Computational Intelligence Laboratory

Lecture 7: Optimization

Martin Jaggi

ETH Zurich – cil.inf.ethz.ch

April 17, 2015

Outline

Optimization

General Optimization Problem

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{with} \quad & \mathbf{x} \in \mathbb{R}^D \end{aligned}$$

for convenience: $f : \mathbb{R}^D \to \mathbb{R}$ is continuous and differentiable

Why? And How?

optimization is everywhere

machine learning, big data, statistics, data analysis of all kinds, finance, logistics, planning, control theory, mathematics, search engines, simulations, and many other applications ...

- **Mathematical Modeling**
  (defining the optimization problem)
- **Solving It**
  (running an optimization algorithm)

Optimization Algorithms

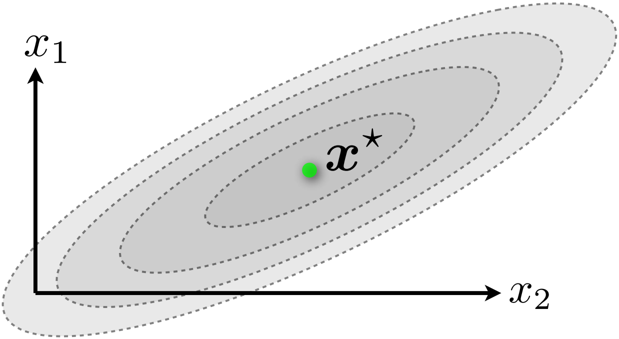
the main contenders:

- **Coordinate Descent**
- **Gradient Descent**
- **Stochastic Gradient Descent**

History: Early roots: Cauchy 1847. Linear Programming in the 1950's. General Optimization in 1980's, together with new Convex Optimization theory. Now active research field again in the wake of big data.

Coordinate Descent

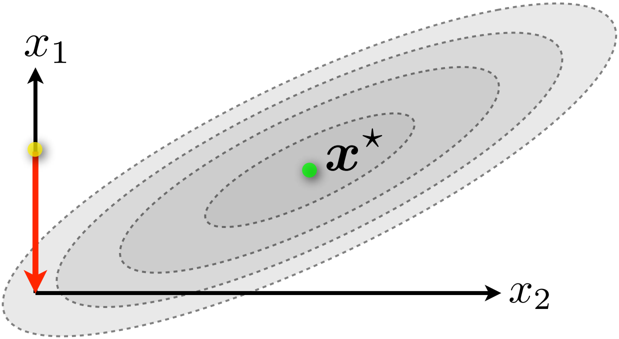
Goal: Find $\mathbf{x}^\star \in \mathbb{R}^D$ minimizing $f(\mathbf{x})$. (Example: $D = 2$)



Idea: Update one coordinate at a time, while keeping others fixed.

Coordinate Descent

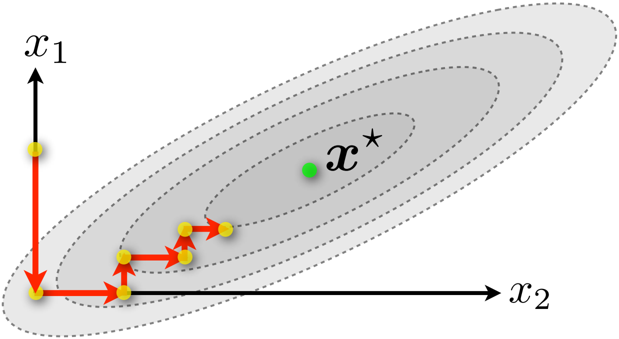
Goal: Find $\mathbf{x}^\star \in \mathbb{R}^D$ minimizing $f(\mathbf{x})$.



Idea: Update one coordinate at a time, while keeping others fixed.

Coordinate Descent

Goal: Find $\mathbf{x}^\star \in \mathbb{R}^D$ minimizing $f(\mathbf{x})$.



Idea: Update one coordinate at a time, while keeping others fixed.

Coordinate Descent

Idea: Update one coordinate at a time, while keeping others fixed.

Coordinate Descent

initialize $\mathbf{x}^{(0)} \in \mathbb{R}^D$

for t = 0:maxIter **do**

sample a coordinate d uniformly at random from $1 \ldots D$.

optimize f w.r.t. that coordinate:

$$u^\star \leftarrow \underset{u \in \mathbb{R}}{\operatorname{argmin}} \; f\big(x_1^{(t)}, \ldots, x_{d-1}^{(t)}, u, x_{d+1}^{(t)}, \ldots, x_D^{(t)}\big)$$

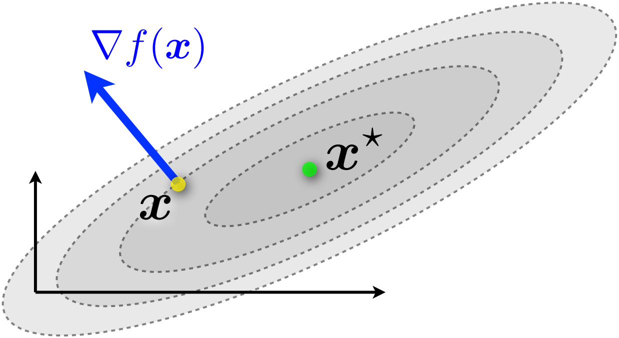
update $x_d^{(t+1)} \leftarrow u^\star$

$x_{d'}^{(t+1)} \leftarrow x_{d'}^{(t)}$ for $d' \neq d$

end for

Navigating the Optimization Landscape

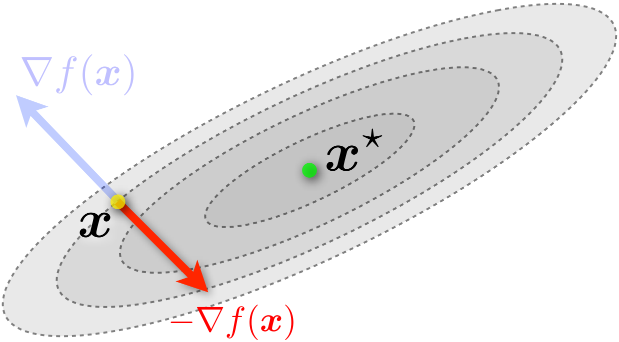
The Direction of Steepest Change:



Gradient of a function $f : \mathbb{R}^D \to \mathbb{R}$ is

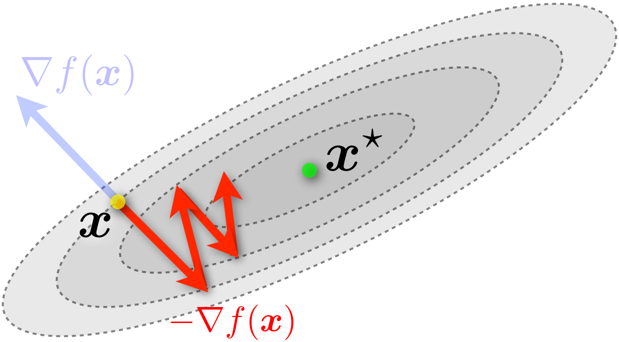
$$\nabla f(\mathbf{x}) := \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_D} \right)^\mathsf{T} \in \mathbb{R}^D$$

Gradient Descent Method



initialize $\mathbf{x}^{(0)} \in \mathbb{R}^D$
for t = 0:maxIter **do**
   update $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} - \gamma \nabla f(\mathbf{x}^{(t)})$
end for

Gradient Descent Method



initialize $\mathbf{x}^{(0)} \in \mathbb{R}^D$
for t = 0:maxIter **do**
  update $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} - \gamma \nabla f(\mathbf{x}^{(t)})$
end for

Gradient Descent Method

initialize $\mathbf{x}^{(0)} \in \mathbb{R}^D$
for t = 0:maxIter **do**
  update $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} - \gamma \nabla f(\mathbf{x}^{(t)})$
end for

Cauchy 1847

- simple to implement
- good scalability and robustness
- **stepsize** γ usually decreasing with $\gamma \approx \frac{1}{t}$

Stochastic Gradient Descent

Optimization Problem Structure

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} f_n(\mathbf{x}) \\ \text{with} \quad & \mathbf{x} \in \mathbb{R}^D \end{aligned}$$

Stochastic Gradient Descent (SGD)

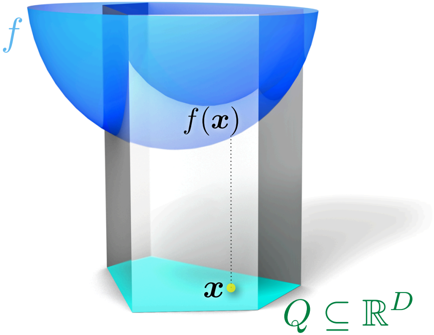
initialize $\mathbf{x}^{(0)} \in \mathbb{R}^D$
for t = 0:maxIter **do**
  sample n uniformly at random from $1 \dots N$.
  update $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} - \gamma \nabla f_n(\mathbf{x}^{(t)})$
end for

Stochastic Gradient Descent - Why Does It Work?

SGD update $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} - \gamma \nabla f_n(\mathbf{x}^{(t)})$

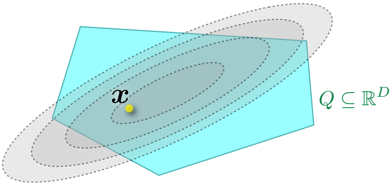
- **Idea:** Cheap but unbiased estimate of the gradient
  - $\mathsf{E}[\nabla f_n(\mathbf{x})] = \nabla f(\mathbf{x})$ over the random choice of n.
- Computing $\nabla f_n(\mathbf{x})$ is much cheaper than computing $\nabla f(\mathbf{x})$.
  - Typically N times cheaper
- Again use a decreasing stepsize $\gamma \approx \frac{1}{t}$

Constrained Optimization



Constrained Optimization Problem

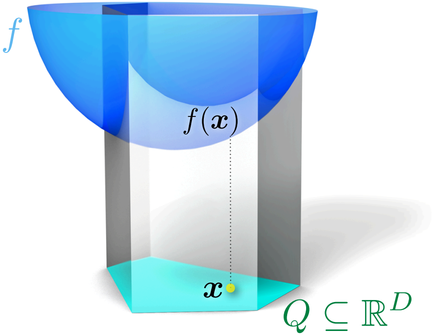
$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in Q \end{aligned}$$



Constrained Optimization

Constrained Optimization Problem

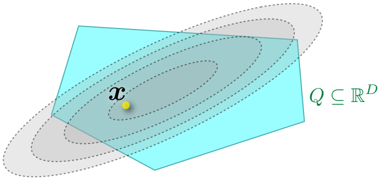
$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in Q \end{array}$$



Solving Constrained Optimization Problems

A Projected Gradient Descent

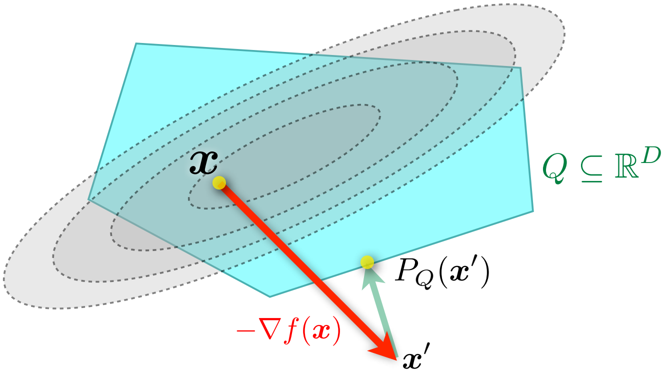
B Transform it into an *unconstrained* problem



Projected Gradient Descent

Idea: add a projection onto Q after every step

$P_Q(\mathbf{x}') := \mathrm{argmin}_{\mathbf{y}\in Q} \|\mathbf{y} - \mathbf{x}'\|$



Projected gradient update $\mathbf{x}^{(t+1)} \leftarrow P_Q\left[\mathbf{x}^{(t)} - \gamma\nabla f(\mathbf{x}^{(t)})\right]$

Turning Constrained into Unconstrained Problems

Use **penalty functions** instead of directly solving $\min_{\mathbf{x}\in Q} \; f(\mathbf{x})$.

- "*brick wall*" (indicator function) $\quad I_Q(\mathbf{x}) := \begin{cases} 0 & \mathbf{x} \in Q \\ \infty & \mathbf{x} \notin Q \end{cases}$

$$\Rightarrow \min_{\mathbf{x}\in\mathbb{R}^D} \; f(\mathbf{x}) + I_Q(\mathbf{x})$$

(disadvantage: non-continuous objective)

- Penalize error
  Example: $Q = \{\mathbf{x} \in \mathbb{R}^D \mid A\mathbf{x} = \mathbf{b}\}$

$$\Rightarrow \min_{\mathbf{x}\in\mathbb{R}^D} \; f(\mathbf{x}) + \lambda \, \|A\mathbf{x} - \mathbf{b}\|^2$$

- Linearized Penalty Functions (Lagrange Multipliers)

Optimization Theory

Duality

Convex Optimization

Duality for Constrained Optimization

Constrained Problem Formulation (Standard Form)

$$
\begin{aligned}
\text{minimize} \quad & f(\mathbf{x}) \\
\text{subject to} \quad & g_i(\mathbf{x}) \le 0,\ i = 1, \ldots, m \\
& h_i(\mathbf{x}) = 0,\ i = 1, \ldots, p
\end{aligned}
$$

- $f(\mathbf{x})$: objective function
- $g_i(\mathbf{x})$: inequality constraint functions
- $h_i(\mathbf{x})$: affine equality constraint functions, $h_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} - b_i$

Lagrange Multipliers

Primal Optimization Problem

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \ i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, \ i = 1, \ldots, p \end{aligned}$$

Unconstrained Problem

$$\text{minimize } f(\mathbf{x}) + \sum_{i=1}^{m} I_{\text{-}}(g_i(\mathbf{x})) + \sum_{i=1}^{p} I_0(h_i(\mathbf{x}))$$

- $I_{\text{-}}(u) := \begin{cases} 0 & u \leq 0 \\ \infty & u > 0 \end{cases}$
- $I_0(u) := \begin{cases} 0 & u = 0 \\ \infty & u \neq 0 \end{cases}$

Lagrange Multipliers

Unconstrained Problem

$$\text{minimize } f(\mathbf{x}) + \sum_{i=1}^{m} I_{\text{-}}(g_i(\mathbf{x})) + \sum_{i=1}^{p} I_0(h_i(\mathbf{x}))$$

I_0 and $I_{\text{-}}$ penalize perturbations with violating constraints by "*brick wall*" penalty functions.

We can approximate $I_{\text{-}}(u)$ linearly with $\lambda_i u, \lambda_i \geq 0$, and $I_0(u)$ with $\nu_i u$:

Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x})$$

Lagrange Multipliers

Linear approximation: $I\_(u) \approx \lambda_i u, \lambda_i \geq 0$, and $I_0(u) \approx \nu_i u$.
λ_i, ν_i are called **Lagrange multipliers**.

Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x})$$

Lagrange **dual function**

$$d(\boldsymbol{\lambda}, \boldsymbol{\nu}) := \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \quad \in \mathbb{R}$$

Since $\lambda_i u \leq I\_(u)$ and $\nu_i u \leq I_0(u)$ for all u:

- The value of the dual function is always a lower bound on the primal value $f(\mathbf{x})$ of any feasible $\mathbf{x}$.

Visual Example (Lower bound from a dual feasible point)

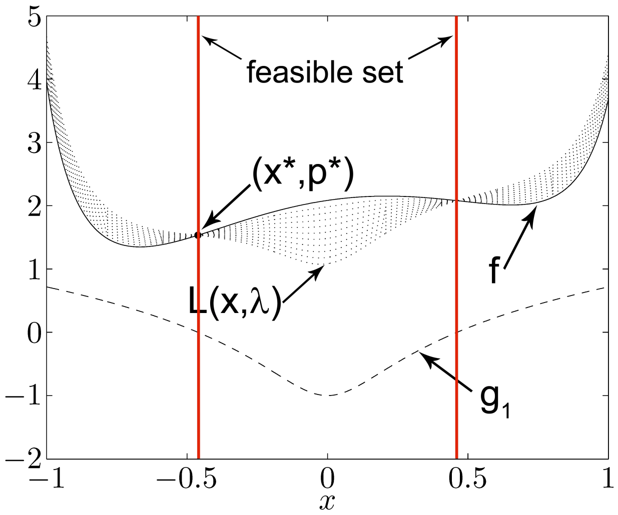


*Figure 5.1 from S. Boyd, L. Vandenberghe

The solid curve shows the objective function f, and the dashed curve shows the constraint function g_1. The feasible set is the interval $[-0.46, 0, 46]$, which is indicated by the two dotted vertical lines. The optimal point and value are $\mathbf{x}^\star = -0.46, f(\mathbf{x}^\star) = 1.54$ (shown as the black dot). The dotted curves show $L(\mathbf{x}, \boldsymbol{\lambda})$ for $\lambda = 0.1, 0.2, \ldots, 1.0$. Each of these has minimum value smaller than $f(\mathbf{x}^\star)$, since on the feasible set (and for $\lambda \geq 0$) we have $L(\mathbf{x}, \boldsymbol{\lambda}) \leq f(\mathbf{x})$.

Visual Example (The dual function $d(\boldsymbol{\lambda})$ is concave)

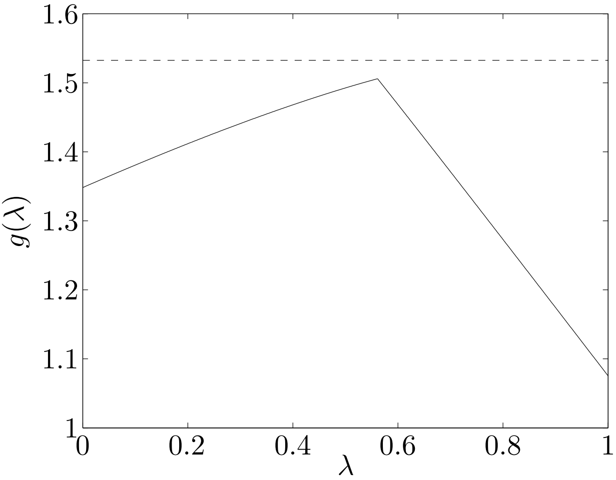


*Figure 5.2 from S. Boyd, L. Vandenberghe

The dual function $d(\boldsymbol{\lambda})$ for the problem. Neither f nor g_1 is convex, but the dual function is concave. The horizontal dashed line shows $f(\mathbf{x}^\star)$, the optimal value of the problem.

Dual Problem

Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x})$$

Lagrange dual function

$$d(\boldsymbol{\lambda}, \boldsymbol{\nu}) := \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$$

Now find the best lower bound on the optimum $f(\mathbf{x}^\star)$:

Lagrange **dual problem**

$$\begin{aligned} \text{maximize} \quad & d(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq 0 \end{aligned}$$

Dual Problem

Lagrange dual problem

$$\begin{aligned} \text{maximize} \quad & d(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq 0 \end{aligned}$$

- It is always a lower bound on the primal value $f(\mathbf{x})$ of any feasible $\mathbf{x}$.
  $\Rightarrow$ It is a lower bound on the (unknown) solution value $f(\mathbf{x}^\star)$ of the primal problem!

Strong Duality

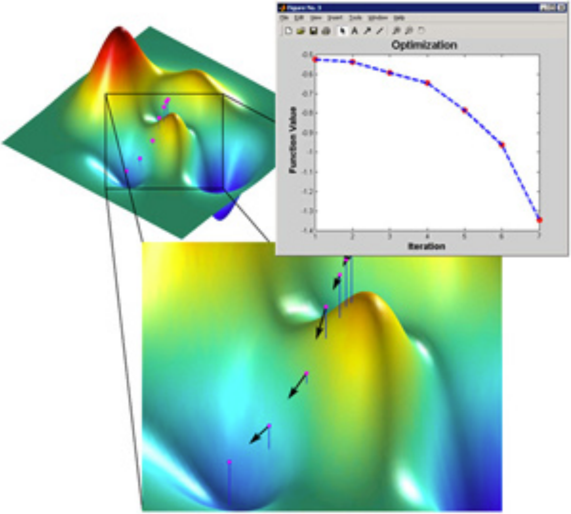
- If the primal optimization problem is **convex** *(to be defined below)*, and under *some additional conditions*, the solution value of the dual problem is *equal* to the solution value $f(\mathbf{x}^\star)$ of the primal problem.

So Everything is Fine and Well Optimized?

no!

Algorithms Can Get Stuck in Local Optima!

Example: Gradient Descent (and also the other algorithms we have seen)

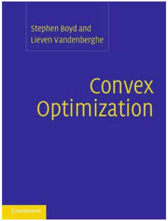


*from mathworks.com

Convex Optimization

comes to help

or: if you can't solve it, re-define the problem



($\approx 25\,000$ citations)

Convex Set

A set Q is **convex** if the line segment between any two points of Q lies in Q, i.e., if for any $\mathbf{x}, \mathbf{y} \in Q$ and any θ with $0 \leq \theta \leq 1$, we have

$$\theta \mathbf{x} + (1 - \theta)\mathbf{y} \in Q.$$

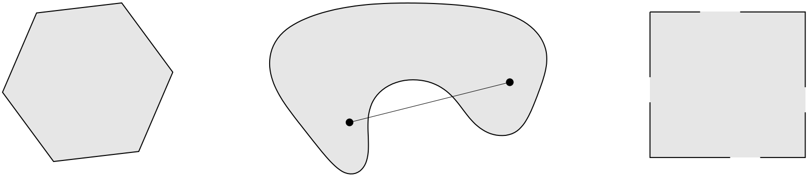


*Figure 2.2 from S. Boyd, L. Vandenberghe

Left Convex.

Middle Not convex, since line segment not in set.

Right Not convex, since some, but not all boundary points are contained in the set.

Properties of Convex Sets

- Intersections of convex sets are convex
- Projections onto convex sets are *unique*.
  (and often efficient to compute)

recall $P_Q(\mathbf{x}') := \operatorname{argmin}_{\mathbf{y}\in Q} \|\mathbf{y}-\mathbf{x}'\|$

Convex Function

Definition

A function $f : \mathbb{R}^D \to \mathbb{R}$ is **convex** if $\operatorname{dom} f$ is a convex set and if for all $\mathbf{x}, \mathbf{y} \in \operatorname{dom} f$, and θ with $0 \leq \theta \leq 1$, we have

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}).$$

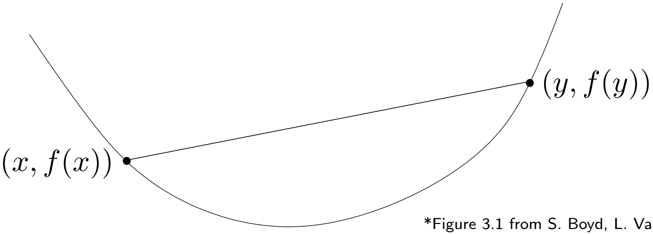


*Figure 3.1 from S. Boyd, L. Vandenberghe

Geometrically: The line segment between $(\mathbf{x}, f(\mathbf{x}))$ and $(\mathbf{y}, f(\mathbf{y}))$ lies above the graph of f.

Convex Function

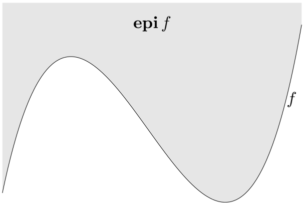
Epigraph: The *graph* of a function $f : \mathbb{R}^D \rightarrow \mathbb{R}$ is defined as

$$\{(\mathbf{x}, f(\mathbf{x})) \,|\, \mathbf{x} \in \operatorname{dom} f\},$$

The **epigraph** of a function $f : \mathbb{R}^D \rightarrow \mathbb{R}$ is defined as

$$\{(\mathbf{x}, t) \,|\, \mathbf{x} \in \operatorname{dom} f, f(\mathbf{x}) \leq t\},$$

A function is convex *iff* its epigraph is a convex set.



Convex?

*Figure 3.5 from S. Boyd, L. Vandenberghe

Convex Function

Examples of convex functions

- Linear functions: $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$
- Affine functions: $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b$
- Exponential: $f(\mathbf{x}) = e^{\alpha\mathbf{x}}$
- Norms. Every norm on $\mathbb{R}^D$ is convex.

Convexity of a norm $f(\mathbf{x})$

By the triangle inequality $f(\mathbf{x}+\mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ and homogeneity of a norm $f(a\mathbf{x}) = |a| f(\mathbf{x})$, a scalar:

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq f(\theta\mathbf{x}) + f((1-\theta)\mathbf{y}) = \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}).$$

We used the triangle inequality for the inequality and homogeneity for the equality.

Convex Optimization

Convex Optimization Problems are of the form

$$\min \; f(\mathbf{x}) \qquad \text{s.t.} \qquad \mathbf{x} \in Q$$

where both

- f is a convex function
- Q is a convex set (note: $\mathbb{R}^D$ is convex)

Properties of Convex Optimization Problems

- Every local minimum is a **global minimum**

Solving Convex Optimization Problems (provably)

For convex optimization problems, all algorithms

- Coordinate Descent
- Gradient Descent
- Stochastic Gradient Descent
- Projected Gradient Descent (projections onto convex sets do work!)

do **converge** to the global optimum! (assuming f differentiable)

Theorem: For convex problems, the **convergence rate** of the above four algorithms is proportional with $\frac{1}{t}$, i.e.

$$f(\mathbf{x}^{(t)}) - f(\mathbf{x}^\star) \leq \frac{c}{t}$$

(where $\mathbf{x}^\star$ is some optimal solution to the problem.)

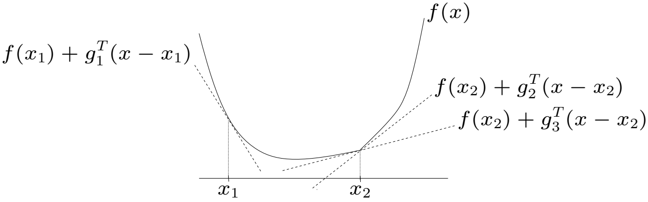
caveat: SGD rate can be $1/\sqrt{t}$ if f is not strongly convex

SubGradient Descent

What if f is not differentiable?

Subgradient: $\mathbf{g} \in \mathbb{R}^D$ is a **subgradient** of f at $\mathbf{x}$ if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{g}^\mathsf{T}(\mathbf{y} - \mathbf{x}) \qquad \text{for all } \mathbf{y}$$



Subgradient Descent: In algorithms, replace the gradient with a subgradient.

Theorem: For convex problems, the **convergence rate** of [plain or projected] subgradient descent is proportional with $\frac{1}{\sqrt{t}}$, i.e.

$$f(\mathbf{x}^{(t)}) - f(\mathbf{x}^\star) \leq \frac{c}{\sqrt{t}}$$

Optimization for Matrix Factorizations

General Formulation

$$\begin{aligned} \min_{\mathbf{U},\mathbf{Z}} \quad & f(\mathbf{U},\mathbf{Z}) \\ \text{s.t.} \quad & \mathbf{U} \in Q_1 \subseteq \mathbb{R}^{D\times K} \\ & \mathbf{Z} \in Q_2 \subseteq \mathbb{R}^{N\times K} \end{aligned}$$

and assume $f(\mathbf{U},\mathbf{Z}) = h(\mathbf{U}\mathbf{Z}^T)$ for some function $h : \mathbb{R}^{D\times N} \to \mathbb{R}$

Examples

- $f(\mathbf{U},\mathbf{Z}) = \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{Z}^T\|_F^2$,
  with $Q_1 = \mathbb{R}^{D\times K}$, $Q_2 = \mathbb{R}^{N\times K}$.

  Has an explicit solution: Singular Value Decomposition (first K singular vector pairs)

  Unfortunately, this case is **a rare exception**!

Optimization for Matrix Factorizations: Examples

- **K-means**
  $f(\mathbf{U}, \tilde{\mathbf{Z}}) = \|\mathbf{X} - \mathbf{U}\tilde{\mathbf{Z}}^T\|_F^2 = \sum_{n=1}^{N} \sum_{k=1}^{K} \tilde{Z}_{nk} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2$
  with $Q_1 = \mathbb{R}^{D \times K}$,
  $Q_2 = \{\tilde{\mathbf{Z}} \in \mathbb{R}_{\geq \mathbf{0}}^{N \times K} \mid \sum_k \tilde{Z}_{nk} = 1,\ \tilde{\mathbf{z}}_{:k}^{\mathsf{T}} \tilde{\mathbf{z}}_{:h} = 0 \text{ for } k \neq h\}$.

- **Non-Negative Matrix Factorizations**
  $f(\mathbf{U}, \mathbf{Z}) = ...$
  with $Q_1 = \mathbb{R}_{\geq \mathbf{0}}^{D \times K}$,
  $Q_2 = \mathbb{R}_{\geq \mathbf{0}}^{N \times K}$.

- **Collaborative Filtering / Matrix Completion**
  $f(\mathbf{U}, \mathbf{Z}) = \frac{1}{|\Omega|} \sum_{(d,n) \in \Omega} \frac{1}{2} \left[\mathbf{X}_{dn} - (\mathbf{U}\mathbf{Z}^T)_{dn}\right]^2$
  with $Q_1 = \mathbb{R}^{D \times K}$,
  $Q_2 = \mathbb{R}^{N \times K}$.

  where Ω is the set of observed ratings

Matrix Factorizations are Typically Non-Convex

Even if we are given a **convex** objective function

$$h(\mathbf{W}) \quad : \quad \mathbb{R}^{D \times N} \to \mathbb{R},$$

the same objective function in its factorized form

$$f(\mathbf{U}, \mathbf{Z}) := h(\mathbf{U}\mathbf{Z}^T) \quad : \quad \mathbb{R}^{(D+N) \times K} \to \mathbb{R}$$

is typically **not convex** (in its complete argument $(\mathbf{U}, \mathbf{Z})$).

Proof:
Identity function $h(w) := w$, and $D = N = 1$. The resulting objective $f(u, z) = uz$ is a saddle function over its two variables.

Alternating Minimization

$$\min_{\mathbf{U}\in Q_1, \mathbf{Z}\in Q_2} f(\mathbf{U}, \mathbf{Z})$$

Idea:

... remember coordinate descent ...

for t = 0:maxIter **do**

update

$$\mathbf{U}^{(t+1)} \leftarrow \underset{\mathbf{U}\in Q_1}{\operatorname{argmin}} \quad f(\mathbf{U}, \mathbf{Z}^{(t)})$$

$$\mathbf{Z}^{(t+1)} \leftarrow \underset{\mathbf{Z}\in Q_2}{\operatorname{argmin}} \quad f(\mathbf{U}^{(t+1)}, \mathbf{Z})$$

end for

Hardt, M. (2013). Understanding Alternating Minimization for Matrix Completion.

Alternating Minimization

- Often, while the original optimization problem might be non-convex, the two subproblems in the algorithm (w.r.t. $\mathbf{U}$ and $\mathbf{Z}$ separately) can be convex.

- Many **Algorithm Variants**:
  - In each step, optimize only over smaller parts of $\mathbf{U}$ and $\mathbf{Z}$ respectively.
  - [Stochastic] Gradient steps on the parts, instead of perfect optimization
    (Winner of the Netflix Prize Competition)

    related story: Simon Funk, 2006, Blog Post "Netflix Update: Try This at Home"

Reading Material

If you want to learn more about optimization:

S. Boyd, L. Vandenberghe: **Convex Optimization**.
Cambridge Univ. Press, (2004).
Mostly chapters 4 and 5. It's free:
`http://www.stanford.edu/~boyd/cvxbook/`.