Optimization for Machine Learning
CS-439

Lecture 10: Coordinate Descent

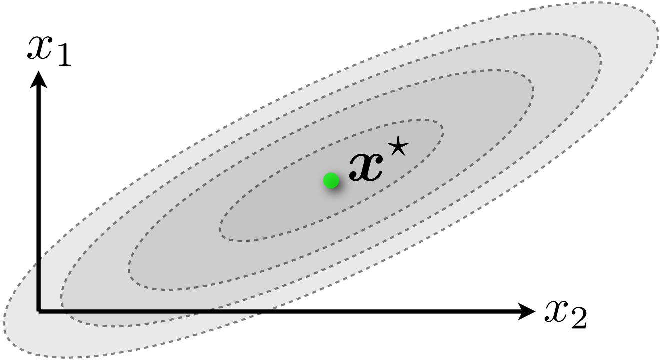
Martin Jaggi

EPFL – `github.com/epfml/OptML_course`

May 10, 2019

Coordinate Descent

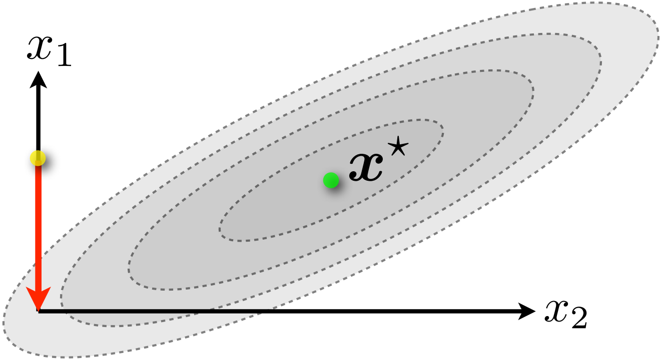
Goal: Find $\mathbf{x}^\star \in \mathbb{R}^d$ minimizing $f(\mathbf{x})$. (Example: $d = 2$)



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

Coordinate Descent

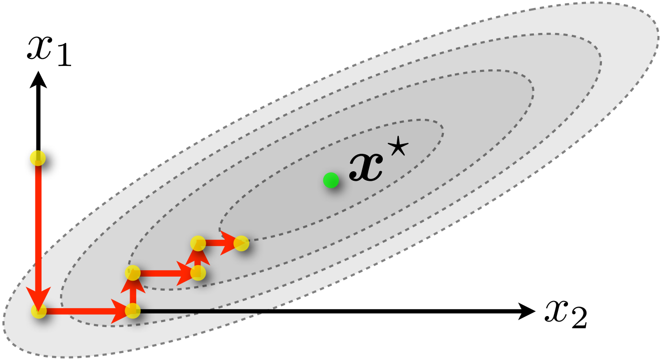
Goal: Find $\mathbf{x}^\star \in \mathbb{R}^d$ minimizing $f(\mathbf{x})$.



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

Modify only one coordinate per step:

$$\text{select } i_t \in [d]$$
$$\mathbf{x}_{t+1} := \mathbf{x}_t + \gamma \mathbf{e}_{i_t}$$

Two main variants:

- Gradient-based step-size:

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \frac{1}{L} \nabla_{i_t} f(\mathbf{x}_t)\, \mathbf{e}_{i_t}$$

- Exact coordinate minimization: solve the single-variable minimization $\operatorname{argmin}_{\gamma \in \mathbb{R}} f(\mathbf{x}_t + \gamma \mathbf{e}_{i_t})$ in closed form.

Randomized Coordinate Descent

select $i_t \in [d]$ uniformly at random

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \frac{1}{L} \nabla_{i_t} f(\mathbf{x}_t)\, \mathbf{e}_{i_t}$$

- Faster convergence than gradient descent
  (if coordinate step is significantly cheaper than full gradient step)

Convergence Analysis

Assume coordinate-wise smoothness:

$$f(\mathbf{x} + \gamma \mathbf{e}_i) \leq f(\mathbf{x}) + \gamma \nabla_i f(\mathbf{x}) + \frac{L}{2}\gamma^2 \qquad \forall \mathbf{x} \in \mathbb{R}^d,\ \forall \gamma \in \mathbb{R},\ \forall i$$

Is equivalent to coordinate-wise Lipschitz gradient:
$|\nabla_i f(\mathbf{x} + \gamma \mathbf{e}_i) - \nabla_i f(\mathbf{x})| \leq L|\gamma|, \quad \forall \mathbf{x} \in \mathbb{R}^d,\ \forall \gamma \in \mathbb{R},\ \forall i.$

- Additionally assume strong convexity

Convergence Analysis: Linear Rate

Theorem

Let f be coordinate-wise smooth with constant L, and strongly convex with parameter $\mu > 0$. Then, coordinate descent with a step-size of $1/L$,

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \tfrac{1}{L} \nabla_{i_t} f(\mathbf{x}_t)\, \mathbf{e}_{i_t} \,.$$

when choosing the active coordinate i_t uniformly at random, has an expected linear convergence rate of

$$\mathbb{E}\big[f(\mathbf{x}_t) - f^\star\big] \leq \left(1 - \frac{\mu}{dL}\right)^t [f(\mathbf{x}_0) - f^\star].$$

Convergence Proof

Proof.

Plugging the update rule, into the smoothness condition, we have

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L}|\nabla_{i_t} f(\mathbf{x}_t)|^2.$$

Take expectation with respect to i_t:

$$\begin{aligned}
\mathbb{E}\left[f(\mathbf{x}_{t+1})\right] &\leq f(\mathbf{x}_t) - \frac{1}{2L}\mathbb{E}\left[|\nabla_{i_t} f(\mathbf{x}_t)|^2\right] \\
&= f(\mathbf{x}_t) - \frac{1}{2L}\frac{1}{d}\sum_i |\nabla_i f(\mathbf{x}_t)|^2 \\
&= f(\mathbf{x}_t) - \frac{1}{2dL}\|\nabla f(\mathbf{x}_t)\|^2.
\end{aligned}$$

[**Lemma:** strongly convex f satisfy **PL**: $\frac{1}{2}\|\nabla f(\mathbf{x})\|^2 \geq \mu(f(\mathbf{x}) - f^\star)\ \forall \mathbf{x}$]

Subtracting $f^\star$ from both sides, we therefore obtain

$$\mathbb{E}[f(\mathbf{x}_{t+1}) - f^\star] \leq \left(1 - \frac{\mu}{dL}\right)[f(\mathbf{x}_t) - f^\star].$$

□

The Polyak-Lojasiewicz Condition

Definition: f satisfies the Polyak-Lojasiewicz Inequality (PL) if the following holds for some $\mu > 0$,

$$\frac{1}{2}\|\nabla f(\mathbf{x})\|^2 \geq \mu(f(\mathbf{x}) - f^\star), \quad \forall\ \mathbf{x}.$$

Lemma (Strong Convexity $\Rightarrow$ PL)

Let f be strongly convex with parameter $\mu > 0$. Then f satisfies PL for the same μ.

Proof.

For all $\mathbf{x}$ and $\mathbf{y}$ we have

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|^2 .$$

minimizing each side of the inequality with respect to $\mathbf{y}$ we obtain

$$f(\mathbf{x}^\star) \geq f(\mathbf{x}) - \frac{1}{2\mu}\|\nabla f(\mathbf{x})\|^2.$$

□

Linear Convergence without Strong Convexity

Examples satisfying PL:

- $f(\mathbf{x}) := g(A\mathbf{x})$ for strongly convex g and arbitrary matrix A, including least squares regression and many other applications in machine learning.

Linear convergence for all f satisfying the PL condition:

Corollary

For minimization of a function f which is coordinate-wise smooth with constant L, satisfies the PL inequality, and has a non-empty solution set $\mathcal{X}^\star$, random coordinate descent with a step-size of $1/L$ has the expected linear convergence rate of

$$\mathbb{E}[f(\mathbf{x}_t) - f^\star] \leq \left(1 - \frac{\mu}{dL}\right)^t [f(\mathbf{x}_0) - f^\star].$$

Importance Sampling

Uniformly random selection is not always best!

- individual smoothness constants L_i for each coordinate i

$$f(\mathbf{x} + \gamma \mathbf{e}_i) \leq f(\mathbf{x}) + \gamma \nabla_i f(\mathbf{x}) + \frac{L_i}{2}\gamma^2$$

Coordinate descent using this modified selection probabilities $P[i_t = i] = \frac{L_i}{\sum_i L_i}$, and using a step-size of $1/L_{i_t}$ converges (Exercise 54) with the faster rate of

$$\mathbb{E}[f(\mathbf{x}_t) - f^\star] \leq \left(1 - \frac{\mu}{d\bar{L}}\right)^t [f(\mathbf{x}_0) - f^\star],$$

where $\bar{L} = \frac{1}{d}\sum_{i=1}^d L_i$.

Often: $\bar{L} \ll L = \max_i L_i$!

Steepest Coordinate Descent

- Coordinate selection rule

$$i_t := \operatorname*{argmax}_{i \in [d]} |\nabla_i f(\mathbf{x}_t)| \, .$$

"Greedy" or steepest coordinate descent.
Deterministic vs random.

Convergence of Steepest Coordinate Descent

Has same convergence rate as for random coordinate descent!

Use

$$\max_i |\nabla_i f(\mathbf{x})|^2 \geq \frac{1}{d} \sum_i |\nabla_i f(\mathbf{x})|^2 \, ,$$

(And: algorithm is deterministic, so no need to take expectations in the proof.)

Corollary

Steepest coordinate descent with a step-size of $1/L$ has the linear convergence rate of

$$\mathbb{E}[f(\mathbf{x}_t) - f^\star] \leq \left(1 - \frac{\mu}{dL}\right)^t [f(\mathbf{x}_0) - f^\star].$$

Faster Convergence of Steepest Coordinate Descent

Faster convergence can be obtained for this algorithm when the strong convexity of f is measured with respect to the ℓ_1-norm instead of the standard Euclidean norm, i.e.

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu_1}{2} \|\mathbf{y} - \mathbf{x}\|_1^2 .$$

Theorem

If f is coordinate-wise L-smooth, and strongly convex w.r.t. the ℓ_1-norm with parameter $\mu_1 > 0$, steepest coordinate descent with a step-size of $1/L$ has the linear convergence rate of

$$\mathbb{E}[f(\mathbf{x}_t) - f^\star] \leq \left(1 - \frac{\mu_1}{L}\right)^t [f(\mathbf{x}_0) - f^\star].$$

Faster Convergence of Steepest Coordinate Descent

Proof: Same as above theorem, but using the following lemma measuring the PL inequality in the ℓ_∞-norm:

Lemma

Let f be strongly convex w.r.t. the ℓ_1-norm with parameter $\mu_1 > 0$. Then f satisfies

$$\frac{1}{2}\|\nabla f(\mathbf{x})\|_\infty^2 \geq \mu_1(f(\mathbf{x}) - f^\star).$$

(Proof: omitted)

Non-smooth objectives

Have proved everything for smooth f. What about non-smooth?

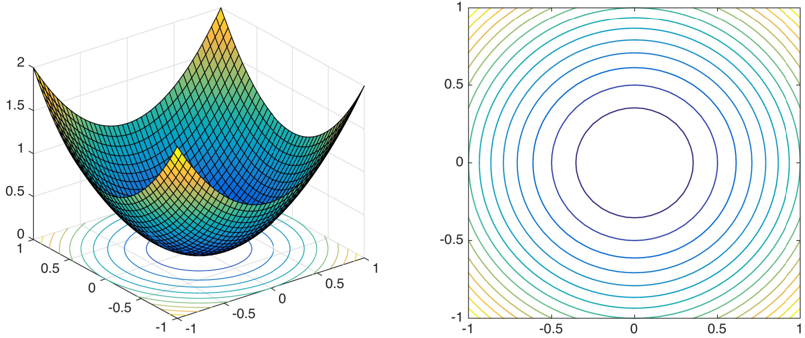


Figure: A smooth function: $f(\mathbf{x}) := \|\mathbf{x}\|^2$.

figure by Alp Yurtsever & Volkan Cevher, EPFL

Non-smooth objectives

For general non-smooth f, coordinate descent fails: gets permanently stuck:

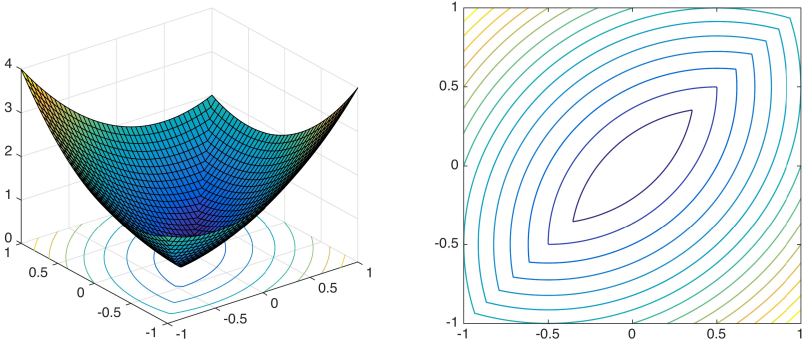


Figure: A non-smooth function: $f(\mathbf{x}) := \|\mathbf{x}\|^2 + |x_1 - x_2|$.

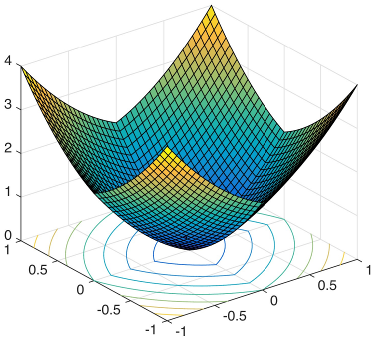
figure by Alp Yurtsever & Volkan Cevher, EPFL

Non-smooth separable objectives

What if the non-smooth part is separable over the coordinates?

$$f(\mathbf{x}) := g(\mathbf{x}) + h(\mathbf{x}) \quad \text{with } h(\mathbf{x}) = \sum_i h_i(x_i)\,,$$

- global convergence!



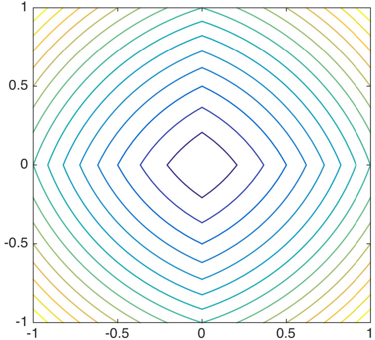


Figure: A non-smooth but separable function: $f(\mathbf{x}) := \|\mathbf{x}\|^2 + \|\mathbf{x}\|_1$.

figure by Alp Yurtsever & Volkan Cevher, EPFL

Applications

- Random coordinate descent
  - is state-of-the-art for generalized linear models $f(\mathbf{x}) := g(A\mathbf{x}) + \sum_i h_i(x_i)$.

    Regression, classification (with different regularizers)
- Steepest coordinate descent
  - Training with the help of GPUs
    (or other hardware of limited memory):

    Use steepest coordinates to decide which subset of the data A to put onto the GPU.
    $\rightarrow$ DuHL algorithm used by IBM & NVIDIA. *link1, link2*

.