 1, \ldots, m \Leftrightarrow b = A\bar{x}.$$

Example (group testing)

Suppose only one component of $\bar{x}$ is different from zero.
Then $\log_2 n$ measurements or fewer suffice to find $\bar{x}$.

Compressive sensing via linear programming

Possible approach to recover sparse $\bar{x}$

Take $m \ll n$ measurements $b = A\bar{x}$ and solve

$$\min_x \quad \|x\|_0 \qquad Ax = b.$$

The above is computationally intractable. Use instead

$$\min_x \quad \|x\|_1 \qquad Ax = b.$$

Theorem (Candès & Tao, 2005)

If $m \gtrsim s \cdot \log n$ and A is suitably chosen. Then the ℓ_1-minimization problem recovers $\bar{x}$ with high probability.

Matrix completion

Problem

Assume $M \in \mathbb{R}^{n \times n}$ has low rank and we observe *some* entries of M. Can we recover M?

Possible approach to recover low rank M

Assume we observe entries in $\Omega \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$. Solve

$$\begin{aligned} \min_{X} \quad & \text{rank}(X) \\ & X_{ij} = M_{ij},\ (i,j) \in \Omega. \end{aligned}$$

Rank-minimization is computationally intractable.

Matrix completion via semidefinite programming

Fazel, 2001: Use instead

$$\begin{aligned} \min_X \quad & \|X\|_* \\ & X_{ij} = M_{ij},\ (i,j) \in \Omega. \end{aligned}$$

Here $\|\cdot\|_*$ is the *nuclear norm:*

$$\|X\|_* := \sum_{i=1}^{n} \sigma_i(X).$$

Theorem (Candès & Recht, 2010)

Assume rank$(M) = r$ *and* Ω *random,* $|\Omega| \geq C\mu r(1+\beta)\log^2 n$*. Then the nuclear norm minimization problem recovers* M *with with high probability.*

Algorithms

Late 20th century: interior-point methods

To solve

$$\min_x \quad \langle c, x\rangle \qquad x \in Q.$$

Trace path $\{x(\mu) : \mu > 0\}$, where $x(\mu)$ minimizes

$$F_\mu(x) := \langle c, x\rangle + \mu \cdot f(x).$$

Here $f : Q \to \mathbb{R}$ a suitable *barrier* function for Q.

Some barrier functions

Q	f
$\{y : \langle a_i, y\rangle - b_i \geq 0, i = 1, \dots, n\}$	$-\sum_{i=1}^n \log(\langle a_i, y\rangle - b_i)$
$\{y : \sum_{i=1}^m A_j y_j - B \succeq 0\}$	$-\log\det\left(\sum_{j=1}^m A_j y_j - B\right)$

Interior-point methods

Recall $F_\mu(x) = \langle c, x\rangle + \mu \cdot f(x)$ and $f : Q \to \mathbb{R}$ barrier function.

Template of interior-point method

- pick $\mu_0 > 0$ and $x_0 \approx x(\mu_0)$
- for $t = 0, 1, 2, \ldots$

  pick $\mu_{t+1} < \mu_t$

  $x_{t+1} := x_t - [F''_{\mu_{t+1}}(x_t)]^{-1} F'_{\mu_{t+1}}(x_t)$

  end for

The above can be done so that $x_t \to x^*$, where x^* solves

$$\begin{aligned} \min_x \quad & \langle c, x\rangle \\ & x \in Q. \end{aligned}$$

Interior-point methods

Features

- Superb theoretical properties.
- Numerical performance far better than what theory states.
- Excellent accuracy.
- Commercial and open-source implementations.

Limitations

- Barrier function for entire constraint set.
- Solve a system of equations (Newton's step) at each iteration.
- Numerically challenged for very large or dense problems.
- Often inadequate for above applications.

Early 21th century: algorithms with simpler iterations

Tradeoff the above features vs limitations.

In many applications modest accuracy is fine.

Interior-point methods

Need barrier function for *entire* constraint set, *second-order* information (gradient & Hessian), and solve systems of equations.

Simpler algorithms

Use less information about the problem. Avoid costly operations.

Convex feasibility problem

Assume $Q \subseteq \mathbb{R}^m$ is a convex set and consider the problem

$$\text{Find } \; y \in Q.$$

- Any convex optimization problem can be recast this way.
- Difficulty depends on how Q is described.
- Assume a *separation oracle* for $Q \subseteq \mathbb{R}^m$ is available.

Separation oracle for Q

Given $y \in \mathbb{R}^m$, verify $y \in Q$ or generate $0 \neq a \in \mathbb{R}^m$ such that

$$\langle a, y \rangle < \langle a, v \rangle, \; \forall v \in Q.$$

Examples

- Linear inequalities: $a_i \in \mathbb{R}^m, b_i \in \mathbb{R},\ i = 1, \ldots, n$

$$Q = \{y \in \mathbb{R}^m : \langle a_i, y\rangle - b_i \geq 0,\ i = 1, \ldots, n\}.$$

Oracle: Given y, check each $\langle a_i, y\rangle - b_i \geq 0$.

- Linear matrix inequalities: $B, A_j \in \mathbb{R}^{n\times n}$, $j = 1, \ldots, m$ symmetric,

$$Q = \left\{ y \in \mathbb{R}^m : \sum_{j=1}^{m} A_j y_j - B \succeq 0 \right\}.$$

Oracle: Given y, check $\sum_{j=1}^{m} A_j y_j - B \succeq 0$. If this fails, get $u \neq 0$ such that

$$\sum_{j=1}^{m} \langle u, A_j u\rangle y_j < \langle u, Bu\rangle \leq \sum_{j=1}^{m} \langle u, A_j u\rangle v_j,\ \forall v \in Q.$$

Relaxation method (Agmon, Motzkin-Schoenberg)

Assume $\|a_i\|_2 = 1,\ i = 1, \ldots, n$ and consider

$$Q = \{y \in \mathbb{R}^m : \langle a_i, y\rangle \geq b_i,\ i = 1, \ldots, n\}.$$

Relaxation method

- $y_0 := 0;\ t := 0$
- while there exists i such that $\langle a_i, y_t\rangle < b_i$

  $y_{t+1} := y_t - \lambda(b_i - \langle a_i, y_t\rangle)a_i$

  $t := t + 1$

  end

Theorem (Agmon, 1954)

If $Q \neq \emptyset$ *and* $\lambda \in (0, 2)$ *then* $y_t \to \bar{y} \in Q$.

Theorem (Motzkin-Schoenberg, 1954)

If $\text{int}(Q) \neq \emptyset$ *and* $\lambda = 2$ *then* $y_t \in Q$ *for* t *large enough.*

Perceptron algorithm (Rosenblatt, 1958)

Consider

$$C = \{y \in \mathbb{R}^m : A^\mathsf{T} y > 0\},$$

where $A = \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} \in \mathbb{R}^{m \times n},\ \ \|a_i\|_2 = 1,\ i = 1, \dots, n.$

Perceptron algorithm

- $y_0 := 0;\ t := 0$
- while there exists i such that $\langle a_i, y_t \rangle \leq 0$

  $y_{t+1} := y_t + a_i$

  $t := t + 1$

  end

Cone width

Assume $C \subseteq \mathbb{R}^m$ is a convex cone. The *width* of C is

$$\tau_C := \sup_{\|y\|_2=1} \{r : \mathbb{B}_2(y,r) \subseteq C\}.$$

Observe: $\tau_C > 0$ if and only if $\mathrm{int}(C) \neq \emptyset$.

Theorem (Block, Novikoff 1962)

Assume $C = \{y \in \mathbb{R}^m : A^\mathsf{T} y > 0\} \neq \emptyset$. Then the perceptron algorithm finds $y \in C$ is at most $\frac{1}{\tau_C^2}$ iterations.

General perceptron algorithm

The perceptron algorithm and the above convergence rate hold for a general convex cone C provided a separation oracle is available.

Notation

$$\mathbb{S}^{m-1} := \{v \in \mathbb{R}^m : \|v\|_2 = 1\}.$$

Perceptron algorithm (general case)

- $y_0 := 0; \; t := 0$
- while $y \notin C$

  let $a \in \mathbb{S}^{m-1}$ be such that $\langle a, y \rangle \leq 0 < \langle a, v \rangle, \forall v \in C$

  $y_{t+1} := y_t + a$

  $t := t + 1$

  end

Rescaled perceptron algorithm (Soheili-P 2013)

Key idea

If $C \subseteq \mathbb{R}^m$ is a convex cone and $a \in \mathbb{S}^{m-1}$ is such that

$$C \subseteq \left\{ y \in \mathbb{R}^m : 0 \leq \langle a, y \rangle \leq \frac{1}{\sqrt{6m}} \|y\|_2 \right\},$$

then dilate space along a to get wider $\hat{C} := (I + aa^{\mathsf{T}})C$.

Lemma

If $C, a, \hat{C}$ *are as above then* $\mathrm{vol}(\hat{C} \cap \mathbb{S}^{m-1}) \geq 1.5\,\mathrm{vol}(C \cap \mathbb{S}^{m-1})$.

Lemma

If C *is a convex cone then* $\mathrm{vol}(C \cap \mathbb{S}^{m-1}) \geq \frac{\tau_C}{\sqrt{1+\tau_C^2}}\,\mathrm{vol}(\mathbb{S}^{m-1})$.

Rescaled perceptron algorithm (Soheili-P 2013)

Assume a separation oracle for C is available.

Rescaled perceptron

(1) Run perceptron for C up to $6m^4$ steps

(2) Identify $a \in \mathbb{S}^{m-1}$ such that

$$C \subseteq \left\{ y \in \mathbb{R}^m : 0 \le \langle a, y \rangle \le \frac{1}{\sqrt{6m}} \|y\|_2 \right\}.$$

(3) Rescale: $C := (I + aa^\mathsf{T})C$; and go back to (1).

Theorem (Soheili-P 2013)

Assume $\mathrm{int}(C) \neq \emptyset$. *The above rescaled perceptron algorithm finds* $y \in C$ *is at most* $\mathcal{O}\left(m^5 \log\left(\frac{1}{\tau_C}\right)\right)$ *perceptron steps.*

Recall: Perceptron stops after $\frac{1}{\tau_C^2}$ steps.

Perceptron algorithm again

Consider again $A^\mathsf{T} y > 0$ where $\|a_i\|_2 = 1,\ i = 1, \ldots, n$.

Perceptron algorithm (slight variant)

- $y_0 := 0$;
- for $t = 0, 1, \ldots$

  $a_i :=$ argmin$\{\langle a_j, y_t\rangle : j = 1, \ldots, n\}$

  $y_{t+1} := y_t + a_i$

  end

Let $x(y) := \text{argmin}_{x \in \Delta_n} \langle A^\mathsf{T} y, x\rangle$, where

$$\Delta_n := \{x \in \mathbb{R}^n : x \geq 0,\ \|x\|_1 = 1\}.$$

Normalized perceptron algorithm

- $y_0 := 0$;
- for $t = 0, 1, \ldots$

  $y_{t+1} := (1 - \frac{1}{t+1})y_t + \frac{1}{t+1} A x(y_t)$

  end

Smooth perceptron (Soheili-P 2011)

Key idea

Use a smooth version of

$$x(y) = \underset{x \in \Delta_n}{\mathsf{argmin}} \langle A^{\mathsf{T}} y, x \rangle,$$

namely,

$$x_\mu(y) := \frac{\exp(-A^{\mathsf{T}} y/\mu)}{\|\exp(-A^{\mathsf{T}} y/\mu)\|_1}$$

for some $\mu > 0$.

Smooth Perceptron Algorithm

Let $\theta_t := \frac{2}{t+2}$; $\mu_t := \frac{4}{(t+1)(t+2)}$, $t = 0, 1, \ldots$

Smooth Perceptron Algorithm

- $y_0 := \frac{1}{n} A\mathbf{1}$; $x_0 := x_{\mu_0}(y_0)$;
- for $t = 0, 1, \ldots$

  $$y_{t+1} := (1-\theta_t)(y_t + \theta_t A x_t) + \theta_t^2 A x_{\mu_t}(y_t)$$
  $$x_{t+1} := (1-\theta_t)x_t + \theta_t x_{\mu_{t+1}}(y_{t+1})$$

  end for

Recall main loop in the normalized version:

for $t = 0, 1, \ldots$

$$y_{t+1} := (1 - \tfrac{1}{t+1})y_t + \tfrac{1}{t+1} A x(y_t)$$

end for

Theorem (Soheili & P, 2011)

Assume $C = \{y \in \mathbb{R}^m : A^\mathsf{T} y > 0\} \neq \emptyset$. *Then the above smooth perceptron algorithm finds* $y \in C$ *in at most*

$$\frac{2\sqrt{2\log(n)}}{\tau_C}$$

elementary iterations.

Remarks

- Smooth version retains the algorithm's original simplicity.
- Improvement on perceptron iteration bound $\frac{1}{\tau_C^2}$.
- Very weak dependence on n.

Binary classification again

Classification data

$\mathcal{D} = \{(u_1, \ell_1), \ldots, (u_n, \ell_n)\}$, with $u_i \in \mathbb{R}^d,\ \ell_i \in \{-1, 1\}$.

Linear classification

Find $\beta \in \mathbb{R}^d$ such that for $i = 1, \ldots, n$

$$\operatorname{sgn}(\beta^\mathsf{T} u_i) = \ell_i \Leftrightarrow \ell_i u_i^\mathsf{T} \beta > 0.$$

Taking $A = \begin{bmatrix} \ell_1 u_1 & \cdots & \ell_n u_n \end{bmatrix}$ and $y = \beta$ can rephrase as

$$A^\mathsf{T} y > 0.$$

Kernels and Reproducing Kernel Hilbert Spaces

- Assume $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ symmetric positive definite kernel:

$$\forall x_1, \ldots, x_m \in \mathbb{R}^d,\ [K(x_i, x_j)]_{ij} \succeq 0.$$

- Reproducing Kernel Hilbert Space

$$\mathcal{F}_K := \left\{ f(\cdot) = \sum_{i=1}^{\infty} \beta_i K(\cdot, z_i),\ \beta_i \in \mathbb{R},\ z_i \in \mathbb{R}^d,\ \|f\|_K < \infty \right\}.$$

- Feature mapping

$$\begin{aligned} \phi : \mathbb{R}^d &\to \mathcal{F}_K \\ u &\mapsto K(\cdot, u) \end{aligned}$$

- For $f \in \mathcal{F}_K$ and $u \in \mathbb{R}^d$ we have $f(u) = \langle f, \phi(u) \rangle_K$

Kernelized classification

Nonlinear kernelized classification

Find $f \in \mathcal{F}_K$ such that for $i = 1, \dots, n$

$$\operatorname{sgn}(f(u_i)) = \ell_i \Leftrightarrow \ell_i f(u_i) > 0$$

Separation margin

Assume $\mathcal{D} = \{(u_1, \ell_1), \dots, (u_n, \ell_n)\}$ and K are given. Define the *margin* ρ_K as

$$\rho_K := \sup_{\|f\|_K = 1} \min_{i=1,\dots,n} \ell_i f(u_i).$$

Kernelized smooth perceptron

Theorem (Ramdas & P 2014)

Assume $\rho_K > 0$.

(a) *Kernelized version of the smooth perceptron finds a nonlinear separator after at most* $\frac{2\sqrt{2\log n}\|\mathcal{D}\|}{\rho_K}$ *iterations.*

(b) *Kernelized smooth perceptron generates* $f_t \in \mathcal{F}_K$ *such that*

$$\|f_t - f^*\|_K \leq \frac{2\sqrt{2\log n}\|\mathcal{D}\|}{t},$$

where $f^* \in \mathcal{F}$ *separator with best margin.*

Open problems

Smale's 9th problem

Is there a polynomial-time algorithm over the real numbers which decides the feasibility of the linear system of inequalities $Ax \geq b$?

Related work

- Tardos, 1986: A polynomial algorithm for combinatorial linear programs.
- Renegar, Freund, Cucker, P (2000s): Algorithms that are polynomial in problem dimension and *condition number* $C(A, b)$.
- Ye, 2005: A polynomial interior-point algorithm for the Markov Decision Problem with fixed discount rate.
- Ye, 2011: The simplex method is polynomial for the Markov Decision Problem with fixed discount rate.

Hirsch conjecture

A *polyhedron* is a set of form

$$\{y \in \mathbb{R}^m : \langle a_i, y \rangle - b_i \geq 0,\ i = 1, \ldots, n\}.$$

A *face of a polyhedron* is a non-empty intersection with a non-cutting hyperplane.

Vertices: zero-dimensional faces.
Edges: one-dimensional faces.
Facets: highest-dimensional faces.

Observation

The vertices and edges of a polyhedron form a graph.

Hirsch conjecture

Conjecture (Hirsch, 1957)

For every polyhedron P with n facets and dimension d

$$\text{diam}(P) \leq n - d.$$

Related work

- Klee and Walkup, 1967: Unbounded counterexample.
- True for special classes of bounded polyhedra.
- Santos, 2012: First bounded counterexample.
- Todd, 2014: $\text{diam}(P) \leq d^{\log_2(n-d)}$.

Question

Small bound (e.g., linear in n, d) on $\text{diam}(P)$?

Lax conjecture

Definition

A homogeneous polynomial $p \in \mathbb{R}[x]$ is *hyperbolic* if there exists $e \in \mathbb{R}^n$ such that for every $x \in \mathbb{R}^n$ the roots of

$$t \mapsto p(x + te)$$

are real.

Theorem (Garding, 1959)

Assume p *is hyperbolic. Then each connected component of* $\{x \in \mathbb{R}^n : p(x) > 0\}$ *is an open convex cone.*

Hyperbolicity cone: Connected component of $\{x \in \mathbb{R}^n : p(x) > 0\}$ for some hyperbolic polynomial p.

Lax conjecture

Question

Can every hyperbolicity cone be described in terms of linear matrix inequalities?

$$\left\{ y \in \mathbb{R}^m : \sum_{j=1}^{m} A_j y_j \succeq 0 \right\}.$$

Related work

- Helton and Vinnikov, 2007: Every hyperbolicity cone in $\mathbb{R}^3$ is of the form

  $$\{y \in \mathbb{R}^3 : I x_1 + A_2 x_2 + A_3 x_3 \succeq 0\},$$

  for some symmetric matrices A_2, A_3 (Lax conjecture, 1958).
- Branden, 2011: Disproved some versions of this conjecture for more general hyperbolicity cones in $\mathbb{R}^n$.

Slides and references

jpena@cmu.edu

http://www.andrew.cmu.edu/user/jfp/