Short Course
Robust Optimization and Machine Learning

Lecture 1: Optimization Models

Laurent El Ghaoui

EECS and IEOR Departments
UC Berkeley

Spring seminar TRANSP-OR, Zinal, Jan. 16-19, 2012

Outline

Outline

Optimization problem

A standard form

An optimization problem is a problem of the form

$$p^* := \min_x \; f_0(x) \text{ subject to } f_i(x) \le 0, \quad i = 1, \ldots, m,$$

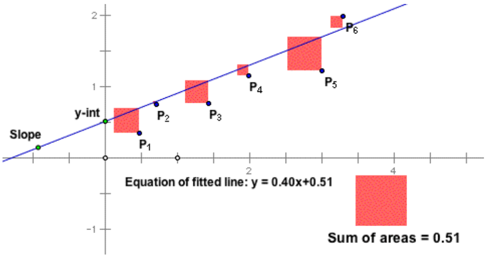
where

- $x \in \mathbf{R}^n$ is the *decision variable* ;
- $f_0 : \mathbf{R}^n \to \mathbf{R}$ is the *objective* (or, *cost*) function;
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$ represent the *constraints* ;
- p^* is the *optimal value* .

Often the above is referred to as a “mathematical program” (for historical reasons).

Example

Least-squares regression



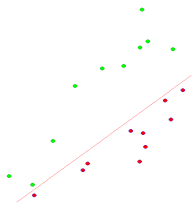
$$\min_{w} \|X^T w - y\|_2$$

where

- $X = [x_1, \ldots, x_m]$ is a $n \times m$ matrix of data points ($x_i \in \mathbf{R}^n$);
- y is a response vector;
- $\| \cdot \|_2$ is the l_2 (*i.e.*, Euclidean) norm.
- Many variants (with *e.g.*, constraints) exist (more on this later).
- Perhaps the most popular / useful optimization problem.

Example

Linear classification



$$\min_{w,b} \sum_{i=1}^{m} \max(0, 1 - y_i(w^T x_i + b))$$

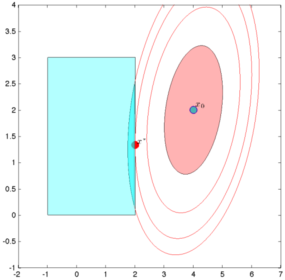
where

- $X = [x_1, \ldots, x_m]$ is a $n \times m$ matrix of data points ($x_i \in \mathbf{R}^n$);
- $y \in \{-1, 1\}$ is a *binary* response vector;
- Many variants (with *e.g.*, constraints) exist (more on this later).
- Very useful for classifying data (*e.g.*, text documents).

Nomenclature

A toy optimization problem

$$\min_{x} \quad 0.9x_1^2 - 0.4x_1x_2 - 0.6x_2^2 - 6.4x_1 - 0.8x_2$$
$$\text{s.t.} \quad -1 \le x_1 \le 2, \quad 0 \le x_2 \le 3.$$



- *Feasible set* in light blue.
- 0.1- *suboptimal set* in darker blue.
- *Unconstrained minimizer* : x_0; optimal point: x^*.
- *Level sets* of objective function in red lines.
- A *sub-level set* in red fill.

Other standard forms

Equality constraints. We may single out equality constraints, if any:

$$\min_x \; f_0(x) \text{ subject to } \begin{array}{l} h_i(x) = 0, \quad i = 1, \ldots, p, \\ f_i(x) \leq 0, \quad i = 1, \ldots, m, \end{array}$$

where h_i's are given. Of course, we may reduce the above problem to the standard form above, representing each equality constraint by a pair of inequalities.

Abstract forms. Sometimes, the constraints are described abstractly via a set condition, of the form $x \in \mathcal{X}$ for some subset $\mathcal{X}$ of $\mathbf{R}^n$. The corresponding notation is

$$\min_{x \in \mathcal{X}} \; f_0(x).$$

Minimization vs. maximization

Some problems come in the form of maximization problems. Such problems are readily cast in standard form via the expression

$$\max_{x \in \mathcal{X}} f_0(x) = -\min_{x \in \mathcal{X}} : g_0(x),$$

where $g_0 := -f_0$.

- *Minimization* problems correspond to loss, cost or risk minimization.
- *Maximization* problems typically correspond to utility or return (*e.g.*, on investment) maximization.

Penalization

A trade-off between two objecgives is commonly accomplished via a *penalized* problem:

$$\max_x \; f(x) + \lambda g(x),$$

where f and g represent loss and risk functions, and $\lambda > 0$ is a risk-aversion parameter.

Example: penalized least-squares

$$\min_w \; \|X^T w - y\|_2^2 + \lambda \|w\|_2^2$$

Here, the risk term $\|w\|_2^2$ controls the variance associated with noise in X.

Robust optimization
Definition

In many instances the problem data is not known exactly. Assume that the functions f_i in the original problem also depend on an "uncertainty" vector u that is unknown, but bounded: $u \in \mathcal{U}$, with the set $\mathcal{U}$ given.

Robust counterpart:

$$\begin{array}{ll} \min_x \max_{u \in \mathcal{U}} & f_0(x, u) \\ \text{subject to} & \forall\, u \in \mathcal{U}, \quad f_i(x, u) \leq 0, \quad i = 1, \ldots, m. \end{array}$$

- Robust counterparts are sometimes tractable.
- If not, systematic procedures exist to generate approximations.

Robust optimization

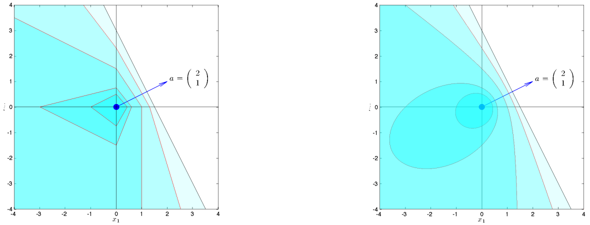
Geometry

Given $a \in \mathbf{R}^n$, $b \in \mathbf{R}$, consider the constraint in $x \in \mathbf{R}^n$

$$(a+u)^T x \leq b,$$

with u's components are only known within a given set $\mathcal{U}$. The robust counterpart is:

$$\forall\, u \in \mathcal{U} \;:\; (a+u)^T x \leq b.$$



Robust counterpart when $\mathcal{A}$ is a box (left panel) and a sphere (right panel).

Stochastic optimization

Definition

In stochastic programming, the uncertainty is described by a random variable, with known distribution.

Two-stage stochastic linear program with recourse:

$$\min_{x \in \mathcal{X}} \; a^T x + f(x) \;:\; f(x) = \mathbf{E}_w[\min_{y \in \mathcal{Y}(x,w)} c(w)^T y].$$

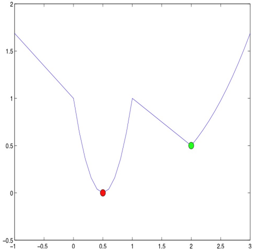
- x-variables correspond to decisions taken now.
- y-variables correspond to decisions taken when uncertainty w is revealed.

- Stochastic problems are usually very hard.
- Most known approaches are very expensive to solve.

Outline

Global vs. local minima

The curse of optimization



- Point in red is globally optimal (optimal for short).
- Point in green is only locally optimal.
- In many applications, we are interested in global minima.

Curse of optimization

Optimization algorithms for general problems can be trapped in local minima.

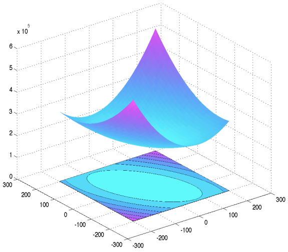
Convex function

Definition

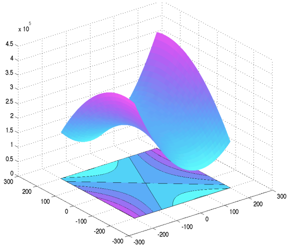
A function $f : \mathbf{R}^n \to \mathbf{R}$ is convex if it satisfies the condition

$$\forall\, x, y \in \mathbf{R}^n, \lambda \in [0, 1] \;:\; f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y)$$

Geometrically, the graph of the function is “bowl-shaped”.



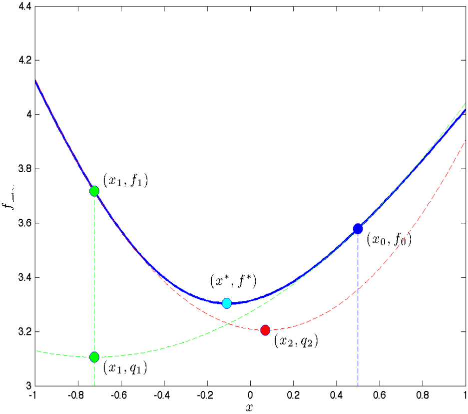
Convex function.



Non-convex function.

Convexity and local minima

When trying to minimize convex functions, specialized algorithms will always converge to a global minimum, irrespective of the starting point, provided some (weak) assumptions on the function hold.



The Newton algorithm.

Convex optimization

Definition

The problem in standard form

$$p^* := \min_x \; f_0(x) \text{ subject to } f_i(x) \leq 0, \quad i = 1, \dots, m,$$

is convex if the functions $f_0, \dots, f_m$ are all convex.

Examples:

- Linear programming ($f_0, \dots, f_m$ affine).
- Quadratic programming (f_0 convex quadratic, $f_1, \dots, f_m$ affine).
- Second-order cone programming (f_0 linear, f_i's of the form $\|A_i x + b_i\|_2 + c_i^T x + d_i$, for appropriate data A_i, b_i, c_i, d_i).

Software for convex optimization

- Free (if you have matlab): CVX [3], Yalmip, Mosek's student version [1].
- Really free: [4] (in development).
- Commercial: Mosek, CPLEX, etc.

Non-convex problems

Examples

- *Boolean/integer optimization:* some variables are constrained to be Boolean or integers. Convex optimization can be used for getting (sometimes) good approximations.
- *Cardinality-constrained problems:* we seek to bound the number of non-zero elements in a vector variable. Convex optimization can be used for getting good approximations.
- *Non-linear programming:* usually non-convex problems with differentiable objective and functions. Algorithms provide only local minima.

Not all non-convex problems are hard! *e.g.*, low-rank approximation problem.

Outline

References

E. D. Andersen and K. D. Andersen.
The MOSEK optimization package.

A. Bental, L. El Ghaoui, and A. Nemirovski.
Robust Optimization.
Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

S. Boyd and M. Grant.
The CVX optimization package, 2010.

S. Boyd and T. Tinoco.
The CVXPY optimization package, 2010.

S. Boyd and L. Vandenberghe.
Convex Optimization.
Cambridge University Press, 2004.

G. Cornuejols and R. Tütüncü.
Optimization Methods in Finance.
Cambridge University Press, 2007.