Stochastic Optimization for Machine Learning

Shai Shalev-Shwartz

School of CS and Engineering,
The Hebrew University of Jerusalem

Nesterov's 60 Birthday, February 2016

Happy Birthday Yuri

The single paper that made the largest impact on my PhD thesis.

- Primal-dual subgradient methods for convex problems. (2005, Technical report, 2009 Math. Prog.).

Happy Birthday Yuri

The single paper that made the largest impact on my PhD thesis.

- Primal-dual subgradient methods for convex problems. (2005, Technical report, 2009 Math. Prog.).

Connections between optimization and machine learning:

- Online learning and first order methods
- Sample complexity and oracle complexity
- Covering numbers and convergence of sub-gradient descent
- Strong convexity, stability, and generalization
- PAC learning and stochastic optimization
- ...

Outline

PAC Learning as Stochastic Optimization

Goal (informal): Learn an accurate mapping $h : \mathcal{X} \to \mathcal{Y}$ based on examples $((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n$

PAC Learning as Stochastic Optimization

Goal (informal): Learn an accurate mapping $h : \mathcal{X} \to \mathcal{Y}$ based on examples $((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n$

Parametrized learning:

- Each mapping $h : \mathcal{X} \to \mathcal{Y}$ is parameterized by a weight vector $w \in \mathbb{R}^d$, so our goal is to learn the vector w
- The quality of w on example (x, y) is assessed by $\ell(w, (x, y))$

PAC Learning as Stochastic Optimization

Goal (informal): Learn an accurate mapping $h : \mathcal{X} \to \mathcal{Y}$ based on examples $((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n$

Parametrized learning:

- Each mapping $h : \mathcal{X} \to \mathcal{Y}$ is parameterized by a weight vector $w \in \mathbb{R}^d$, so our goal is to learn the vector w
- The quality of w on example (x, y) is assessed by $\ell(w, (x, y))$

PAC Learning is Stochastic Optimization:

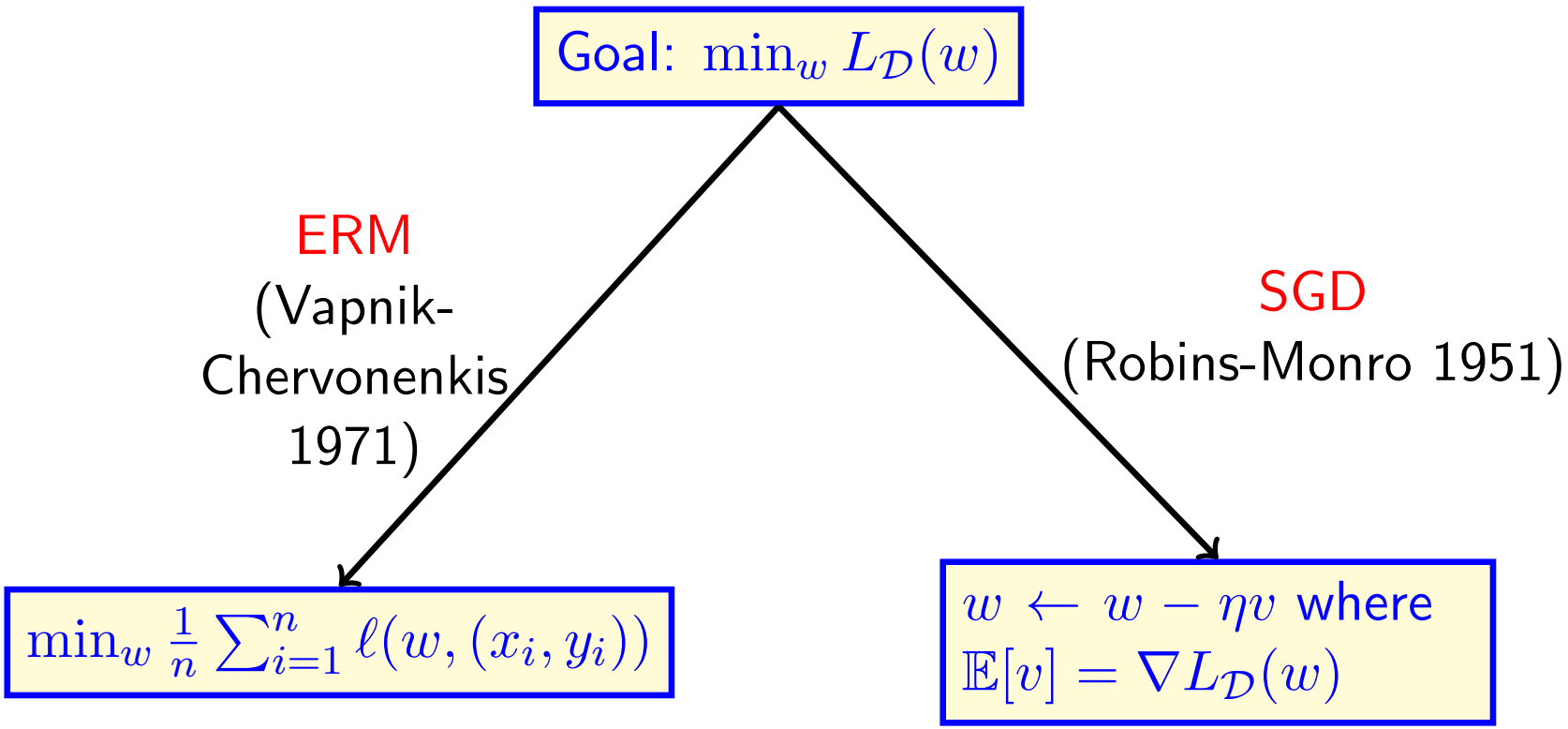
- Given distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ the goal is to approximately solve

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} \; \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} [\ell(w, (x, y))]$$

- We can only obtain i.i.d. samples from $\mathcal{D}$

How to Solve Stochastic Optimization

- Our goal: minimize over w the risk $L_{\mathcal{D}}(w) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(w,(x,y))]$



Outline

Stochastic Gradient Descent for Direct Risk Minimization

- Start with some initial w
- For $t = 1, 2, \ldots, T$
  - Sample $(x, y) \sim \mathcal{D}$
  - Update $w = w - \eta \nabla \ell(w, (x, y))$

Stochastic Gradient Descent for Direct Risk Minimization

- Start with some initial w
- For $t = 1, 2, \ldots, T$
  - Sample $(x, y) \sim \mathcal{D}$
  - Update $w = w - \eta \nabla \ell(w, (x, y))$

Theorem

Assume that ℓ is convex and ρ-Lipschitz w.r.t. w. Fix some w^*. Then, if $T \geq \Omega\left(\frac{\rho^2 \|w^*\|^2}{\epsilon^2}\right)$ we have that the average w satisfies (with constant probability)

$$L_{\mathcal{D}}(w) \;\leq\; L_D(w^*) + \epsilon \;.$$

Optimality of SGD

- A learner can be written as $A : \bigcup_n (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}^d$

Optimality of SGD

- A learner can be written as $A : \bigcup_n (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}^d$
- Sample complexity: What should be n s.t. exists A s.t. for any $\mathcal{D}$ and $W \subset \mathbb{R}^d$ we have $L_{\mathcal{D}}(A(S_n)) \leq \min_{w \in W} L_{\mathcal{D}}(w) + \epsilon$

Optimality of SGD

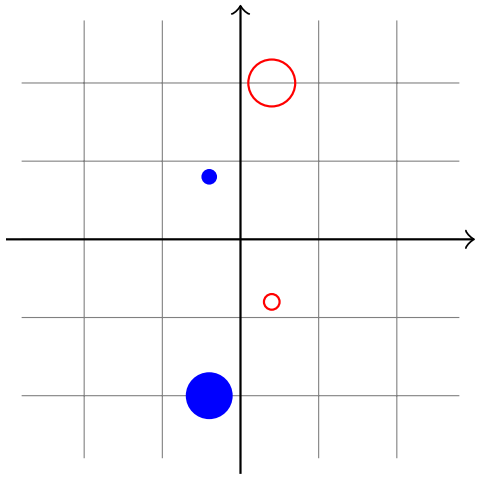
- A learner can be written as $A : \bigcup_n (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}^d$
- Sample complexity: What should be n s.t. exists A s.t. for any $\mathcal{D}$ and $W \subset \mathbb{R}^d$ we have $L_{\mathcal{D}}(A(S_n)) \leq \min_{w \in W} L_{\mathcal{D}}(w) + \epsilon$
- Claim: For $\mathcal{D} = \{w : \|w\| \leq B\}$ we must have $n \geq \Omega\left(\frac{\rho^2 B^2}{\epsilon^2}\right)$

Optimality of SGD

- A learner can be written as $A : \bigcup_n (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}^d$
- Sample complexity: What should be n s.t. exists A s.t. for any $\mathcal{D}$ and $W \subset \mathbb{R}^d$ we have $L_{\mathcal{D}}(A(S_n)) \leq \min_{w \in W} L_{\mathcal{D}}(w) + \epsilon$
- Claim: For $\mathcal{D} = \{w : \|w\| \leq B\}$ we must have $n \geq \Omega\left(\frac{\rho^2 B^2}{\epsilon^2}\right)$
- Conclusion: SGD is optimal (one pass over the data)

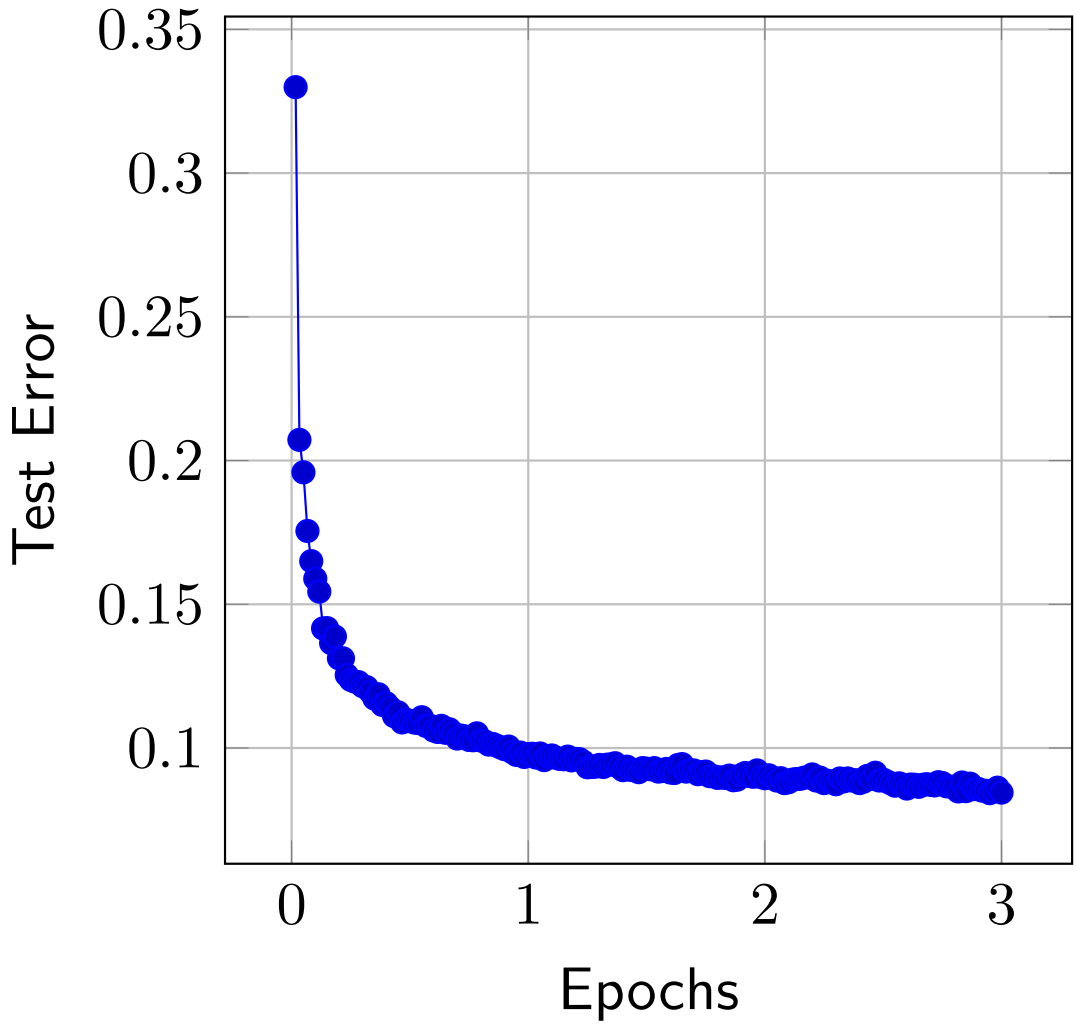
Outline

Worst case doesn't tell the whole story



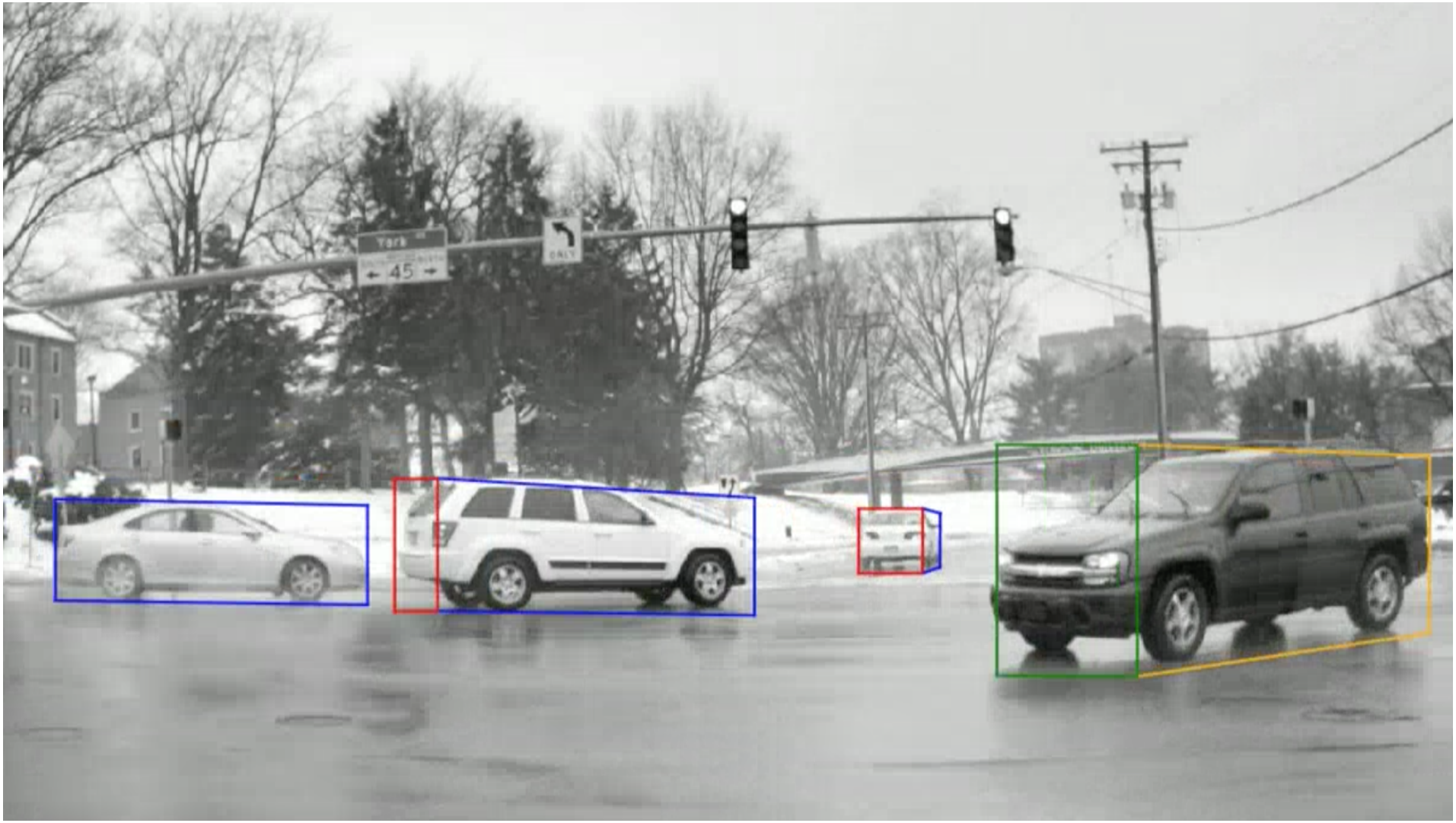
- Probability of small circles is ϵ, margin is γ
- Claim: SGD (with every η) requires $\Omega\left(\frac{1}{\gamma\epsilon}\right)$ iterations
- Claim: Sample complexity of ERM is $O\left(\frac{1}{\epsilon}\right)$

Worst case doesn't tell the whole story

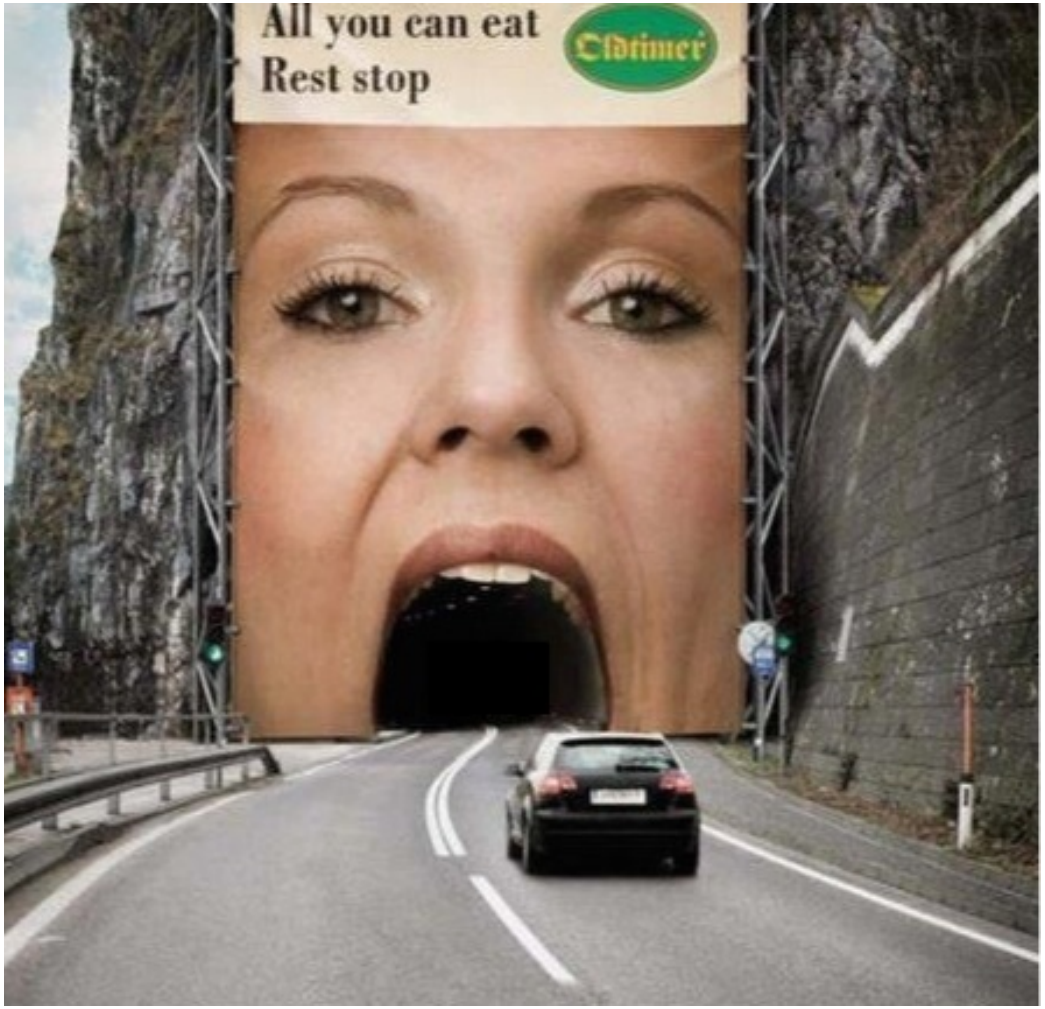


What is the true objective ?

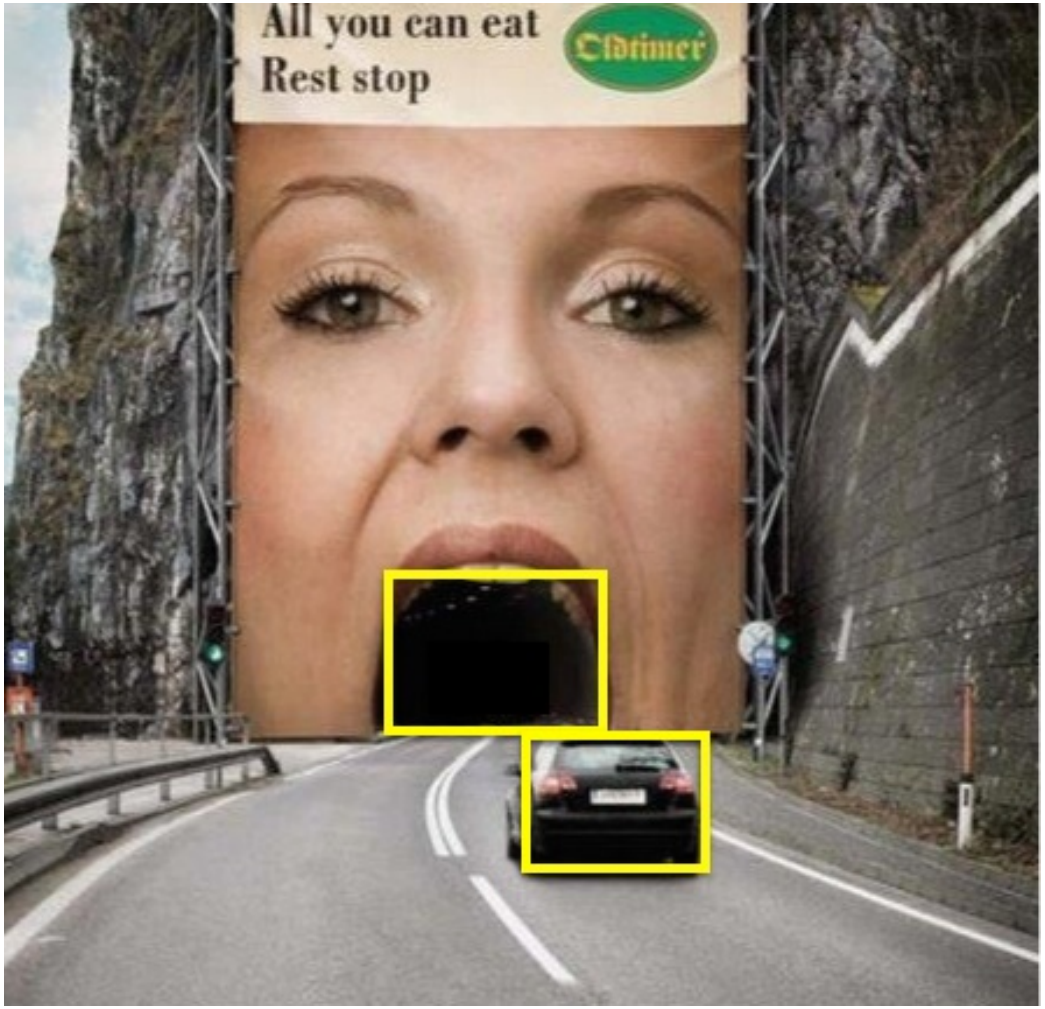
What is the true objective ?



What is the true objective ?



What is the true objective ?



What is the true objective ?

PAC learning with train/test mismatch:

- Consider two distributions $\mathcal{D}_1, \mathcal{D}_2$

What is the true objective ?

PAC learning with train/test mismatch:

- Consider two distributions $\mathcal{D}_1, \mathcal{D}_2$
- Goal: be good on both of them

What is the true objective ?

PAC learning with train/test mismatch:

- Consider two distributions $\mathcal{D}_1, \mathcal{D}_2$
- Goal: be good on both of them
- Our training set is sampled i.i.d. from $\mathcal{D} = \lambda_1 \mathcal{D}_1 + \lambda_2 \mathcal{D}_2$, $\lambda_1 \gg \lambda_2$

What is the true objective ?

PAC learning with train/test mismatch:

- Consider two distributions $\mathcal{D}_1, \mathcal{D}_2$
- Goal: be good on both of them
- Our training set is sampled i.i.d. from $\mathcal{D} = \lambda_1 \mathcal{D}_1 + \lambda_2 \mathcal{D}_2$, $\lambda_1 \gg \lambda_2$
- What is the sample complexity of ERM ?

What is the true objective ?

PAC learning with train/test mismatch:

- Consider two distributions $\mathcal{D}_1, \mathcal{D}_2$
- Goal: be good on both of them
- Our training set is sampled i.i.d. from $\mathcal{D} = \lambda_1 \mathcal{D}_1 + \lambda_2 \mathcal{D}_2$, $\lambda_1 \gg \lambda_2$
- What is the sample complexity of ERM ?
  - Naive analysis: $\mathrm{VC}(H)/(\lambda_2 \epsilon)$

What is the true objective ?

PAC learning with train/test mismatch:

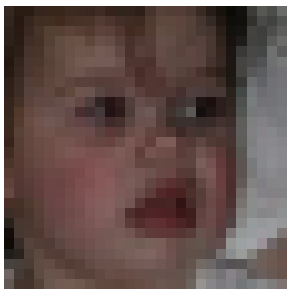
- Consider two distributions $\mathcal{D}_1, \mathcal{D}_2$
- Goal: be good on both of them
- Our training set is sampled i.i.d. from $\mathcal{D} = \lambda_1 \mathcal{D}_1 + \lambda_2 \mathcal{D}_2$, $\lambda_1 \gg \lambda_2$
- What is the sample complexity of ERM ?
  - Naive analysis: $\mathrm{VC}(H)/(\lambda_2 \epsilon)$
  - Refined analysis in the next slides

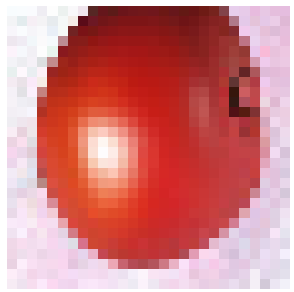
What is the true objective ?

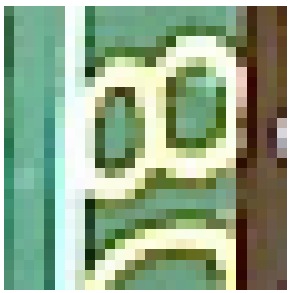
PAC learning with train/test mismatch:

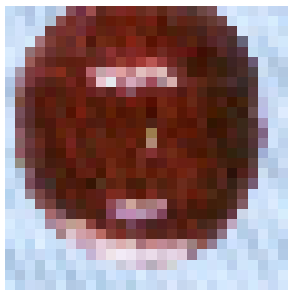
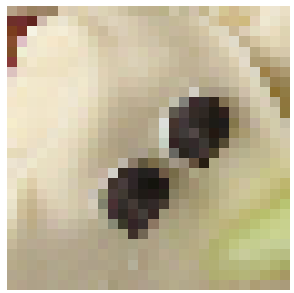
- Consider two distributions $\mathcal{D}_1, \mathcal{D}_2$
- Goal: be good on both of them
- Our training set is sampled i.i.d. from $\mathcal{D} = \lambda_1 \mathcal{D}_1 + \lambda_2 \mathcal{D}_2$, $\lambda_1 \gg \lambda_2$
- What is the sample complexity of ERM ?
  - Naive analysis: $\mathrm{VC}(H)/(\lambda_2 \epsilon)$
  - Refined analysis in the next slides
- How many SGD iterations are required ?

Typical vs. Rare distributions



Refined Sample Complexity Analysis

- Naive analysis: We need $\mathrm{VC}(H)/\epsilon$ from D_2 and the averaged number of examples from D_2 is $\lambda_2 n$. Therefore, we need $n \geq \mathrm{VC}(H)/(\lambda_2 \epsilon)$
- How to improve:
  - Use examples from D_1 to decrease the term $\mathrm{VC}(H)$
  - The $1/\epsilon$ term in the lower bound comes from a peculiar distribution. Can it be eliminated ?

Refined Sample Complexity Analysis

Theorem

Define

- $\mathcal{H}_{1,\epsilon} = \{h \in \mathcal{H} : L_{D_1}(h) \leq \epsilon\}$
- $c = \max\{c' \in [\epsilon, 1) : \forall h \in \mathcal{H}_{1,\epsilon},\ L_{D_2}(h) \leq c' \Rightarrow L_{D_2}(h) \leq \epsilon\}$.

Then, sample complexity is order of

$$\frac{\mathrm{VC}(\mathcal{H})}{\epsilon} + \frac{\mathrm{VC}(\mathcal{H}_{1,\epsilon})}{c\,\lambda_2}$$

Refined Sample Complexity Analysis

Theorem

Define

- $\mathcal{H}_{1,\epsilon} = \{h \in \mathcal{H} : L_{D_1}(h) \leq \epsilon\}$
- $c = \max\{c' \in [\epsilon, 1) : \forall h \in \mathcal{H}_{1,\epsilon},\ L_{D_2}(h) \leq c' \Rightarrow L_{D_2}(h) \leq \epsilon\}$.

Then, sample complexity is order of

$$\frac{\mathrm{VC}(\mathcal{H})}{\epsilon} + \frac{\mathrm{VC}(\mathcal{H}_{1,\epsilon})}{c\,\lambda_2}$$

Proof idea:

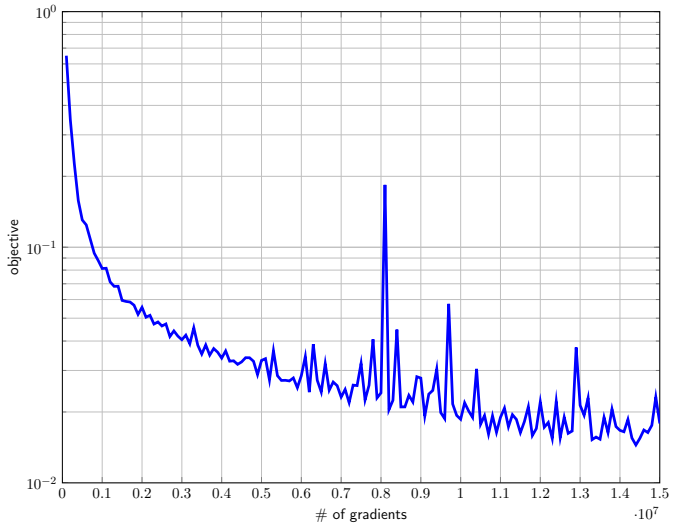
- Think about ERM as two steps: (1) find $\mathcal{H}_{1,\epsilon}$ based on examples from D_1 (2) find a hypothesis within $\mathcal{H}_{1,\epsilon}$ that is good on the examples from D_2
- "Shell analysis" (Haussler-Kearns-Seung-Tishby 1996) for the second step

Outline

The ERM problem

$$\min_{w \in \mathbb{R}^d} P(w) := \frac{1}{n} \sum_{i=1}^{n} \phi_i(w)$$

Why SGD is slow at the end?



- Rare mistakes: Suppose all but 1% of the examples are correctly classified. SGD will now waste 99% of its time on examples that are already correct by the model
- High variance, even close to the optimum

Solving ERM for Classification Problems

$$\min_{w \in \mathbb{R}^d} P(w) := \frac{1}{n} \sum_{i=1}^{n} \phi_i(w)$$

- $\phi_i(w) = 1[h_w(x_i) \neq y_i]$ (non-convex, non-continuos).
- Assumption: There exists an online learner for w with a mistake bound C

The Mistake Bound Model (Littlestone 1988)

- The Online Game: At each round t, learner picks w_t, adversary responds with i_t, and learner pays $\phi_{i_t}(w_t) = 1[h_{w_t}(x_{i_t}) \neq y_{i_t}]$

The Mistake Bound Model (Littlestone 1988)

- The Online Game: At each round t, learner picks w_t, adversary responds with i_t, and learner pays $\phi_{i_t}(w_t) = 1[h_{w_t}(x_{i_t}) \neq y_{i_t}]$
- Mistake Bound: The learner enjoys a mistake bound C if for any T and any sequence $i_1, \ldots, i_T$, it makes at most T mistakes

The Mistake Bound Model (Littlestone 1988)

- The Online Game: At each round t, learner picks w_t, adversary responds with i_t, and learner pays $\phi_{i_t}(w_t) = 1[h_{w_t}(x_{i_t}) \neq y_{i_t}]$
- Mistake Bound: The learner enjoys a mistake bound C if for any T and any sequence $i_1, \ldots, i_T$, it makes at most T mistakes
- Example: The Perceptron (Rosenblatt 1958):

The Mistake Bound Model (Littlestone 1988)

- The Online Game: At each round t, learner picks w_t, adversary responds with i_t, and learner pays $\phi_{i_t}(w_t) = 1[h_{w_t}(x_{i_t}) \neq y_{i_t}]$
- Mistake Bound: The learner enjoys a mistake bound C if for any T and any sequence $i_1, \ldots, i_T$, it makes at most T mistakes
- Example: The Perceptron (Rosenblatt 1958):
  - $h_w(x) = \text{sign}(\langle w, x \rangle)$, $y \in \{\pm 1\}$

The Mistake Bound Model (Littlestone 1988)

- The Online Game: At each round t, learner picks w_t, adversary responds with i_t, and learner pays $\phi_{i_t}(w_t) = 1[h_{w_t}(x_{i_t}) \neq y_{i_t}]$
- Mistake Bound: The learner enjoys a mistake bound C if for any T and any sequence $i_1, \ldots, i_T$, it makes at most T mistakes
- Example: The Perceptron (Rosenblatt 1958):
  - $h_w(x) = \text{sign}(\langle w, x \rangle)$, $y \in \{\pm 1\}$
  - The Perceptron rule: $w_{t+1} = w_t + \phi_{i_t}(w_t)\, x_{i_t} / \|x_{i_t}\|$

The Mistake Bound Model (Littlestone 1988)

- The Online Game: At each round t, learner picks w_t, adversary responds with i_t, and learner pays $\phi_{i_t}(w_t) = 1[h_{w_t}(x_{i_t}) \neq y_{i_t}]$
- Mistake Bound: The learner enjoys a mistake bound C if for any T and any sequence $i_1, \ldots, i_T$, it makes at most T mistakes
- Example: The Perceptron (Rosenblatt 1958):
  - $h_w(x) = \text{sign}(\langle w, x \rangle)$, $y \in \{\pm 1\}$
  - The Perceptron rule: $w_{t+1} = w_t + \phi_{i_t}(w_t)\, x_{i_t} / \|x_{i_t}\|$
  - Theorem (Agmon 1954, Minsky, Papert 1969):
    If exists w^* s.t. for every i, $y_i \langle w^*, x_i \rangle / \|x_i\| \geq 1$, then Perceptron's mistake bound is $C = \|w^*\|^2$

Solving ERM for Classification Problems

$$\min_{w \in \mathbb{R}^d} P(w) := \frac{1}{n} \sum_{i=1}^{n} \phi_i(w)$$

Solving ERM for Classification Problems

$$\min_{w \in \mathbb{R}^d} P(w) := \frac{1}{n} \sum_{i=1}^{n} \phi_i(w)$$

Naive approach I:

- Apply the online learner with random examples from $[n]$
- Analysis: error decreases as $\frac{C}{T}$
- Runtime for zero error: Need $C/T < 1/n$ so $T > n\,C\,d$

Solving ERM for Classification Problems

$$\min_{w \in \mathbb{R}^d} P(w) := \frac{1}{n} \sum_{i=1}^{n} \phi_i(w)$$

Naive approach I:

- Apply the online learner with random examples from $[n]$
- Analysis: error decreases as $\frac{C}{T}$
- Runtime for zero error: Need $C/T < 1/n$ so $T > n\,C\,d$

Naive approach II:

- Apply the online learner while feeding it with the worst example
- Runtime for zero error: Need C iterations, each cost $d\,n$, so $T > n\,C\,d$

Solving ERM for Classification Problems

$$\min_{w \in \mathbb{R}^d} P(w) := \frac{1}{n} \sum_{i=1}^{n} \phi_i(w)$$

Naive approach I:

- Apply the online learner with random examples from $[n]$
- Analysis: error decreases as $\frac{C}{T}$
- Runtime for zero error: Need $C/T < 1/n$ so $T > n\,C\,d$

Naive approach II:

- Apply the online learner while feeding it with the worst example
- Runtime for zero error: Need C iterations, each cost $d\,n$, so $T > n\,C\,d$

Our approach: runtime is $(n + C)\,d$

Our Approach: Focused Online Learning

Min-max problem:

$$\min_{w} \max_{p \in \mathbb{S}_n} \sum_{i=1}^{n} p_i \phi_i(w)$$

- Zero-sum game between w player and p player
- Use the online learner for the w player
- Use a variant of EXP3 (Auer, Cesa-Bianchi, Freund, Schapire, 2002) for the p player
- Our variant explores w.p. $1/2$, this leads to low-variance, and crucial for the analysis

Our Approach: Focused Online Learning

- Initialize: $q = (1/n, \ldots, 1/n)$
- For $t = 1, 2, \ldots, T$
  - Sample i_t according to $p = 0.5\, q + 0.5\, (1/n, \ldots, 1/n)$
  - Feed i_t to the online learner
  - Update $q_{i_t} = q_{i_t} \exp(\phi_{i_t}(w_t)\,/(2np_{i_t}))$ and normalize

Our Approach: Focused Online Learning

- Initialize: $q = (1/n, \ldots, 1/n)$
- For $t = 1, 2, \ldots, T$
  - Sample i_t according to $p = 0.5\, q + 0.5\, (1/n, \ldots, 1/n)$
  - Feed i_t to the online learner
  - Update $q_{i_t} = q_{i_t} \exp(\phi_{i_t}(w_t)\,/(2np_{i_t}))$ and normalize

Observe: Using tree data-structure, each iteration costs $O(\log(n))$ plus the online learner time

Our Approach: Focused Online Learning

- Initialize: $q = (1/n, \ldots, 1/n)$
- For $t = 1, 2, \ldots, T$
  - Sample i_t according to $p = 0.5\, q + 0.5\, (1/n, \ldots, 1/n)$
  - Feed i_t to the online learner
  - Update $q_{i_t} = q_{i_t} \exp(\phi_{i_t}(w_t) / (2np_{i_t}))$ and normalize

Observe: Using tree data-structure, each iteration costs $O(\log(n))$ plus the online learner time

Theorem

If $T \geq \tilde{\Omega}(n + C)$, and $k = \Omega(\log(n))$, and if $t_1, \ldots, t_k$ are sampled at random from $[T]$, then with high probability

$$\forall i, \quad \phi_i \left(\text{Majority}(w_{t_1}, \ldots, w_{t_k})\right) = 0$$

Proof Sketch

- The vector $z_t = \frac{\phi_{i_t}(w_t)}{p_{i_t}} e_{i_t}$ is an unbiased estimate of the gradient $(\phi_1(w_t), \ldots, \phi_n(w_t))$

Proof Sketch

- The vector $z_t = \frac{\phi_{i_t}(w_t)}{p_{i_t}} e_{i_t}$ is an unbiased estimate of the gradient $(\phi_1(w_t), \ldots, \phi_n(w_t))$
- The update of q is Mirror Descent w.r.t. Entropic regularization with z_t

Proof Sketch

- The vector $z_t = \frac{\phi_{i_t}(w_t)}{p_{i_t}} e_{i_t}$ is an unbiased estimate of the gradient $(\phi_1(w_t), \ldots, \phi_n(w_t))$
- The update of q is Mirror Descent w.r.t. Entropic regularization with z_t
- A certain generalized definition of variance of z_t is bounded by $2n$ because of the strong exploration

Proof Sketch

- The vector $z_t = \frac{\phi_{i_t}(w_t)}{p_{i_t}} e_{i_t}$ is an unbiased estimate of the gradient $(\phi_1(w_t), \ldots, \phi_n(w_t))$
- The update of q is Mirror Descent w.r.t. Entropic regularization with z_t
- A certain generalized definition of variance of z_t is bounded by $2n$ because of the strong exploration
- A Bernstein's type inequality for Martingales enables to obtain strong concentration

Proof Sketch

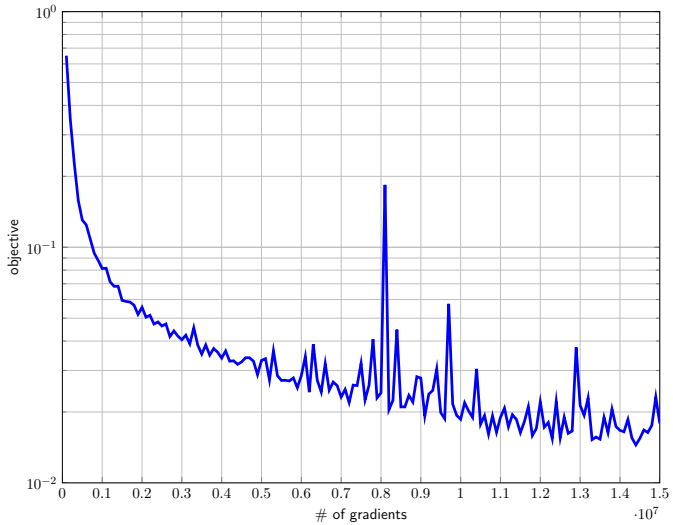
- The vector $z_t = \frac{\phi_{i_t}(w_t)}{p_{i_t}} e_{i_t}$ is an unbiased estimate of the gradient $(\phi_1(w_t), \ldots, \phi_n(w_t))$
- The update of q is Mirror Descent w.r.t. Entropic regularization with z_t
- A certain generalized definition of variance of z_t is bounded by $2n$ because of the strong exploration
- A Bernstein's type inequality for Martingales enables to obtain strong concentration
- Union bound over every i enables to conclude the proof

Related Work

- Auer et al 2002: The main idea is there, but EXP3.P.1 costs $\Omega(n)$ per iteration
- Hazan, Clarckson, Woodruff 2012, Hazan, Koren, Srebro 2011: Only for linear classifiers, rate of $(n+d)C$.
- AdaBoost (Freund & Schapire 1995): Only for binary classification, batch nature, similar rate.
  In practice: AdaBoost's predictor is ensemble but ours is a single classifier

Outline

Why SGD is slow at the end?



- Rare mistakes: Suppose all but 1% of the examples are correctly classified. SGD will now waste 99% of its time on examples that are already correct by the model
- High variance, even close to the optimum

Solving ERM for Strongly-convex and Smooth Problems

$$\min_{w \in \mathbb{R}^d} P(w) := \frac{1}{n} \sum_{i=1}^{n} \phi_i(w) + \frac{\lambda}{2} \|w\|^2$$

- Now assume that ϕ_i is convex and $O(1)$-smooth

Can we improve SGD ?

Theorem

Any algorithm for solving ERM that only accesses the objective using oracle that returns a gradient of a random example and has $\log(1/\epsilon)$ *rate must perform* $\tilde{\Omega}(n^2)$ *iterations*

Can we improve SGD ?

Theorem

Any algorithm for solving ERM that only accesses the objective using oracle that returns a gradient of a random example and has $\log(1/\epsilon)$ *rate must perform* $\tilde{\Omega}(n^2)$ *iterations*

Proof idea:

- Consider two objectives (in both, $\lambda = 1$): for $i \in \{\pm 1\}$

$$P_i(w) = \frac{1}{2n}\left(\frac{n-1}{2}(w-i)^2 + \frac{n+1}{2}(w+i)^2\right)$$

- A stochastic gradient oracle returns $w \pm i$ w.p. $\frac{1}{2} \pm \frac{1}{2n}$
- Easy to see that $w_i^* = -i/n$, $P_i(0) = 1/2$, $P_i(w_i^*) = 1/2 - 1/(2n^2)$
- Therefore, solving to accuracy $\epsilon < 1/(2n^2)$ amounts to determining the bias of the coin

Can we improve SGD ?

A stronger oracle:

- The negative result assumes we only see a gradient of a randomly chosen example
- A slightly stronger oracle: we also see the index of the chosen example

Can we improve SGD ?

A stronger oracle:

- The negative result assumes we only see a gradient of a randomly chosen example
- A slightly stronger oracle: we also see the index of the chosen example

With the stronger oracle, SDCA (and SAG, SVRG, ...) convergence rate is

$$(n + C)\log(1/\epsilon)$$

where $C = 1/\lambda$ (a reasonable measure of capacity).

SDCA = Stochastic Dual Coordinate Ascent

- Maintain "dual" vectors $\alpha_1, \ldots, \alpha_n$
- At iteration t, sample $i \sim [n]$ and update

$$\alpha_i^{(t)} = \alpha_i^{(t-1)} - \eta\lambda n \left(\nabla \phi_i(w^{(t-1)}) + \alpha_i^{(t-1)} \right)$$

$$w^{(t)} = w^{(t-1)} - \eta \left(\nabla \phi_i(w^{(t-1)}) + \alpha_i^{(t-1)} \right)$$

Intuition: Why SDCA is better than SGD

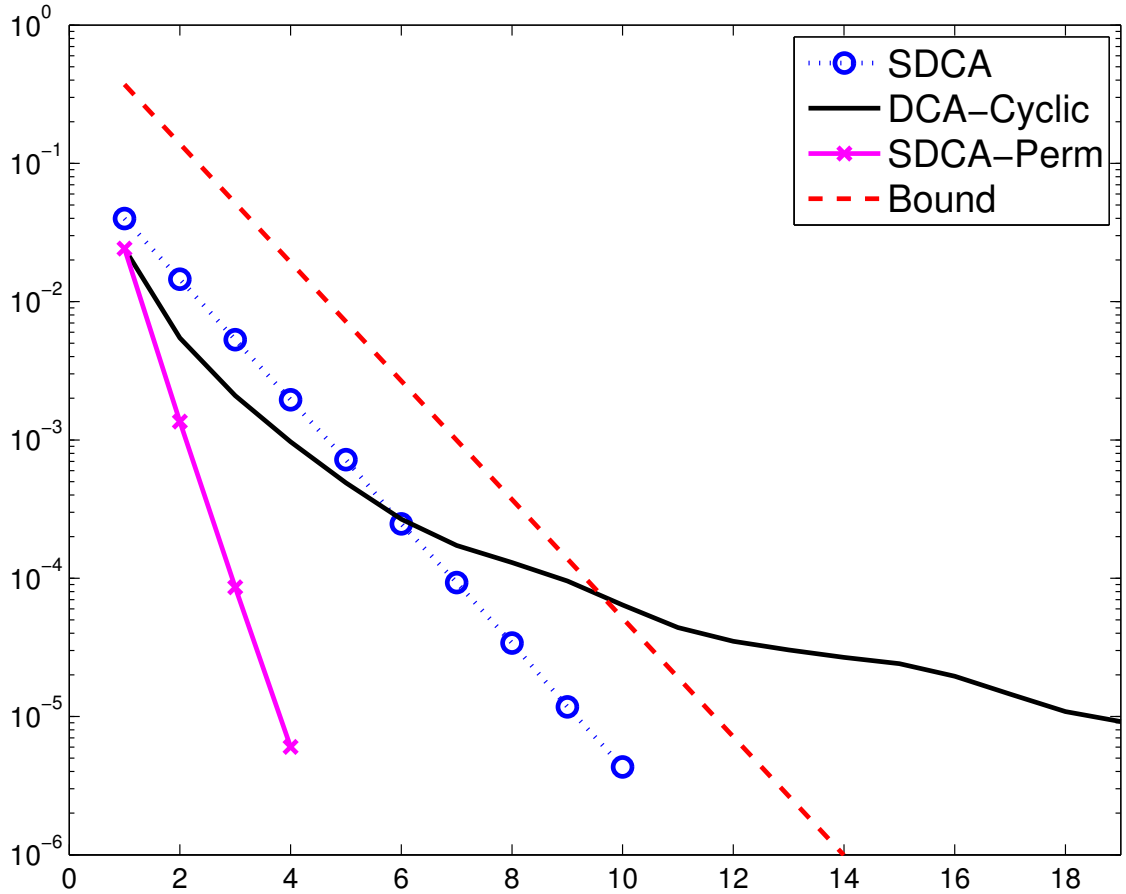
- The update step of both SGD and SDCA is $w^{(t)} = w^{(t-1)} - \eta v^{(t)}$ where

$$v^{(t)} = \begin{cases} \nabla \phi_i(w^{(t-1)}) + \lambda w^{(t-1)} & \text{for SGD} \\ \nabla \phi_i(w^{(t-1)}) + \alpha_i^{(t-1)} & \text{for SDCA} \end{cases}$$

- In both cases $\mathbb{E}[v^{(t)} | w^{(t-1)}] = \nabla P(w^{(t)})$
- What about the variance?
- For SGD, even if $w^{(t-1)} = w^*$, the variance of $v^{(t)}$ is still constant
- For SDCA, the variance of $v^{(t)}$ goes to zero as $w^{(t-1)} \to w^*$

SDCA vs. DCA — Randomization is crucial

- On CCAT dataset, $\lambda = 10^{-4}$, smoothed hinge-loss



- In particular, the bound of Luo and Tseng (1992) holds for cyclic order, hence must be inferior to our bound

Summary

- SGD is worst-case optimal, but in many cases can be inferior to ERM
- SGD converges quickly to an o.k. solution, but then slows down:
  1. Wastes time on already solved cases
  2. High variance even at w^*
- We provide methods with bounds of the from $(n + C)$

Future and Ongoing Work:

- Beyond convexity ...