Efficient learning with partial information on each individual example

Shai Shalev-Shwartz

School of Computer Science and Engineering
The Hebrew University of Jerusalem

Joint work with Nicolo Cesa-Bianchi, Ohad Shamir, Sham Kakade, Ambuj Tewari

Vision Seminar, Weizmann Inst., Jan. 2010

Motivation

Many examples but partial information on each individual example:

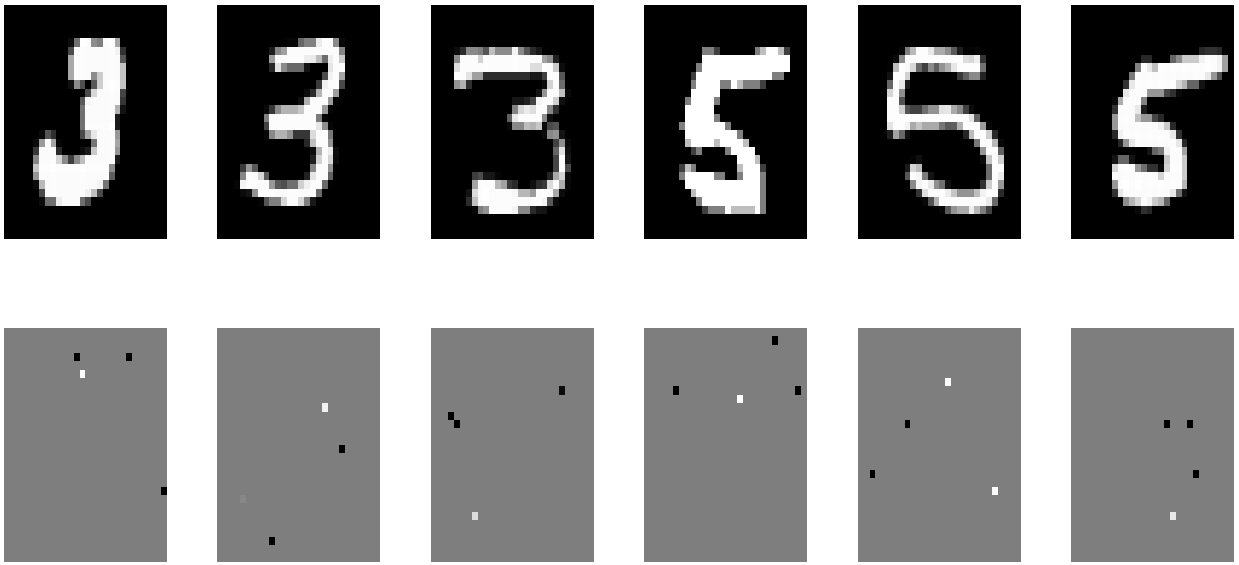
- Missing values (e.g. medical prognosis)
- Partial supervision (e.g. sponsored advertisement)
- Latent variables (e.g. Multi-instance learning)
- Noise
- privacy protected data

Message of this talk:

The availability of many examples can compensate for the lack of full information on each individual example

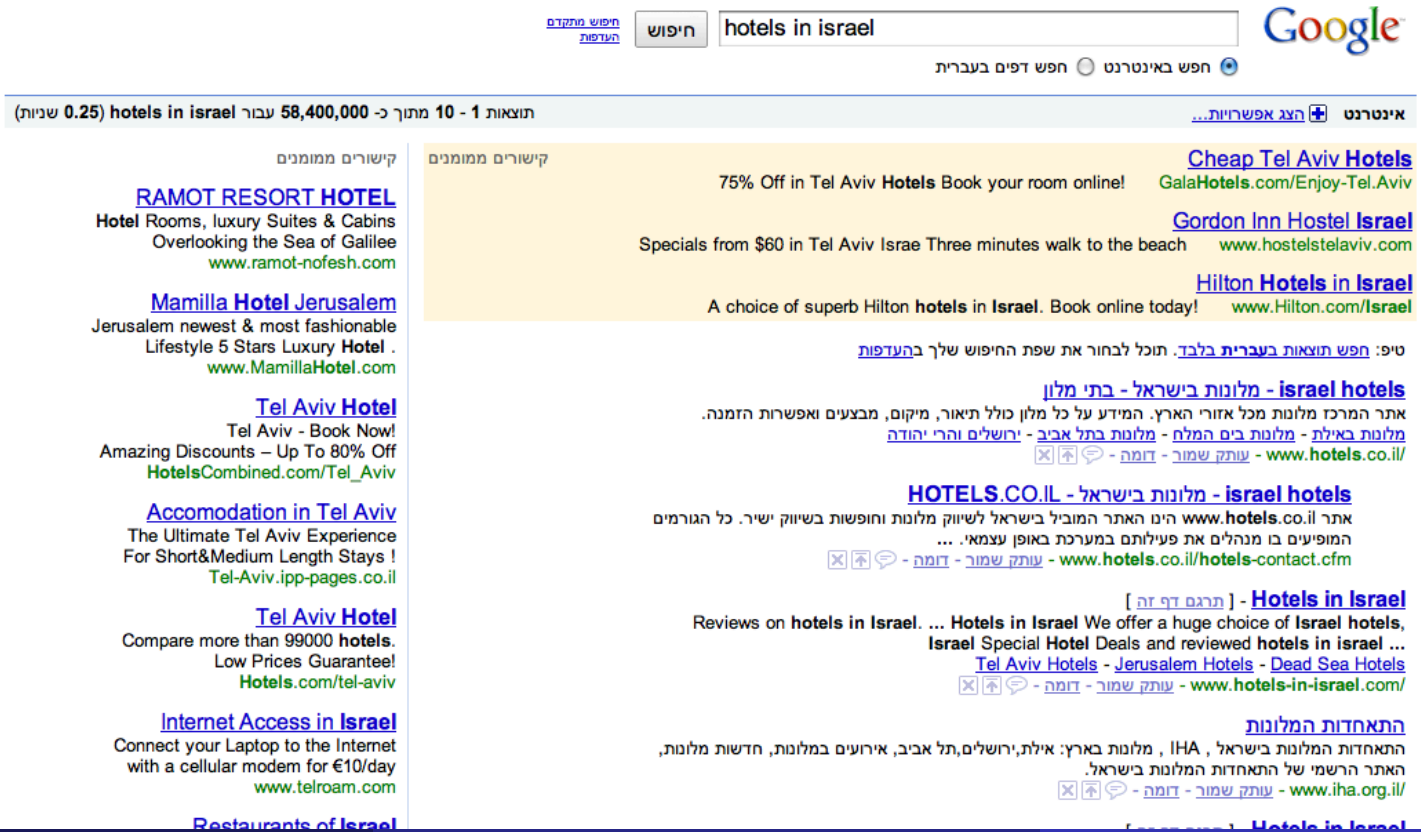
Attribute efficient regression

- Each training example is a pair $(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$
- Partial information: can only view $O(1)$ attributes of each individual example



Sponsored Advertisement (multi-armed bandit)

- Each training example is a pair $(\mathbf{x}, \mathbf{c}) \in \mathbb{R}^d \times [0,1]^k$
- Partial information: Don't know $\mathbf{c}$. Can only guess some y and know the value of c_y

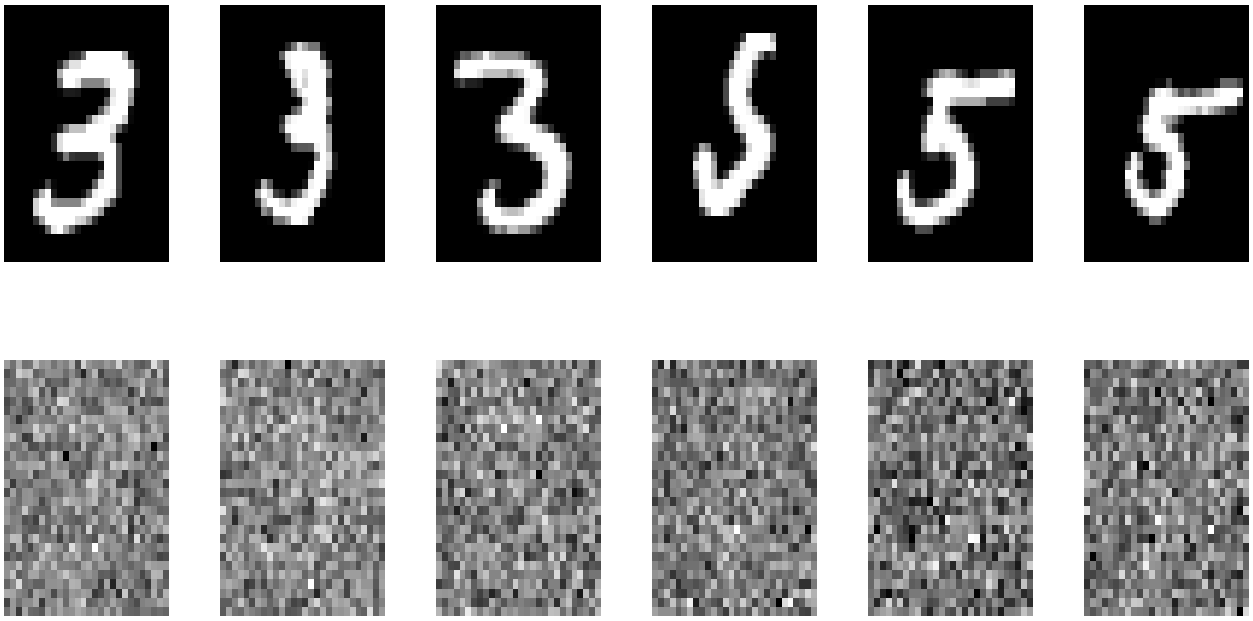


Object Recognition (Latent SVM)

- Each training example is a pair $(\mathbf{x}, y) \in \mathbb{R}^d \times \{\pm 1\}$
- Partial information: Latent variable h (represents the pose)

Privacy preserving learning (Sanitization by noise)

- Data custodian has a "clean" dataset $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$
- Partial information: To maintain privacy, only noisy versions of the examples are observed



How more data helps?

Three main techniques:

1. Missing information as noise
2. Active Exploration — try to "fish" the relevant information
3. Inject structure — problem hard in the original representation but becomes simple in another representation (different hypothesis class)

More data helps because:

1. It reduces variance — compensates for the noise
2. It allows more exploration
3. It compensates for statistical difficulty of learning larger hypotheses classes

Attribute efficient regression

Formal problem statement:

- Unknown distribution $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{R}$
- Goal: learn a linear predictor $\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle$ with low risk:
- Risk (generalization error): $L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{\mathcal{D}}[(\langle \mathbf{w}, \mathbf{x} \rangle - y)^2]$
- Training set: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
- Partial information: For each $(\mathbf{x}_i, y_i)$, learner can view only k attributes of $\mathbf{x}_i$
- Active selection: learner can choose which k attributes to see

Similar to "Learning with restricted focus of attention" (Ben-David & Dichterman 98)

Dealing with missing information

- Usually difficult — exponential ways to complete the missing information
- Popular approach — Expectation Maximization (EM)

Previous methods usually do not come with guarantees
(neither sample complexity nor computational complexity)

Ostrich approach

- Simply set the missing attributes to a default value
- Use your favorite regression algorithm for full information, e.g.,

$$\min_{\mathbf{w}} \sum_{i=1}^{m} (\langle \mathbf{w}, \tilde{\mathbf{x}}_i \rangle - y_i)^2 \; + \; \lambda \|\mathbf{w}\|_p^p$$

  - Ridge Regression: $p = 2$
  - Lasso: $p = 1$

Ostrich approach

- Simply set the missing attributes to a default value
- Use your favorite regression algorithm for full information, e.g.,

$$\min_{\mathbf{w}} \sum_{i=1}^{m} (\langle \mathbf{w}, \tilde{\mathbf{x}}_i \rangle - y_i)^2 \; + \; \lambda \|\mathbf{w}\|_p^p$$

  - Ridge Regression: $p = 2$
  - Lasso: $p = 1$

- Efficient
- Consistent? Sample complexity? How to choose attributes ?

Partial information as noise

- Observation:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} = \frac{1}{d} \begin{pmatrix} dx_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \ldots + \frac{1}{d} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ dx_d \end{pmatrix}$$

- Therefore, choosing i uniformly at random gives

$$\mathop{\mathbb{E}}_{i}[dx_i \mathbf{e}^i] = \mathbf{x} \ .$$

- If $\|\mathbf{x}\|_\infty \leq 1$ then $\|dx_i \mathbf{e}^i\|_\infty \leq d$ (i.e. variance increased)
- Reduced missing information to unbiased noise
- Many examples can compensate for the added noise

Many examples compensates for noise

- True goal: minimize over $\mathbf{w}$ the generalization error$L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)}[(\langle \mathbf{w}, \mathbf{x} \rangle - y)^2]$
- Loss on one example, $(\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$, gives an unbiased estimate for $L_{\mathcal{D}}(\mathbf{w})$
- Averaging loss on many examples reduces variance
- In our case, we construct an unbiased estimate of the loss of each single example
- Variance increases but many examples still reduces it back

Loss-Based Attribute Efficient Regression (LaBAER)

Theorem (Cesa-Bianchi, S, Shamir)

Let $\hat{\mathbf{w}}$ be the output of LaBAER. Then, with overwhelming probability

$$L_{\mathcal{D}}(\hat{\mathbf{w}}) \leq \min_{\mathbf{w}:\|\mathbf{w}\|_1 \leq B} L_D(\mathbf{w}) + \tilde{O}\left(\frac{d^2\, B^2}{\sqrt{m}}\right) \ ,$$

where d is dimension and m is number of examples.

Loss-Based Attribute Efficient Regression (LaBAER)

Theorem (Cesa-Bianchi, S, Shamir)

Let $\hat{\mathbf{w}}$ be the output of LaBAER. Then, with overwhelming probability

$$L_{\mathcal{D}}(\hat{\mathbf{w}}) \leq \min_{\mathbf{w}:\|\mathbf{w}\|_1 \leq B} L_D(\mathbf{w}) + \tilde{O}\left(\frac{d^2\, B^2}{\sqrt{m}}\right) \ ,$$

where d is dimension and m is number of examples.

- Factor of d^4 more examples compensates for the missing information !

Loss-Based Attribute Efficient Regression (LaBAER)

Theorem (Cesa-Bianchi, S, Shamir)

Let $\hat{\mathbf{w}}$ be the output of LaBAER. Then, with overwhelming probability

$$L_{\mathcal{D}}(\hat{\mathbf{w}}) \leq \min_{\mathbf{w}:\|\mathbf{w}\|_1 \leq B} L_D(\mathbf{w}) + \tilde{O}\left(\frac{d^2\, B^2}{\sqrt{m}}\right) \ ,$$

where d is dimension and m is number of examples.

- Factor of d^4 more examples compensates for the missing information !
- Can we do better?

Pegasos Attribute Efficient Regression (PAER)

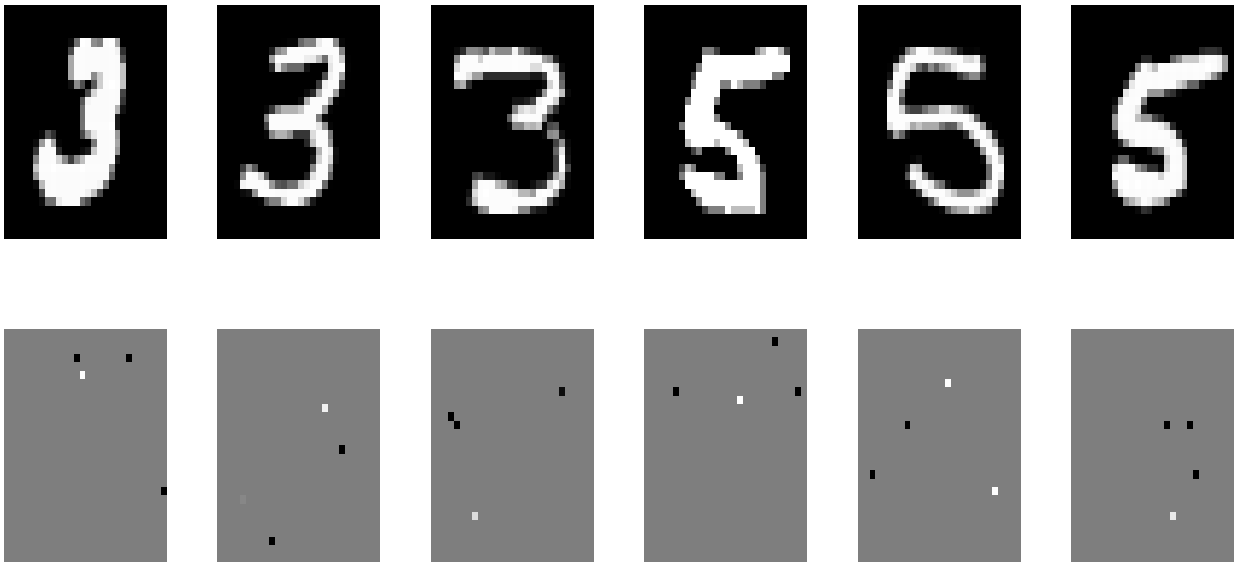
- The factor d^2 in the bound — because we estimate a matrix $\mathbf{x}\mathbf{x}^T$
- Can we avoid estimating the matrix ?
- Yes ! Estimate the gradient of the loss instead of the loss
- The gradient of the loss is a vector: $\nabla \ell(\mathbf{w}) = 2(\langle \mathbf{w}, \mathbf{x} \rangle - y)\mathbf{x}$
- Estimating the gradient:
  - Choose i uniformly from $[d]$ and estimate $\mathbf{x}$ as before
  - Choose j according to $\mathbb{P}[j] = |w_j| / \|\mathbf{w}\|_1$ and set $\hat{y} = \text{sgn}(w_j)\|\mathbf{w}\|_1 \, x_j$
  - Note that $\mathbb{E}[\hat{y}] = \langle \mathbf{w}, \mathbf{x} \rangle$
- Estimation depends on $\mathbf{w}$
- Need an online learning method
- Leads to a much better bound

Comparing Partial to Full information

- How many examples/attributes are needed to achieve error ϵ ?
- Below we show: $\frac{\text{PAER bound}}{\text{Lasso bound}}$

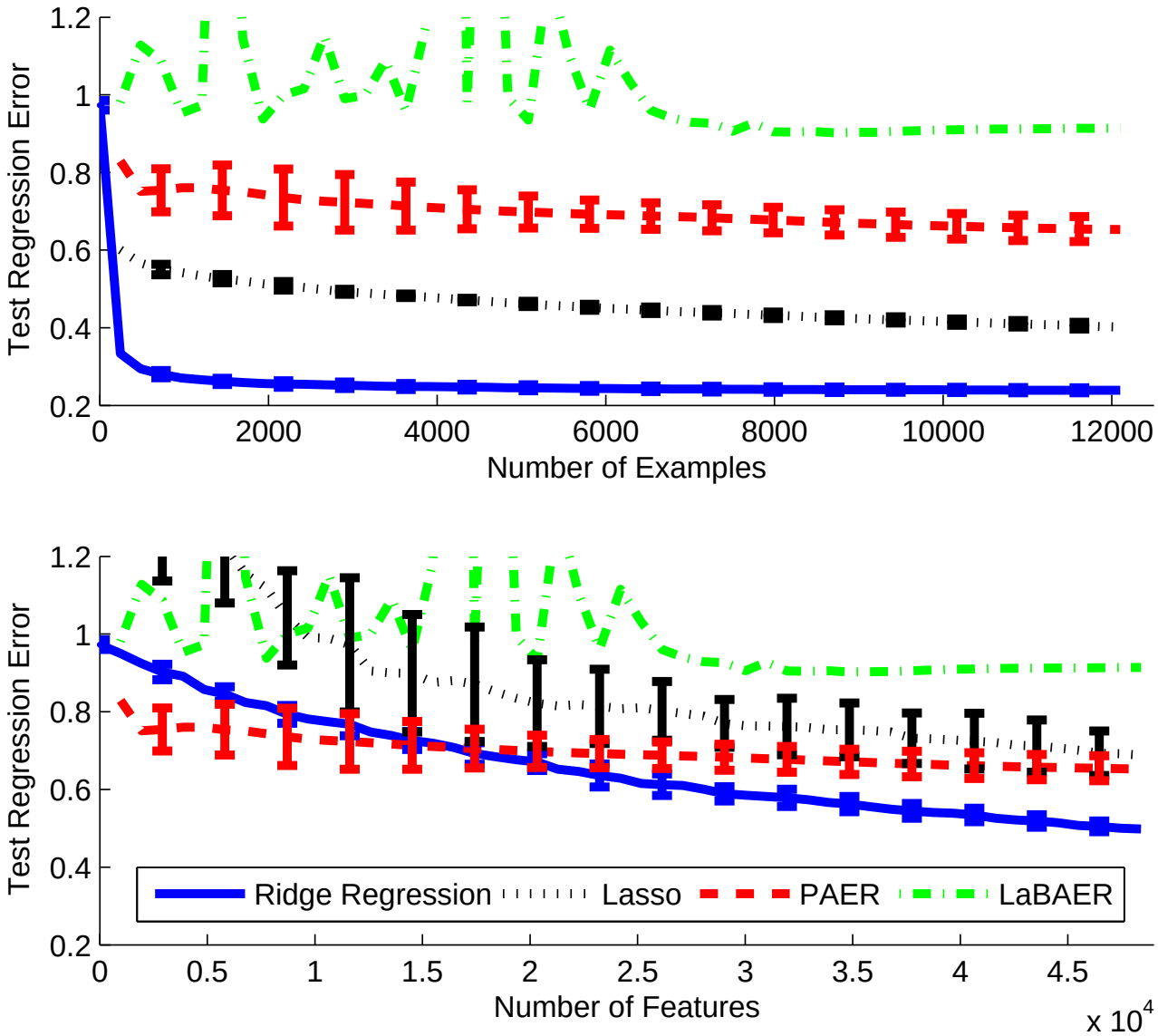
	sparse $\mathbf{w}^\star$	dense $\mathbf{w}^\star$
Sample complexity	d^2	d
Attributes complexity	d	1

Demonstration



- Averaging over all pairs, full information classifiers have generalization error of $\sim 1.5\%$
- Our algorithms have generalization error of $\sim 4\%$ while only observing 4 pixels of each example

Demonstration



Intermediate summary

- Efficient algorithms, Provably correct (finite sample generalization bound)
- Technique: Replace missing information with noise
- Having more examples compensates for the lack of information on each individual example

Intermediate summary

- Efficient algorithms, Provably correct (finite sample generalization bound)
- Technique: Replace missing information with noise
- Having more examples compensates for the lack of information on each individual example

Coming next:

- Partial supervision
- Technique: Active Exploration
- Larger regret can be compensated by having more examples

Second example: Online Ads Placement

For $t = 1, 2, \ldots, m$

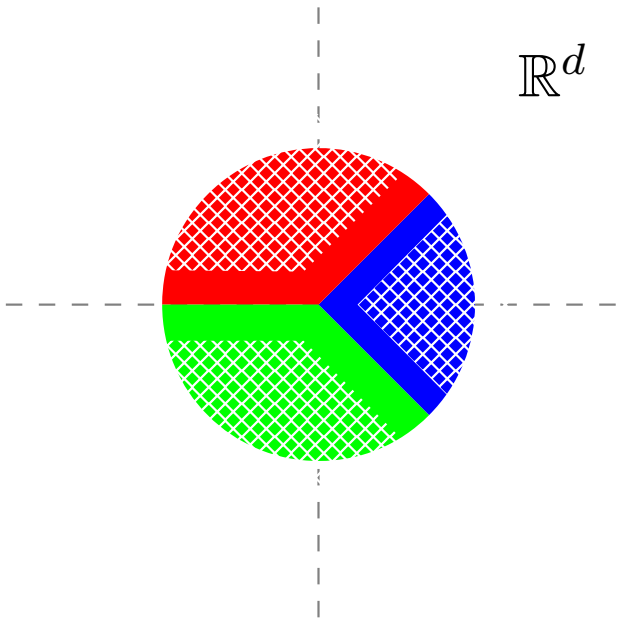
- Learner receives side information $\mathbf{x}_t \in \mathbb{R}^d$
- Learner places an ad $\hat{y}_t \in [k]$
- Learner pay cost $c_t(\hat{y}_t)$
- "Bandit setting" — learner does not know costs of other ads

Goal: Minimize error rate:

$$\text{err} = \frac{1}{m} \sum_{t=1}^{m} c_t(\hat{y}_t) \ .$$

Simplifying assumptions

- Cost vector is $\mathbf{e}^{y_t}$
- Exists W s.t. $\forall r \neq y_t,\ (W\mathbf{x}_t)_{y_t} - (W\mathbf{x}_t)_r \geq \mu$



Halving

- It is possible to adapt the Halving algorithm for our task
- Sample complexity is order of $\frac{k^2/\mu^2}{\epsilon}$
- But runtime grows like $(1/\mu)^{kd'} = (m+k)^{\tilde{O}(k/\mu^2)}$

How can we improve runtime?

- Halving is not efficient because it does not utilize the structure of $\mathcal{H}$
- If we knew y_t we could have used the Perceptron which utilizes convexity and is thus efficient
- Next approach: Lets try to rely on the Perceptron
- But, how can we use Perceptron without knowing the value of y_t?

The Banditron (Kakade, S, Tewari 08)

- Let $\hat{y}_t$ be the predicting of our current model
- Explore: From time to time, instead of predicting $\hat{y}_t$ guess some $\tilde{y}_t$
- Suppose we get the feedback 'correct', i.e. $\tilde{y}_t = y_t$
- Then, we have full information for Perceptron's update: $(\mathbf{x}_t, \hat{y}_t, \tilde{y}_t = y_t)$

The Banditron (Kakade, S, Tewari 08)

- Let $\hat{y}_t$ be the predicting of our current model
- Explore: From time to time, instead of predicting $\hat{y}_t$ guess some $\tilde{y}_t$
- Suppose we get the feedback 'correct', i.e. $\tilde{y}_t = y_t$
- Then, we have full information for Perceptron's update: $(\mathbf{x}_t, \hat{y}_t, \tilde{y}_t = y_t)$
- Exploration-Exploitation Tradeoff:
  - When exploring we may have $\tilde{y}_t = y_t \neq \hat{y}_t$ and can learn from this
  - When exploring we may have $\tilde{y}_t \neq y_t = \hat{y}_t$ and then we had the right answer in our hands but didn't exploit it
- Exploration increases cost but more data compensates for it!

The Banditron

Theorem

- *Banditron's sample complexity is order of* $\frac{k/\mu^2}{\epsilon^2}$
- *Banditron's runtime is* $O(k/\mu^2)$

The Banditron

Theorem

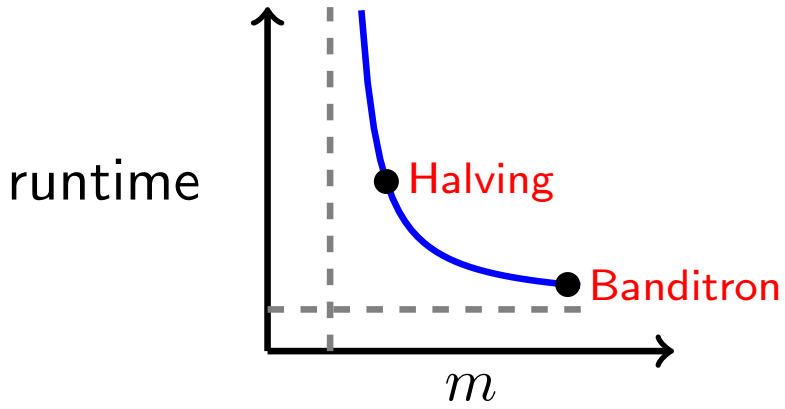
- *Banditron's sample complexity is order of* $\frac{k/\mu^2}{\epsilon^2}$
- *Banditron's runtime is* $O(k/\mu^2)$

The crux of difference between Halving and Banditron:

- Without having the full information, the version space is non-convex and therefore it is hard to utilize the structure of $\mathcal{H}$
- Because we relied on the Perceptron we did utilize the structure of $\mathcal{H}$ and got an efficient algorithm
- We managed to obtain 'full-information examples' by using exploration
- The price of exploration is a higher regret

More data improves runtime

Algorithm	samples	runtime
Halving	$\frac{k^2/\mu^2}{\epsilon}$	$(m+k)^{\tilde{O}(k/\mu^2)}$
Banditron	$\frac{k/\mu^2}{\epsilon^2}$	k/μ^2



Last example: Latent SVMs

- Latent SVM: The goal is to learn a classifier of the form

$$f_{\mathbf{w}}(\mathbf{x}) = \text{sgn}\left(\max_{z \in Z(\mathbf{x})} \langle \mathbf{w}, \Phi(\mathbf{x}, z) \rangle\right) \ ,$$

  where $\mathbf{w}$ is the parameter vector and $Z(\mathbf{x})$ is a set of possible latent values.
- Felzenszwalb, McAllester and Ramanan used latent SVM for training a deformable part model for object detection: $\Phi(\mathbf{x}_i, z)$ specifies a score of a HOG pyramid which is places according to z.

Latent SVM

- In the separable case, the learning problem is to find $\mathbf{w}$ with minimal norm s.t.

$$\forall (\mathbf{x}, y) \in S, \quad y \max_{z \in Z(\mathbf{x})} \langle \mathbf{w}, \Phi(\mathbf{x}, z) \rangle \geq 1 \ .$$

Latent SVM

- In the separable case, the learning problem is to find $\mathbf{w}$ with minimal norm s.t.

$$\forall(\mathbf{x}, y) \in S, \quad y \max_{z \in Z(\mathbf{x})} \langle \mathbf{w}, \Phi(\mathbf{x}, z) \rangle \geq 1 \ .$$

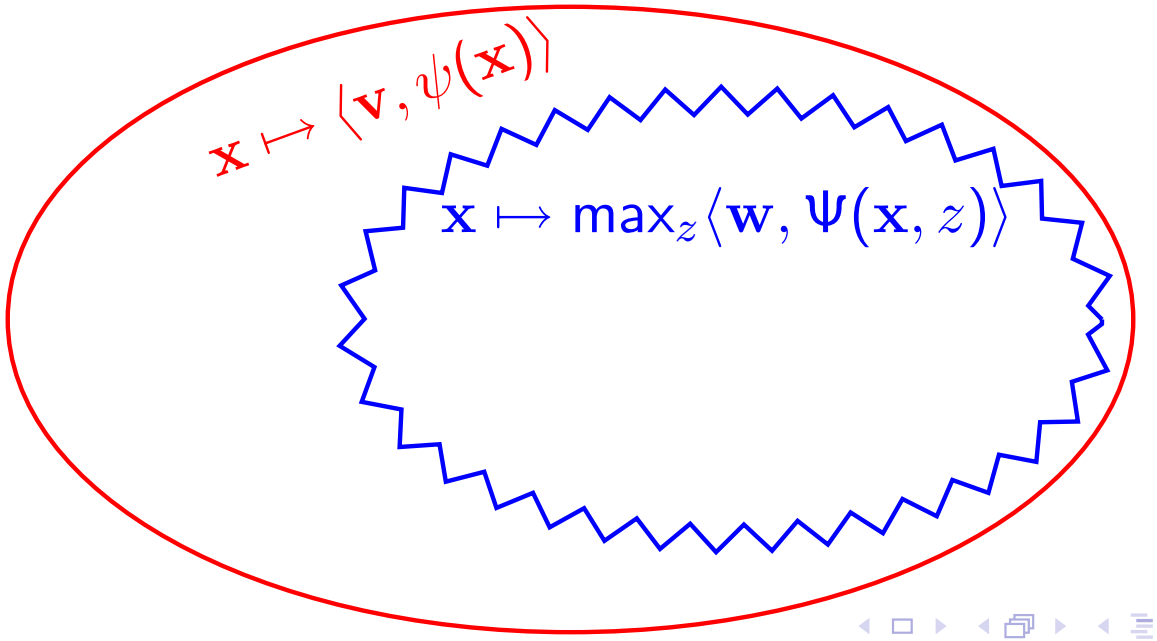
- Problem: Non-convex constraints

Latent SVM

- In the separable case, the learning problem is to find $\mathbf{w}$ with minimal norm s.t.

$$\forall(\mathbf{x}, y) \in S, \quad y \max_{z \in Z(\mathbf{x})} \langle \mathbf{w}, \Phi(\mathbf{x}, z) \rangle \geq 1 \ .$$

- Problem: Non-convex constraints
- Main idea: Work with a larger hypothesis class for which the constraints become convex



Latent SVM

Step 1:

- Note that for any $\beta > 0$ and vector $(a_1, \ldots, a_r)$:

$$\max_z a_z \leq \frac{1}{\beta} \log \left(\sum_z e^{\beta a_z} \right) \leq \max_z a_z + \frac{\log(r)}{\beta}$$

- Therefore, for β large enough we have

$$\max_z \langle \mathbf{w}, \Phi(\mathbf{x}, z) \rangle \approx \frac{1}{\beta} \log \left(\sum_z e^{\beta \langle \mathbf{w}, \Phi(\mathbf{x}, z) \rangle} \right) .$$

- Original constraint: $y \max_z \langle \mathbf{w}, \Phi(\mathbf{x}, z) \rangle \geq 1$
- New constraint: $y \frac{1}{\beta} \log \left(\sum_z e^{\beta \langle \mathbf{w}, \Phi(\mathbf{x}, z) \rangle} \right) \geq 1$

Latent SVM

Step 2:

- Equivalent constraint: $\sum_z e^{\beta\langle \mathbf{w}, \Phi(\mathbf{x}, z)\rangle} \geq e^{\beta - y}$
- Based on Taylor expansion, can write $e^{\beta\langle \mathbf{w}, \Phi(\mathbf{x}, z)\rangle} = \langle \mathbf{v}, \Psi(\mathbf{x}, z)\rangle$, where Ψ is a mapping to some Reproducing Kernel Hilbert Space.
- Therefore, constraints become convex, and problem is solvable using the kernel-trick
- The price: we learn a larger hypothesis class, hence need more data

Summary

- Learning theory: Many examples $\Rightarrow$ smaller estimation error
- This work: Many examples $\Rightarrow$ more efficient algorithms for partial information case
- How can more data reduce runtime:
  1. Missing information as noise
  2. Active Exploration
  3. Inject structure