Optimization Models

EECS 127 / EECS 227AT

Laurent El Ghaoui

EECS department
UC Berkeley

Fall 2018

LECTURE 25

Algorithms for Convex Optimization

The word algorithm itself derives from the 9th Century Persian mathematician Muhammad ibn Musa al-Khwarizmi, Latinized Algoritmi.

Wikipedia.

Outline

Goals

- Present principles in algorithm design
- Focus on broad types:
  - 1D problems
  - Coordinate descent methods
  - First-order (gradient) methods
  - Second-order methods

Some criteria for algorithm design:

- convergence to local vs global optima;
- speed of convergence;
- computational cost, per iteration and overall.

1D problems

Goal: solve

$$\min_x \ f(x)$$

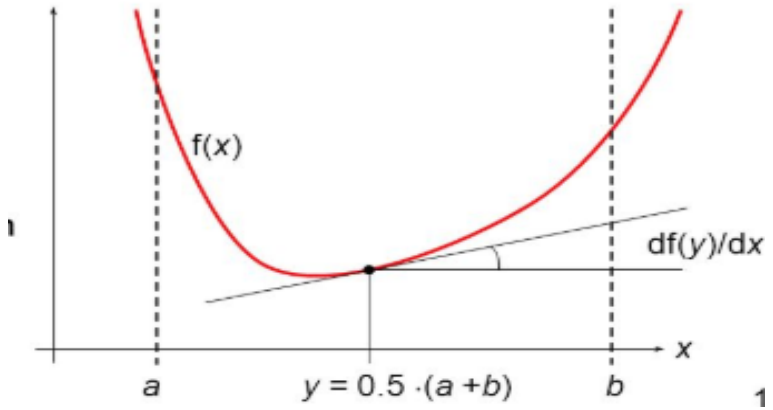
where $f : \mathbb{R} \to \mathbb{R}$. We assume that we know a finite interval $[x_{\text{low}}, x_{\text{up}}]$ that is guaranteed to contain an optimal point.

- If f is easy to evaluate, we can proceed by gridding.
- If f is convex and differentiable, we can do better.
- Method easily extends to the non-differentiable case.

Bisection for convex problems

In the bisection method, we update an interval $[x_{\mathrm{low}}, x_{\mathrm{up}}]$ as:

1. set $x = \frac{1}{2}(x_{\mathrm{up}} + x_{\mathrm{low}})$;
2. if $f'(x) > 0$, set $x_{\mathrm{up}} = x$; otherwise, set $x_{\mathrm{low}} = x$.



- At each step we half the interval, hence the convergence to an ϵ-suboptimal point is in $O(\log(1/\epsilon))$.
- The cost of each step is dominated by the cost of evaluating the gradient of f at x.

Initial guess for bisection

If an initial interval is not known, we can guess one; if we find that we optimum found is at one of the end points we may double the size of the interval and re-run.

In some cases it is possible to find an interval via direct analysis. Consider the regularized problem

$$p^* = \min_x \ f_0(x) + \lambda|x|$$

where $\lambda > 0$.
Assume that a lower bound $f_{\min}$ on f is known: for every x, $f(x) \geq f_{\min}$. Since the optimal value is less than the value of the objective at $x = 0$, we have $f_0(x^*) + \lambda|x^*| \leq f_0(0)$, hence

$$|x^*| \leq \frac{1}{\lambda}(f_0(0) - f(x^*)) \leq x_{\max} \doteq \frac{1}{\lambda}(f_0(0) - f_{\min}).$$

Extending the reach of 1D optimization

1D optimization is conceptually very simple but surprisingly useful, thanks to duality.

Foer example consider a problem of the form

$$p^* = \min_x \sum_{i=1}^m f_i(x_i) \ : \ a^T x = b,$$

where $f_i \ : \ \mathbb{R} \to \mathbb{R}$, $i = 1, \ldots, m$ are convex 1D function, and $a \in \mathbb{R}^n$, $b \in \mathbb{R}$ are given. Here the "coupling" constraint makes it impossible to directly apply 1D optimization.

Assuming strong duality, we have $p^* = \max_\nu \ g(\nu)$, where g is the dual function:

$$g(\nu) \doteq b\nu + \min_x \sum_{i=1}^m (f_i(x_i) - \nu a_i x_i)$$

Problem decomposition

Since the primal objective is decomposable (as a sum of independent functions), we can exchange the "min" and the sum in the definition of the dual function:

$$g(\nu) = b\nu + \sum_{i=1}^{m} \min_{x_i} \left(f_i(x_i) - \nu a_i x_i\right) = b\nu - \sum_{i=1}^{m} f_i^*(\nu a_i),$$

where f_i^* is the so-called "conjugate" of f:

$$f_i^*(z) \doteq \max_{\xi} z\xi - f_i(\xi).$$

- Each f_i^* is a convex function, which can be evaluated by bisection, or sometimes in closed-form;
- The dual problem is amenable to bisection as well.
- It can be shown that if the maximizing point ξ_i^* in the definition of f_i^* is unique, then f_i^* is differentiable, with gradient at z equal to ξ_i^*.

Coordinate descent methods

Another way to extend the reach of 1D optimization is via coordinate descent methods. These are well-suited to problems with simple "box" constraints:

$$\min_x \ f(x) \ : \ \|x\|_\infty \leq 1,$$

or penalized problems of the form

$$\min_x \ f(x) + \lambda \|x\|_1,$$

where f is convex.

Basic idea:

- Optimize over one variable at a time.
- Each step is amenable to 1D optimization, sometimes via a closed-form expression.
- Can be proven to converge to the global optimum if f is strictly convex.

Coordinate descent methods

- Coordinate descent methods, or more generally block-coordinate descent methods, apply to problems where each variable (or block of variables) is *independently* constrained.
- We consider a special case of a generic minimization problem

$$\min_{x=(x_1,\ldots,x_\nu)} f_0(x) \;:\; x_i \in \mathcal{X}_i, \;\; i = 1,\ldots,\nu. \tag{1}$$

  In words, the variable x can be decomposed into ν blocks $x_1, \ldots, x_\nu$, and each block x_i is independently constrained to belong to the set $\mathcal{X}_i$.
- Coordinate descent methods are based on iteratively minimizing with respect to one block, with all the other blocks being fixed.
- If $x^{(k)} = (x_1^{(k)}, \ldots, x_\nu^{(k)})$ denotes the value of the decision variable at iteration k, partial minimization problems of the form

$$\min_{x_i \in \mathcal{X}_i} f_0(x_1^{(k)}, \ldots, x_{i-1}^{(k)}, x_i, x_{i+1}^{(k)}, \ldots, x_\nu^{(k)}), \tag{2}$$

  are solved.
- Different methods ensue, based on how exactly we form the next iterate.

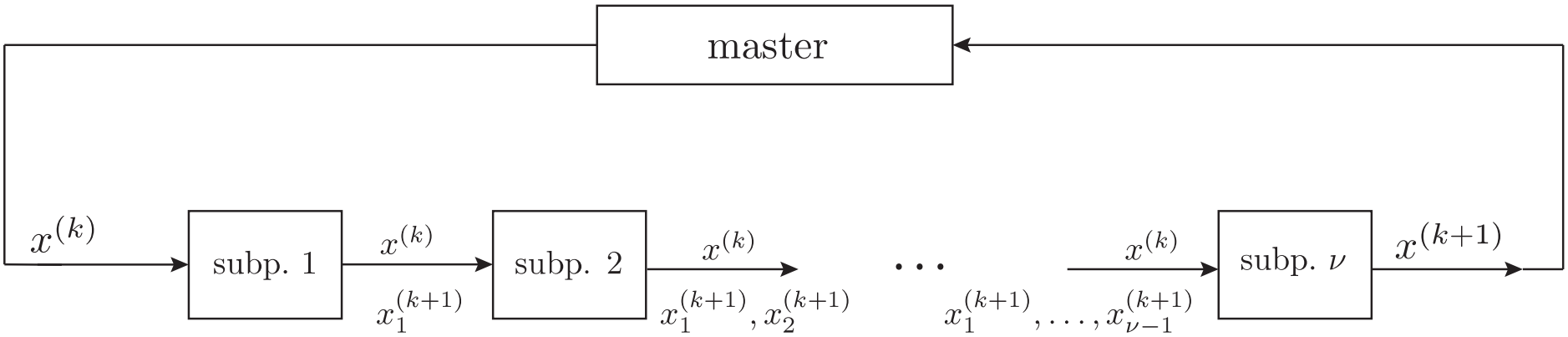
The Gauss–Seidel method

- In the standard (block) coordinate minimization method (BCM), also known as the Gauss–Seidel method, the variable blocks are updated *sequentially*, according to the recursion

$$x_i^{(k+1)} = \arg\min_{x_i \in \mathcal{X}_i} f_0(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i, x_{i+1}^{(k)}, \ldots, x_\nu^{(k)}), \tag{3}$$

  for $i = 1, \ldots, \nu$.

- The scheme is depicted in the following figure.



The Gauss–Seidel method

Convergence.

Theorem 1

- *Assume f_0 is convex and continuously differentiable on $\mathcal{X}$. Moreover, let f_0 be strictly convex in x_i, when the other variable blocks x_j, $j \neq i$ are held constant.*
- *If the sequence $\{x^{(k)}\}$ generated by the BCM algorithm is well defined, then every limit point of $\{x^{(k)}\}$ converges to an optimal solution of problem (1).*

The Gauss–Seidel method

Convergence.

- The sequential block-coordinate descent method may fail to converge, in general, for non-smooth objectives, even under convexity assumptions.
- An important exception, however, arises when f_0 is a composite function which can be written as the sum of a convex and differentiable function ϕ and a separable convex (but possibly non-smooth) term, that is

$$f_0(x) = \phi(x) + \sum_{i=1}^{\nu} \psi_i(x_i), \tag{4}$$

  where ϕ is convex and differentiable, and ψ_i, $i = 1, \ldots, \nu$ are convex.
- Notice that this setup includes the possibility of convex independent constraints on the variables of the form $x_i \in \mathcal{X}_i$, since functions ψ_i may include a term given by the indicator function of the set $\mathcal{X}_i$.
- The structure (4) also includes various ℓ_1-norm regularized problems, such as the LASSO, for which convergence of sequential coordinate descent methods is thus guaranteed.

Coordinate minimization for the LASSO

Exercise.

Code in Matlab a coordinate minimization algorithm for solving the LASSO problem

$$\min_{x \in \mathbb{R}^n} \ \frac{1}{2}\|Ax - y\|_2^2 + \lambda \|x\|_1,$$

where $A \in \mathbb{R}^{m,n}$ is a given matrix, $y \in \mathbb{R}^m$ is a given vector, and $\lambda \geq 0$ is an assigned scalar tradeoff parameter.

Hints...

- If x is the current point, then the i-th coordinate minimization problem takes the form (all but the i-variable are fixed to the values in x, and we minimize with respect to x_i)

$$\min_{x_i \in \mathbb{R}} \ \frac{1}{2}\|a_i x_i - y^{(i)}\|_2^2 + \lambda |x_i|_1 + \lambda \|x_{(-i)}\|_1,$$

  where a_i is the i-th column of A, $x_{(-i)}$ is a vector obtained from x by fixing the i-th entry to zero, and $y^{(i)} \doteq y - A x_{(-i)}$.

- Prove that the above uni-variate minimization problem has the following optimal solution:

$$x_i^* = \begin{cases} 0 & \text{if } |a_i^\top y^{(i)}| \leq \lambda, \\ \xi_i - \operatorname{sgn}(\xi_i) \frac{\lambda}{\|a_i\|_2^2} & \text{if } |a_i^\top y^{(i)}| > \lambda, \end{cases}$$

  where

$$\xi_i \doteq \frac{a_i^\top y^{(i)}}{\|a_i\|_2^2}$$

  corresponds to the solution of the problem for $\lambda = 0$. Verify that this solution can be expressed more compactly as $x_i^* = \operatorname{sthr}_{\lambda/\|a_i\|_2^2}(\xi_i)$, where sthr is the *soft threshold* function.

Non-convex example: power iteration

Consider the low-rank approximation problem:

$$\min_{x,y} \|M - xy^\top\|_F^2 = \|x\|_2^2 \cdot \|y\|_2^2 - 2x^\top My + \text{constant},$$

where $M \in \mathbb{R}^{n\times m}$ is given. In a "block" version of coordinate descent, we optimize in x, y alternatively; each sub-problem has a closed-form solution. This leads to the updates

$$x^{(k+1)} = \frac{1}{\|y^{(k)}\|_2^2} My^{(k)}, \quad y^{(k+1)} = \frac{1}{\|y^{(k)}\|_2^2} M^\top x^{(k)}.$$

We can normalize the iterates to have unit-norm at each step, leading to the so-called power iteration method, which is the preferred method for this problem in a large-scale setting.

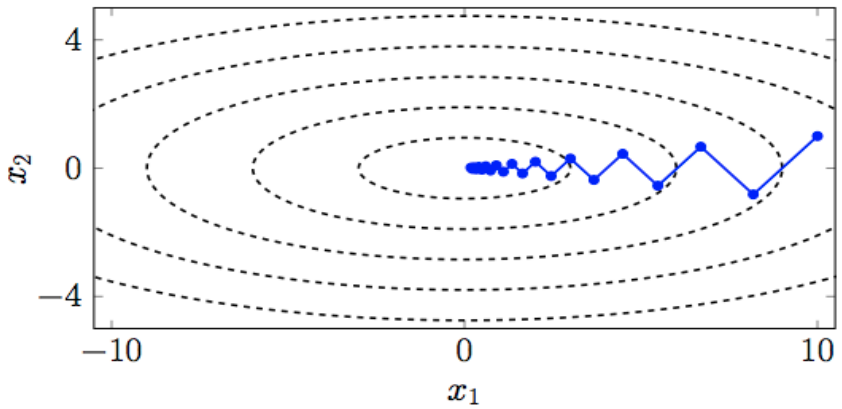
The method is naturally extended to generalized low-rank models, such as non-negative matrix factorization (a variant of the above in which x, y are required to be non-negative componentwise).

Basic gradient method

The basic gradient method is to solve

$$\min_x \ f_0(x)$$

where f_0 is convex, differentiable.



The iterates take the form

$$x^{(k+1)} = x^{(k)} - s_k \nabla f_0(x^{(k)}).$$

where s_k is a "step size", referred to as the "learning rate" in machine learning applications when chosen to be constant.

- convergence can be proven based on mild conditions on f_0;
- step size can be either constant, or defined based on more sophisticated rules;
- gradient method is often slow; convergence very dependent on variable scaling.

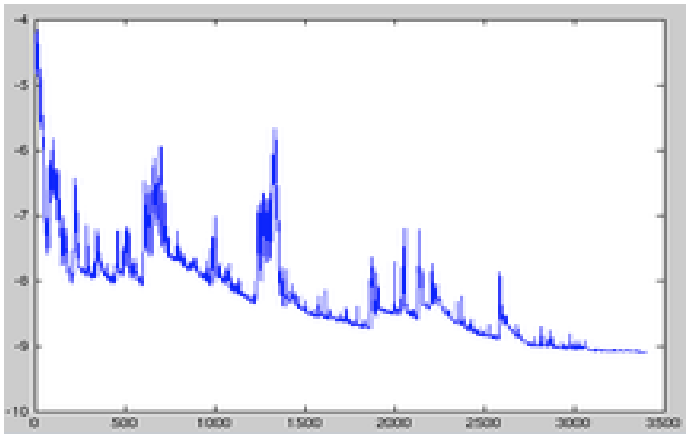
Stochastic gradient descent (SGD)

Stochastic gradient descent methods are well adapted to the minimization of functions of the "sum" form, such as

$$\min_w \ f_0(w) = \frac{1}{m} \sum_{i=1}^{n} \mathcal{L}(w^\top x_i)$$

where $\mathcal{L}$ is a loss function, $x_i \in \mathbb{R}^n$ are data points, $i = 1, \ldots, m$. In a SGD method, the gradient is approximated via a small sample, or "mini-batch". For example, we can approximate the gradient at point w by the gradient of a single term in the sum, *i.e.* one data point i:

$$\nabla f_0(w) \approx \mathcal{L}'(w^\top x_i) x_i.$$



The SGD method can be proven to converge when the objective function is convex, but it is extremely slow.

Projected gradient method

Projected gradient methods can be applied to constrained problem of the form

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

where the set $\mathcal{X}$ is "simple" enough, so that one can easily find the projection $\pi_{\mathcal{X}}(x)$ of any point x on $\mathcal{X}$. For example for $\mathcal{X} = \mathbb{R}^n_+$, $\pi_{\mathcal{X}}(x) = \max(0, x)$, the vector obtained from x by zeroing out negative elements.

The method consists in projecting gradient iterates on $\mathcal{X}$ at each step:

$$x^{(k+1)} = \pi_{\mathcal{X}}(x^{(k)} - s_k \nabla f_0(x^{(k)})).$$

For example with $\mathcal{X} = \mathbb{R}^n_+$:

$$x^{(k+1)} = \max(0, x^{(k)} - s_k \nabla f_0(x^{(k)})).$$

Extension via duality

The reach of the method can be extended via Lagrange duality. Consider for example the QP

$$\min_x \ \frac{1}{2}\|x\|_2^2 \ : \ Ax \leq b$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. The dual is a sign-constrained QP:

$$\max_{\lambda \geq 0} \ -\lambda^\top b - \frac{1}{2}\lambda^\top AA^\top \lambda,$$

which is amenable to projected gradient:

$$\lambda^{(k+1)} = \max(0, \lambda^{(k)} - s_k AA^\top \lambda^{(k)}).$$

Once solved, we set $x^* = -A^\top \lambda^*$.

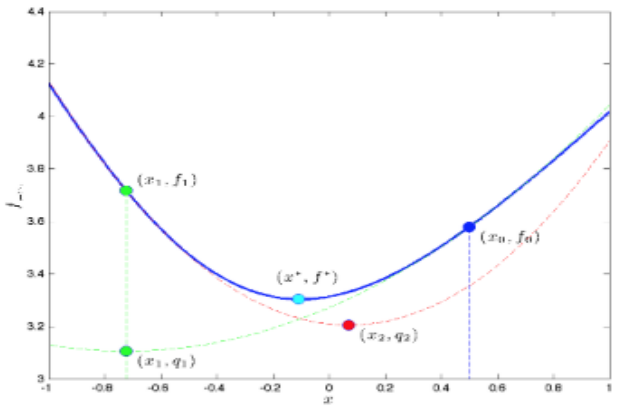
Newton's method for unconstrained minimization

Consider the convex problem

$$\min_x \ f_0(x)$$

Assume that ∇f, $\nabla^2 f$ are available. Idea: at each step, find a minimizer of the local quadratic approximation to min

$$\begin{aligned} x^{(k+1)} &= \arg\min_x \ f(x^{(k)}) + \nabla f(x^{(k)})^\top (x - x^{(k)}) + (x - x^{(k)})^\top \nabla^2 f(x^{(k)})(x - x^{(k)}) \\ &= x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}). \end{aligned}$$



Two initial steps of Newton's method to minimize the function with values on $\mathbb{R}^2$

$$f(x) = \log(\exp(x - 3) + \exp(-2x + 2)).$$

Interior-point method for constrained problems

Consider a convex, constrained problem:

$$\min_x \ f_0(x) \ : \ f_i(x) \leq 0, \quad i = 1, \ldots, m,$$

we approximate the above problem via the convex unconstrained problem

$$\min_x \ f_0(x) - \mu \sum_{i=1}^{m} \log(-f_i(x))$$

where $\mu > 0$.

- iterates are always "interior" to the feasible set;
- setting $\mu = m/\epsilon$ guarantees that unconstrained minimizer is ϵ-suboptimal;
- advanced methods use iteration-dependent μ values;
- convergence extremely fast, but each step is costly;
- used in CVX.

Summary

- First-order method are slow, but each iteration is very cheap;
- they are well-adapted to unconstrained problems, and some problems with "simple" constraints;
- second-order methods can handle (almost) any convex constrained problem; each iteration is expensive, and more memory-hungry.

For non-convex problems, algorithms

- for unconstrained problems, usually only find a local minimum;
- for constrained problems, may fail to find a feasible point.