Convex Optimization

Lecture 16

Today:

- Projected Gradient Descent
- Conditional Gradient Descent
- Stochastic Gradient Descent
- Random Coordinate Descent

Recall: Gradient Descent

Gradient descent algorithm:

Init	$x^{(0)} \in dom(f)$
Iterate	$x^{(k+1)} \leftarrow x^{(k)} - t^{(k)} \nabla f\left(x^{(k)}\right)$

Convergence:[1]

	#iter $\mu \preccurlyeq \nabla^2 \preccurlyeq M$	#iter $\nabla^2 \preccurlyeq M$	#iter $\lVert\nabla\rVert \le L$	$\lVert\nabla\rVert \le L$ $\mu \preccurlyeq \nabla^2$	Oracle/ops
GD	$\kappa \log 1/\epsilon$	$\frac{M\lVert x^*\rVert^2}{\epsilon}$	$\frac{L^2\lVert x^*\rVert^2}{\epsilon^2}$	$\frac{L^2}{\mu\epsilon}$	$\nabla f + O(n)$

[1] $\kappa = M/\mu$

Smoothness and Strong Convexity

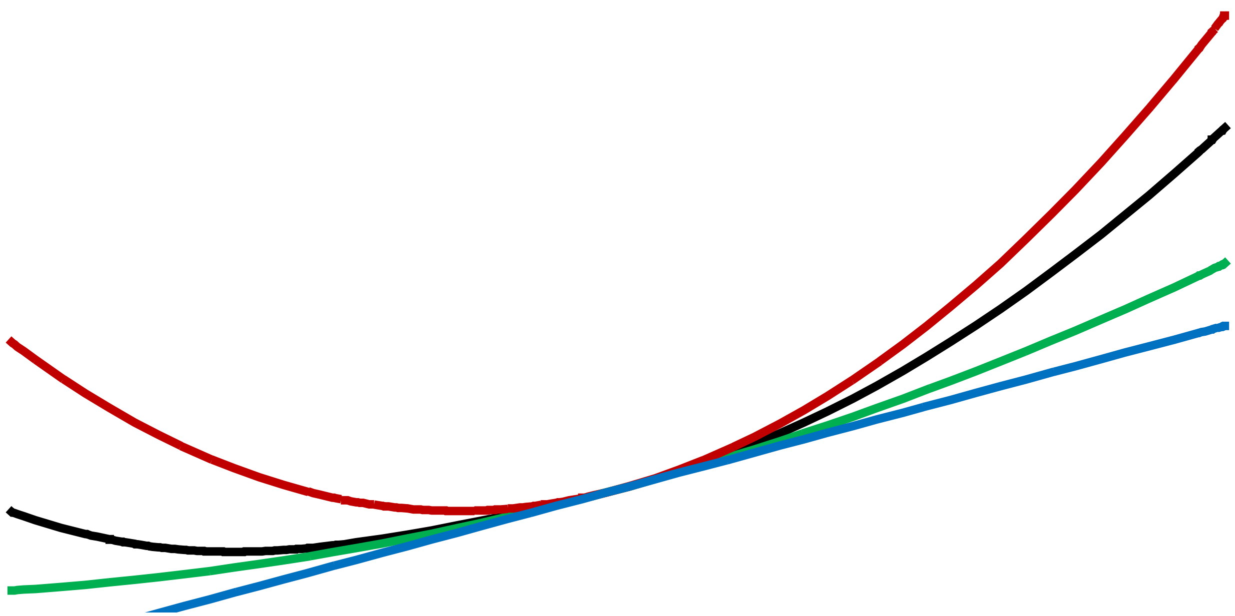
Def: f is μ-strongly convex

Def: f is M-smooth

$$f(x) + \langle \nabla f(x), \Delta x \rangle + \frac{\boldsymbol{\mu}}{2}\|\Delta x\|_2^2 \leq f(x+\Delta x) \leq f(x) + \langle \nabla f(x), \Delta x \rangle + \frac{\boldsymbol{M}}{2}\|\Delta x\|_2^2$$

Can be viewed as a condition on the directional 2nd derivatives

$$\boldsymbol{\mu} \leq f_v''(x) = \frac{\partial^2}{\partial t^2} f(x+tv) = v^\top \nabla^2 f(x) v \leq \boldsymbol{M} \quad (\text{for } \|v\|_2 = 1)$$



What about constraints?

$$\min_x f(x)$$
$$\text{s.t. } x \in \mathcal{X}$$

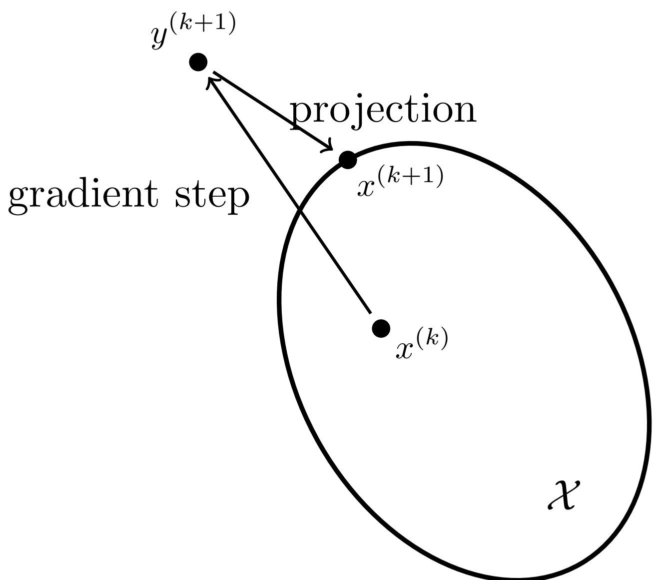
where $\mathcal{X}$ is convex

Projected Gradient Descent

Idea: make sure that points are feasible by projecting onto $\mathcal{X}$

<u>Algorithm:</u>

- $y^{(k+1)} = x^{(k)} - t^{(k)} g^{(k)}$
  where $g^{(k)} \in \partial f(x^{(k)})$
- $x^{(k+1)} = \Pi_{\mathcal{X}}(y^{(k+1)})$



The projection operator $\Pi_{\mathcal{X}}$ onto $\mathcal{X}$:

$$\Pi_{\mathcal{X}}(x) = \min_{z \in \mathcal{X}} \|x - z\|$$

Notice: subgradient instead of gradient (even for differentiable functions)

Projected gradient descent – convergence rate:

$\mu \preceq \nabla^2 \preceq M$	$\nabla^2 \preceq M$	$\lVert\nabla\rVert \leq L$	$\lVert\nabla\rVert \leq L$, $\mu \preceq \nabla^2$
$\kappa \log \frac{1}{\epsilon}$	$\frac{M\lVert x^*\rVert^2+(f(x_1)-f(x^*))}{\epsilon}$	$\frac{L^2\lVert x^*\rVert^2}{\epsilon^2}$	$\frac{L^2}{\mu\epsilon}$

Same as unconstrained case!
But, requires projection... how expensive is that?

Examples:
Euclidean ball
PSD constraints
Linear constraints $Ax \leq b$

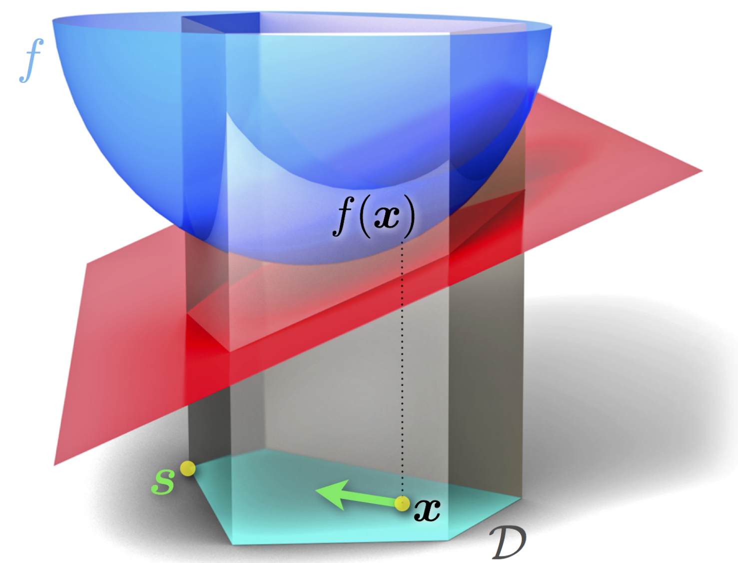
Sometimes as expensive as solving the original optimization problem!

Conditional Gradient Descent

A projection-free algorithm!
Introduced for QP by Marguerite **Frank** and Philip **Wolfe** (1956)

Algorithm

- Initialize: $x^{(0)} \in \mathcal{X}$
- $s^{(k)} = \underset{s \in \mathcal{X}}{\operatorname{argmin}} \langle \nabla f(x^{(k)}), s \rangle$
- $x^{(k+1)} = x^{(k)} + t^{(k)}(s^{(k)} - x^{(k)})$



Notice

- f assumed M-smooth
- $\mathcal{X}$ assumed bounded
- First-order oracle
- Linear optimization (in place of projection)
- Sparse iterates (e.g., for polytope constraints)

Convergence rate

For M-smooth functions with step size $t^{(k)} = \frac{2}{k+1}$:

$$\text{\# iterations required for } \epsilon\text{-optimality: } \frac{MR^2}{\epsilon}$$

where $R = \sup_{x,y\in\mathcal{X}} \|x - y\|$

Proof

$$
\begin{aligned}
f(x^{(k+1)}) \leq & f(x^{(k)}) + \langle \nabla f(x^{(k)}), x^{(k+1)} - x^{(k)} \rangle + \frac{M}{2}\|x^{(k+1)} - x^{(k)}\|^2 && \text{[smoothness]} \\
= & f(x^{(k)}) + t^{(k)} \langle \nabla f(x^{(k)}), s^{(k)} - x^{(k)} \rangle + \frac{M}{2}(t^{(k)})^2 \|s^{(k)} - x^{(k)}\|^2 && \text{[update]} \\
\leq & f(x^{(k)}) + t^{(k)} \langle \nabla f(x^{(k)}), x^* - x^{(k)} \rangle + \frac{M}{2}(t^{(k)})^2 R^2 \\
\leq & f(x^{(k)}) + t^{(k)} (f(x^*) - f(x^{(k)})) + \frac{M}{2}(t^{(k)})^2 R^2 && \text{[convexity]}
\end{aligned}
$$

Define: $\delta^{(k)} = f(x^{(k)}) - f(x^*)$, we have:

$$
\delta^{(k+1)} \leq (1 - t^{(k)})\delta^{(k)} + \frac{M(t^{(k)})^2 R^2}{2}
$$

A simple induction shows that for $t^{(k)} = \frac{2}{k+1}$:

$$
\delta^{(k)} \leq \frac{2MR^2}{k+1}
$$

Same rate as projected gradient descent, but without projection!
Does need linear optimization

What about strong convexity?

Not helpful! Does not give linear rate ($\kappa \log(1/\epsilon)$)

$\star$ Active research

Randomness in Convex Optimization

Insight: first-order methods are robust – inexact gradients are sufficient

As long as gradients are correct on average, the error will vanish

Long history (Robbins & Monro, 1951)

Stochastic Gradient Descent

Motivation

Many machine learning problems have the form of *empirical risk minimization*

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} f_i(x) + \lambda \Omega(x)$$

where f_i are convex and λ is the regularization constant

Classification: SVM, logistic regression
Regression: least-squares, ridge regression, LASSO

Cost of computing the gradient?
$m \cdot n$

What if m is VERY large?
We want cheaper iterations

Idea: Use *stochastic* first-order oracle: for each point $x \in \mathrm{dom}(f)$ returns a stochastic gradient

$$\tilde{g}(x) \quad \text{s.t. } \mathbb{E}[\tilde{g}(x)] \in \partial f(x)$$

That is, $\tilde{g}$ is an *unbiased estimator* of the subgradient

Example

$$\min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \overbrace{(f_i(x) + \lambda \Omega(x))}^{F_i(x)}$$

For this objective, select $j \in \{1, \ldots, m\}$ u.a.r. and return $\nabla F_j(x)$
Then,

$$\mathbb{E}[\tilde{g}(x)] = \frac{1}{m} \sum_i \nabla F_i(x) = \nabla f(x)$$

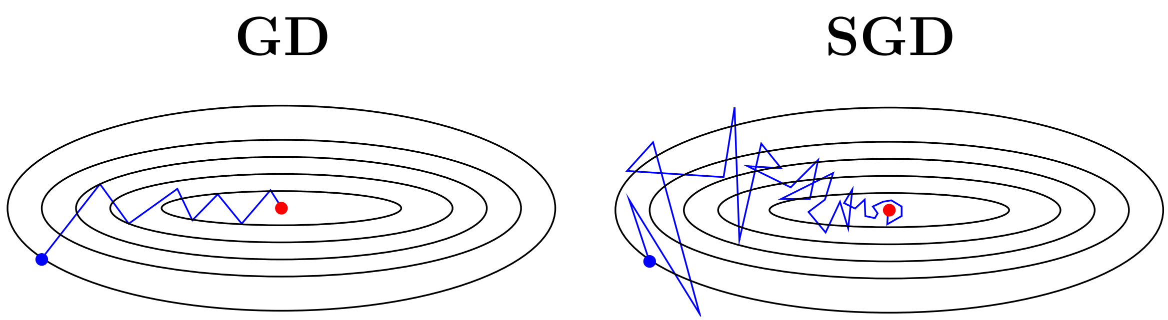
SGD iterates:

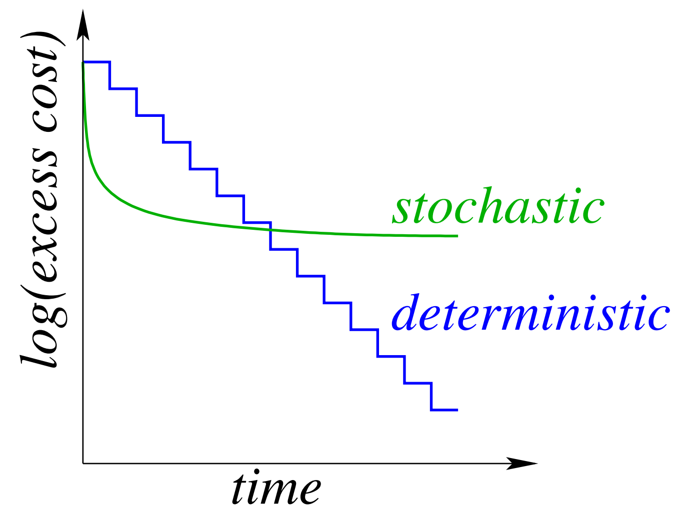
$$x^{(k+1)} \leftarrow x^{(k)} - t^{(k)} \tilde{g}(x^{(k)})$$

How to choose step size $t^{(k)}$?

- Lipschitz case: $t^{(k)} \propto \frac{1}{\sqrt{k}}$
- μ-strongly-convex case: $t^{(k)} \propto \frac{1}{\mu k}$

Note: decaying step size!





(Figures borrowed from Francis Bach's slides)

Convergence rates

	$\mu \preceq \nabla^2 \preceq M$	$\nabla^2 \preceq M$	$\|\nabla\| \leq L$	$\|\nabla\| \leq L$, $\mu \preceq \nabla^2$
GD	$\kappa \log \frac{1}{\epsilon}$	$\frac{M\|x^*\|^2}{\epsilon}$	$\frac{L^2\|x^*\|^2}{\epsilon^2}$	$\frac{L^2}{\mu\epsilon}$
SGD	?	?	$\frac{B^2\|x^*\|^2}{\epsilon^2}$	$\frac{B^2}{\mu\epsilon}$

Additional assumption: $\mathbb{E}[\|\tilde{g}(x)\|^2] \leq B^2$ for all $x \in \mathrm{dom}(f)$

Comment: holds in expectation, with averaged iterates

$$\mathbb{E}\left[f\left(\frac{1}{K}\sum_{k=1}^{K} x^{(k)}\right)\right] - f(x^*) \leq \dots$$

Similar rates as with exact gradients!

	$\mu \preceq \nabla^2 \preceq M$	$\nabla^2 \preceq M$	$\|\nabla\| \leq L$	$\|\nabla\| \leq L$, $\mu \preceq \nabla^2$
GD	$\kappa \log \frac{1}{\epsilon}$	$\frac{M\|x^*\|^2}{\epsilon}$	$\frac{L^2\|x^*\|^2}{\epsilon^2}$	$\frac{L^2}{\mu\epsilon}$
AGD	$\sqrt{\kappa} \log \frac{1}{\epsilon}$	$\frac{M\|x^*\|^2}{\sqrt{\epsilon}}$	$\times$	$\times$
SGD	?	$\frac{\|x^*\|\sigma}{\epsilon^2} + \frac{M\|x^*\|^2}{\epsilon}$	$\frac{B^2\|x^*\|^2}{\epsilon^2}$	$\frac{B^2}{\mu\epsilon}$

where $\mathbb{E}[\|\nabla f(x) - \tilde{g}(x)\|^2] \leq \sigma^2$

Smoothness?
Not helpful! (same rate as non-smooth)
Lower bounds (Nemirovski & Yudin, 1983)
$\star$ Active research

Acceleration?
Cannot be easily accelerated!
Mini-batch acceleration
$\star$ Active research

Random Coordinate Descent

Recall: cost of computing exact GD update: $m \cdot n$
What if n VERY is large?
We want cheaper iterations

Random coordinate descent algorithm:

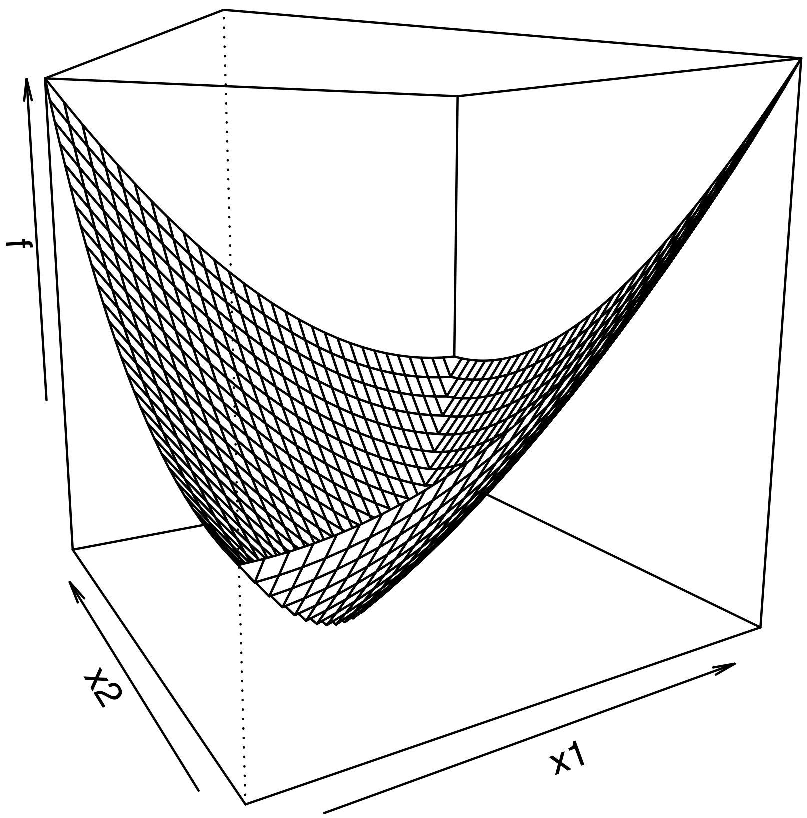
- Initialize: $x^{(0)} \in \mathrm{dom}(f)$
- Iterate: pick $i(k) \in \{1, \ldots, n\}$ randomly

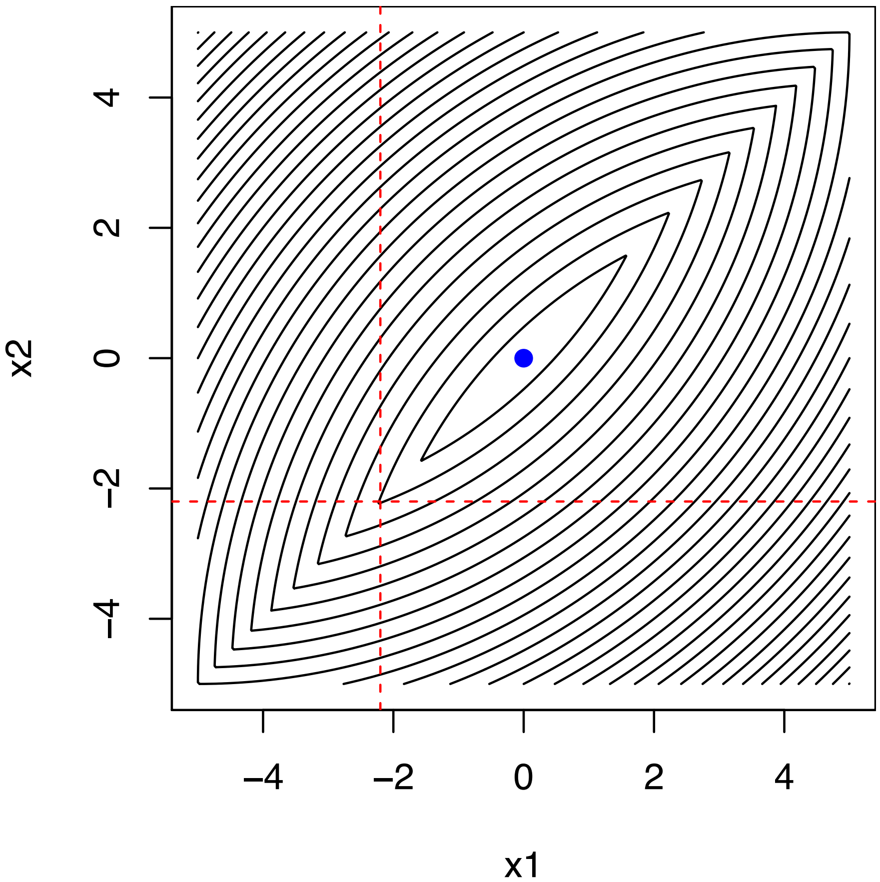
$$x^{(k+1)} = x^{(k)} - t^{(k)} \nabla_{i(k)} f(x^{(k)}) e_{i(k)}$$

where we denote: $\nabla_i f(x) = \frac{\partial f}{\partial x_i}(x)$

Assumption: f is convex and differentiable

What if f not differentiable?





(Figures borrowed from Ryan Tibshirani's slides)

Iteration cost? $\nabla_i f(x) + O(1)$
Compare to $\nabla f(x) + O(n)$ for GD

Example: quadratic

$$\begin{aligned} f(x) &= \frac{1}{2}x^\top Q x - v^\top x \\ \nabla f(x) &= Qx - v \\ \nabla_i f(x) &= q_i^\top x - v_i \end{aligned}$$

Can view CD as SGD with oracle: $\tilde{g}(x) = n\nabla_i f(x) e_i$
Clearly,

$$\mathbb{E}[\tilde{g}(x)] = \frac{1}{n} n \sum_i \nabla_i f(x) e_i = \nabla f(x)$$

Can replace individual coordinates with blocks of coordinates

Example: SVM

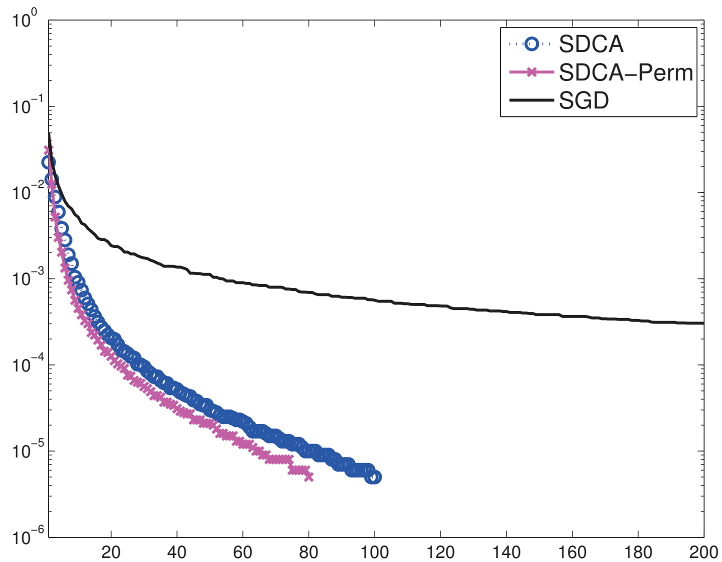
Primal:

$$\min_w \frac{\lambda}{2}\|w\|^2 + \sum_i \max(1 - y_i w^\top z_i, 0)$$

Dual:

$$\min_\alpha \frac{1}{2}\alpha^\top Q\alpha - 1^\top \alpha$$
$$\text{s.t.} \quad 0 \le \alpha_i \le 1/\lambda \quad \forall i$$

where $Q_{ij} = y_i y_j \; z_i^\top z_j$



(Shalev-Schwartz & Zhang, 2013)

Convergence rate

Directional smoothness for f: there exist $M_1, \ldots, M_n$ s.t. for any $i \in \{1, \ldots, n\}$, $x \in \mathbb{R}^n$, and $u \in \mathbb{R}$

$$|\nabla_i f(x + ue_i) - \nabla_i f(x)| \le M_i |u|$$

Note: implies f is M-smooth with $M \le \sum_i M_i$

Consider the update:

$$x^{(k+1)} = x^{(k)} - \frac{1}{M_{i(k)}} \nabla_{i(k)} f(x^{(k)}) \cdot e_{i(k)}$$

No need to know M_i's, can be adjusted dynamically

Rates (Nesterov, 2012):

	$\mu \preceq \nabla^2 \preceq M$	$\nabla^2 \preceq M$	$\|\nabla\| \leq L$	$\|\nabla\| \leq L$, $\mu \preceq \nabla^2$
GD	$\kappa \log \frac{1}{\epsilon}$	$\frac{M\|x^*\|^2}{\epsilon}$	$\frac{L^2\|x^*\|^2}{\epsilon^2}$	$\frac{L^2}{\mu\epsilon}$
CD	$n\kappa \log \frac{1}{\epsilon}$, $\kappa = \frac{\sum_i M_i}{\mu}$	$\frac{n\|x^*\|^2 \sum_i M_i}{\epsilon}$	$\times$	$\times$

Same total cost as GD, but with much cheaper iterations

Comment: holds in expectation

$$\mathbb{E}\left[f(x^{(k)})\right] - f^* \leq \dots$$

Acceleration?
Yes!
$\star$ Active research