Efficient Bandit Algorithms for Online Multiclass Prediction

Sham M. Kakade
Shai Shalev-Shwartz
Ambuj Tewari

TOYOTA TECHNOLOGICAL INSTITUTE
CHICAGO

ICML, July 2008

Motivation

- Online web advertisement systems
  - User submits a query
  - System (the learner) places an ad
  - User either “clicks” or ignores
  - Goal: Maximize number of “clicks”
- Modeling ?
  - Not the common online learning setting --
    If user ignores, we don’t get the “correct” ad
  - Not the common multi-armed bandit --
    We are also provided with a query

Outline

- Online Bandit Multi-class Categorization
- Background: The Multi-class Perceptron
- The Banditron
- Analysis
- Experiments
- The Separable Case
- Extensions and Open Problems

Online Bandit Multiclass Categorization

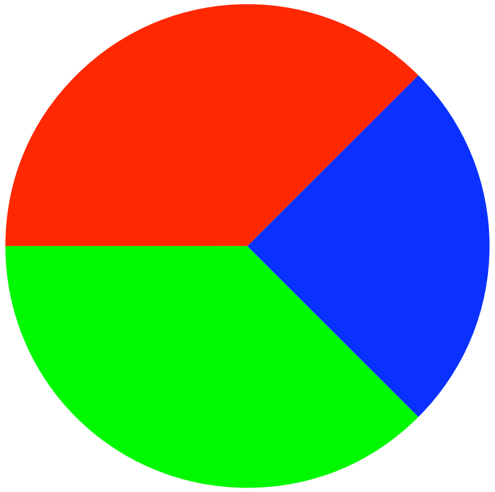
For $t = 1, 2, \ldots, T$

- Receive $\mathbf{x} \in \mathbb{R}^d$ (query)
- Predict $\hat{y}_t \in \{1, \ldots, k\}$ (ad)
- Pay $\mathbf{1}[y_t \neq \hat{y}_t]$ (click feedback)
- y_t is not revealed

Linear Hypotheses

- A hypothesis is a mapping $h : \mathbb{R}^d \rightarrow \{1, \ldots, k\}$
- Linear hypothesis: Exists $k \times d$ matrix W s.t.

$$h(\mathbf{x}) \;=\; \underset{r}{\operatorname{argmax}}\, (W\,\mathbf{x})_r$$



The Multiclass Perceptron

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

- Receive $\mathbf{x}_t \in \mathbb{R}^d$
- Predict $\hat{y}_t \ = \ \underset{r}{\operatorname{argmax}} \, (W^t \, \mathbf{x}_t)_r$
- Receive y_t
- Update: $W^{t+1} = W^t + U^t$ where $U^t \ =$

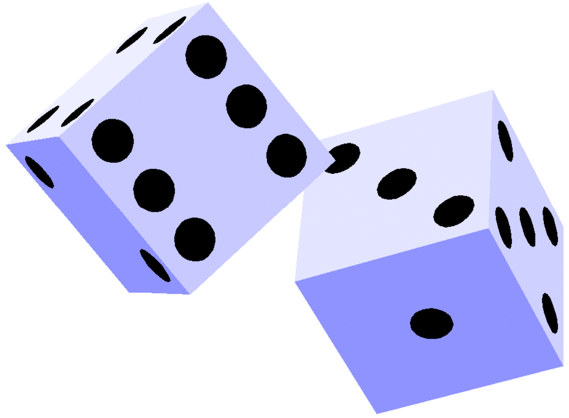
$$
U^t = \left[\begin{array}{ccc}
0 & \ldots & 0 \\
 & \vdots & \\
0 & \ldots & 0 \\
\ldots & \mathbf{x}_t & \ldots \\
0 & \ldots & 0 \\
 & \vdots & \\
0 & \ldots & 0 \\
\ldots & -\mathbf{x}_t & \ldots \\
0 & \ldots & 0 \\
 & \vdots & \\
0 & \ldots & 0
\end{array}\right]
\begin{array}{l}
 \\
 \\
 \\
\leftarrow \text{row } y_t \\
 \\
 \\
 \\
\leftarrow \text{row } \hat{y}_t \\
 \\
 \\

\end{array}
$$

Perceptron in the Bandit Setting

$$U^t = \left[\begin{array}{ccc} 0 & \ldots & 0 \\ & \vdots & \\ 0 & \ldots & 0 \\ \ldots & \mathbf{x}_t & \ldots \\ 0 & \ldots & 0 \\ & \vdots & \\ 0 & \ldots & 0 \\ \ldots & -\mathbf{x}_t & \ldots \\ 0 & \ldots & 0 \\ & \vdots & \\ 0 & \ldots & 0 \end{array}\right] \begin{array}{l} \\ \\ \\ \leftarrow \text{row } y_t \\ \\ \\ \\ \leftarrow \text{row } \hat{y}_t \\ \\ \\ \\ \end{array}$$

- Problem: We're blind to value of y_t
- Solution: Randomization can help !

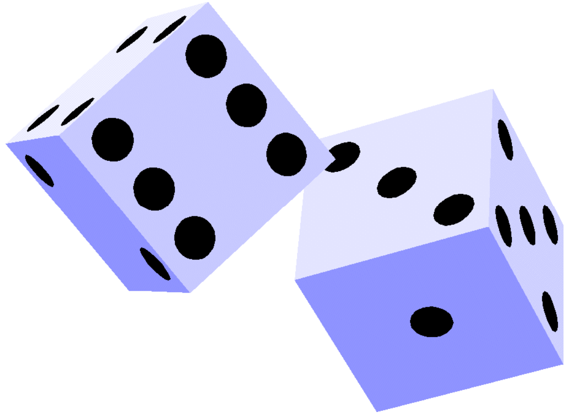


Exploration

- Explore: 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 know that
  - $\hat{y}_t \neq y_t$
  - $y_t = \tilde{y}_t$
- So, we can update W using the matrix U^t

Exploration vs. Exploitation

- But, if our current model is correct, i.e. $\hat{y}_t = y_t$
- And, we guess some other $\tilde{y}_t$
- Then, we both suffer loss and do not know how to update W
- In this case, it's better to Exploit the quality of current model
- We control the exploration-exploitation tradeoff using randomization



The Banditron

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

- Receive $\mathbf{x}_t \in \mathbb{R}^d$
- Set $\hat{y}_t \;=\; \underset{r}{\operatorname{argmax}}\,(W^t\,\mathbf{x}_t)_r$
- Define: $P(r) = (1-\gamma)\mathbf{1}[r = \hat{y}_t] + \frac{\gamma}{k}$
- Randomly sample $\tilde{y}_t$ according to P
- Predict $\tilde{y}_t$ and receive feedback $\mathbf{1}[\tilde{y}_t = y_t]$
- Update: $W^{t+1} = W^t + \tilde{U}^t$

The Banditron

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

- Receive $\mathbf{x}_t \in \mathbb{R}^d$
- Set $\hat{y}_t = \underset{r}{\operatorname{argmax}} \, (W^t \, \mathbf{x}_t)_r$
- Define: $P(r) = (1-\gamma)\mathbf{1}[r = \hat{y}_t] +$
- Randomly sample $\tilde{y}_t$ accordin P
- Predict $\tilde{y}_t$ and receive feedb ck $\mathbf{1}[\tilde{y}_t = y_t]$
- Update: $W^{t+1} = W^t + \tilde{U}^t$

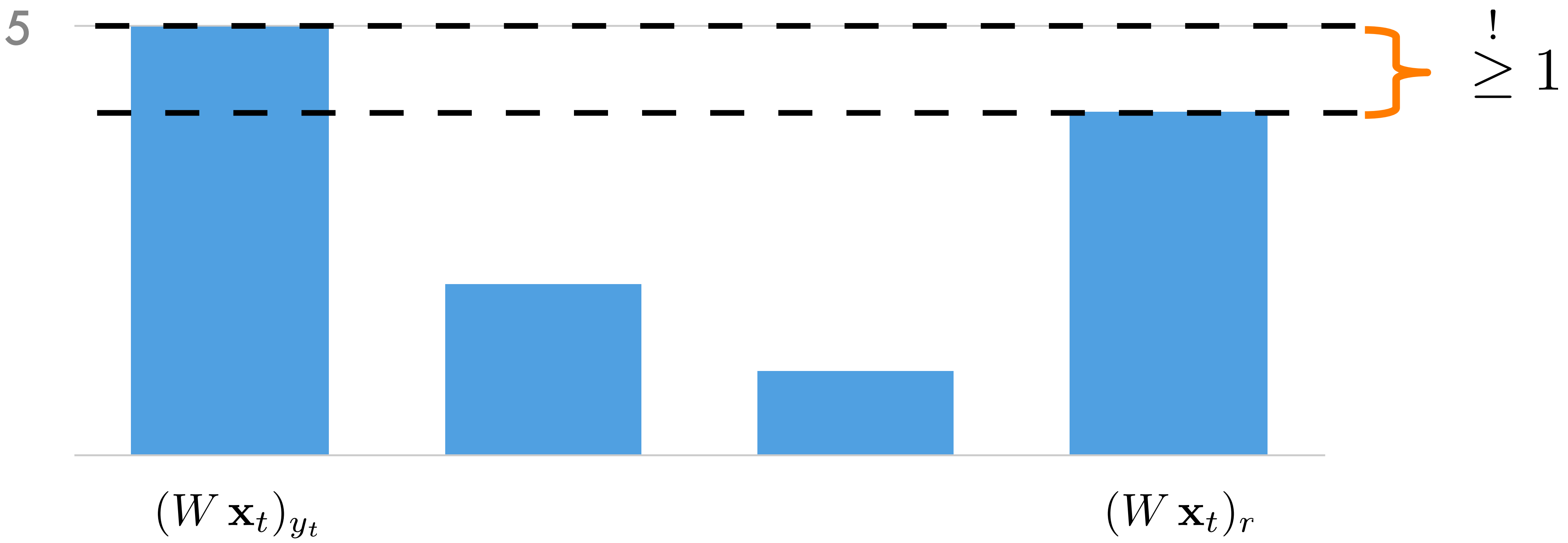
$$\begin{array}{cl}
\left[\begin{array}{ccc}
0 & \ldots & 0 \\
 & \vdots & \\
0 & \ldots & 0 \\
\ldots & \frac{\mathbf{1}[y_t = \tilde{y}_t]}{P(y_t)} \, \mathbf{x}_t & \ldots \\
0 & \ldots & 0 \\
 & \vdots & \\
0 & \ldots & 0 \\
\ldots & -\mathbf{x}_t & \ldots \\
0 & \ldots & 0 \\
 & \vdots & \\
0 & \ldots & 0
\end{array}\right]
&
\begin{array}{l}
\\ \\ \\ \leftarrow \text{row } y_t \\ \\ \\ \\ \leftarrow \text{row } \hat{y}_t \\ \\ \\ \\
\end{array}
\end{array}$$

The Banditron Expected Update

$$\mathbb{E}[\tilde{U}^t] = \sum_r P(r) \begin{bmatrix} 0 & \ldots & 0 \\ & \vdots & \\ 0 & \ldots & 0 \\ \ldots & \frac{\mathbf{1}[y_t = r]}{P(y_t)} \mathbf{x}_t & \ldots \\ 0 & \ldots & 0 \\ & \vdots & \\ 0 & \ldots & 0 \\ \ldots & -\mathbf{x}_t & \ldots \\ 0 & \ldots & 0 \\ & \vdots & \\ 0 & \ldots & 0 \end{bmatrix} = U^t$$

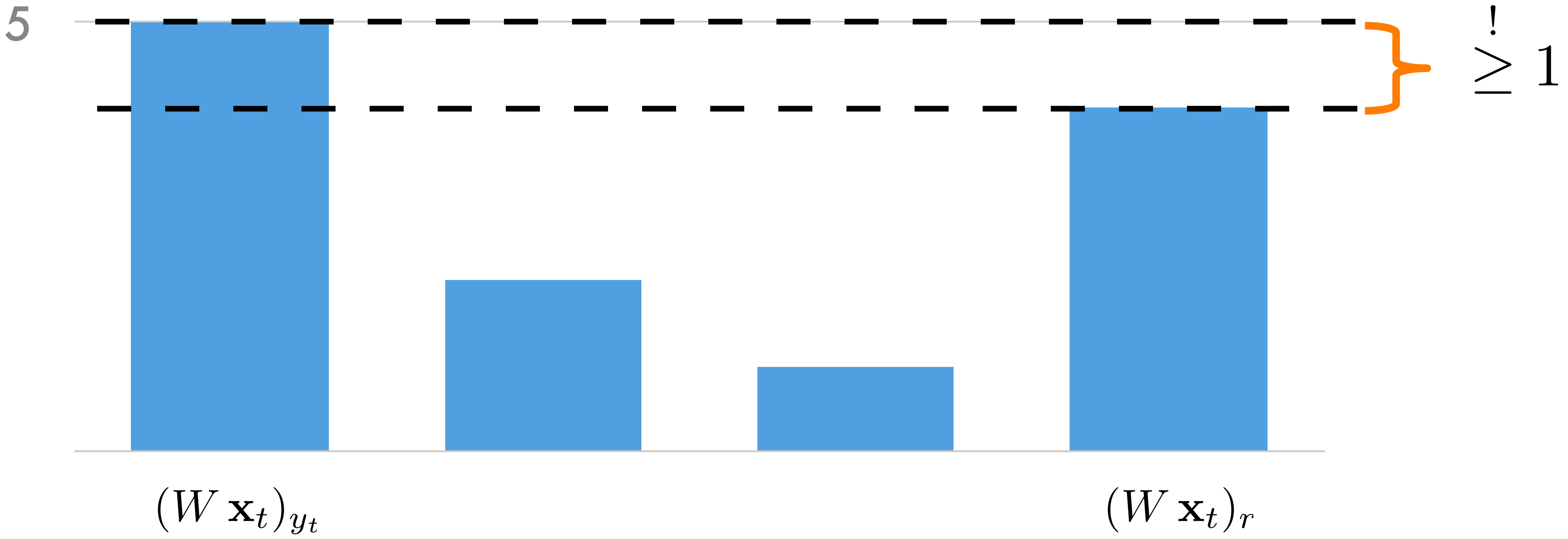
Analysis: The Hinge-Loss

$$\ell_t(W) \;=\; \max_{r \neq y_t} 1 - (W\,\mathbf{x}_t)_{y_t} + (W\,\mathbf{x}_t)_r \qquad \geq \;\; \mathbf{1}[y_t \neq \hat{y}_t]$$

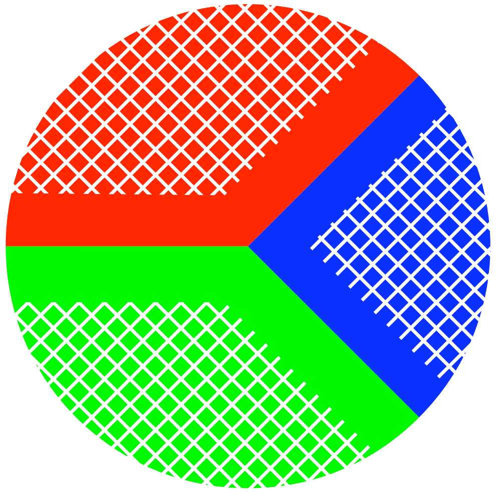


Analysis: The Hinge-Loss

$$\ell_t(W) \;=\; \max_{r\neq y_t} 1 - (W\,\mathbf{x}_t)_{y_t} + (W\,\mathbf{x}_t)_r \qquad \geq \; \mathbf{1}[y_t \neq \hat{y}_t]$$



The Separable Case:



Mistake Bounds

Perceptron:

$$M \quad \leq \quad L + D + \sqrt{L\,D}$$

Banditron:

$$\mathbb{E}[M] \quad \leq \quad L + \gamma\,T + 3\max\left\{\frac{k\,D}{\gamma}\,,\ \sqrt{D\,\gamma\,T}\right\} + \sqrt{\frac{k\,D\,L}{\gamma}}$$

Symbol	Meaning
M	# mistakes
L	competitor loss $\sum_t \ell_t(W^\star)$
D	competitor margin $\|W^\star\|_F^2$
k	# classes
T	# rounds
γ	Exploration-Exploitation parameter

Mistake Bounds (cont.)

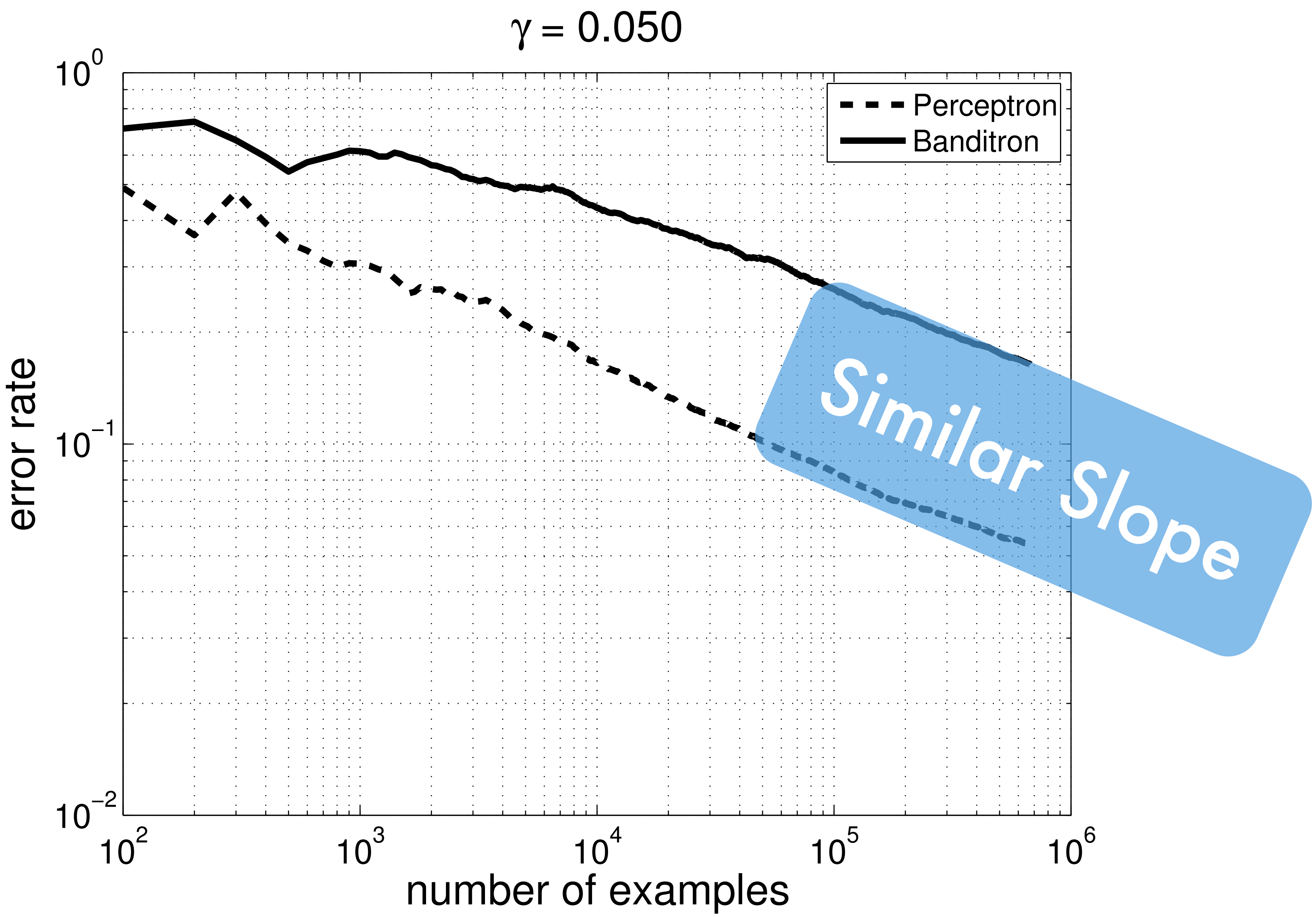
	Perceptron	Banditron
No noise: $L = 0$	D	$\sqrt{k\,D\,T}$
Low noise: $L = O(\sqrt{k\,D\,T})$	$\sqrt{k\,D\,T}$	$\sqrt{k\,D\,T}$
Noisy:	$L + T^{1/2}$	$L + T^{2/3}$

Symbol	Meaning
L	competitor loss $\sum_t \ell_t(W^\star)$
D	competitor margin $\|W^\star\|_F^2$
k	# classes
T	# rounds

Experiments

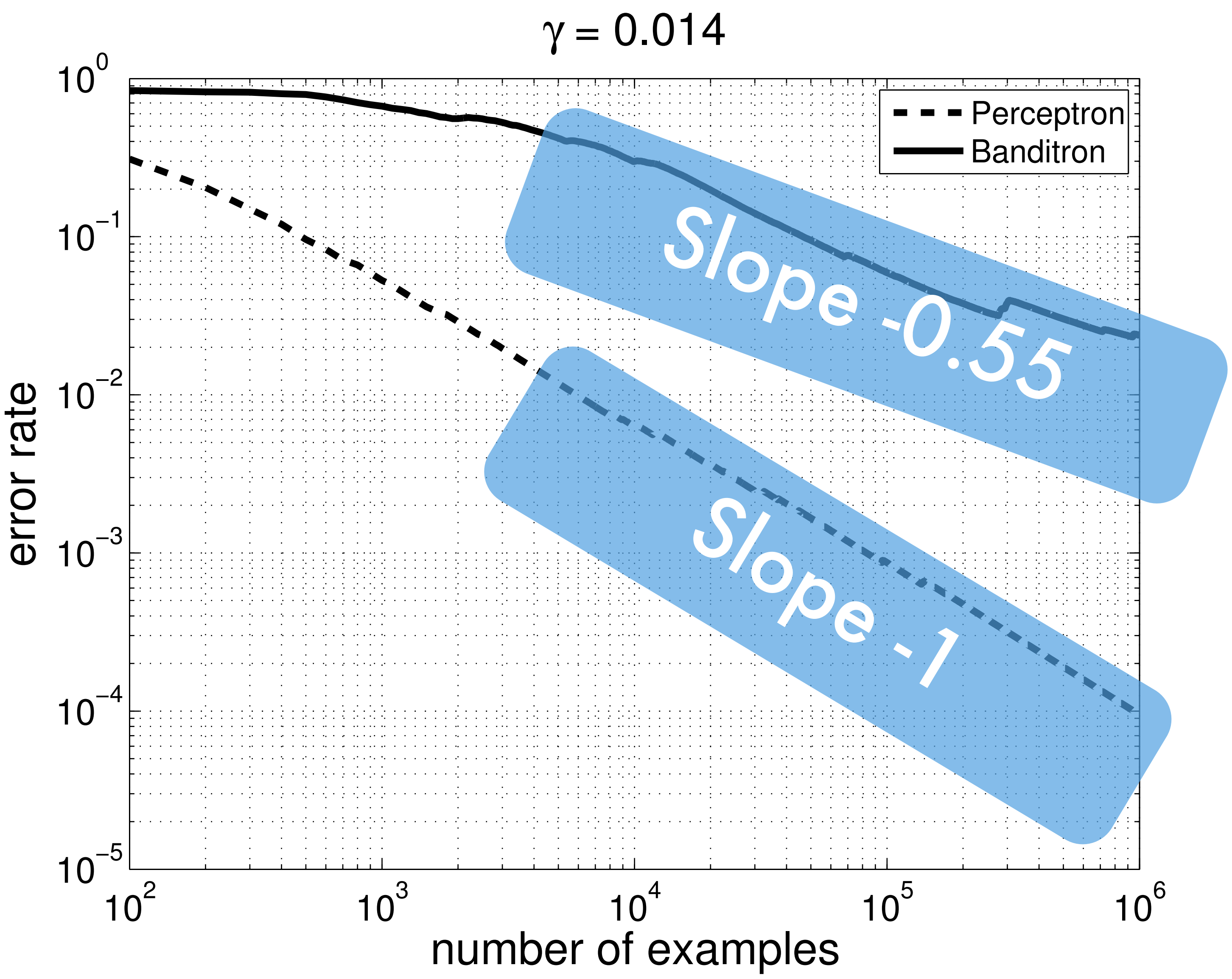
- Reuters RCV1
  - ~700k documents
  - Bag-of-words (d ~ 350k)
  - 4 labels {CCAT, ECAT, GCAT, MCAT}
- Synthetic separable data set
  - 9 classes, d=400, million instances
  - A simple simulation of generating text documents
- Synthetic non-separable data set
  - separable + 5% label noise

Experimental Results – Reuters

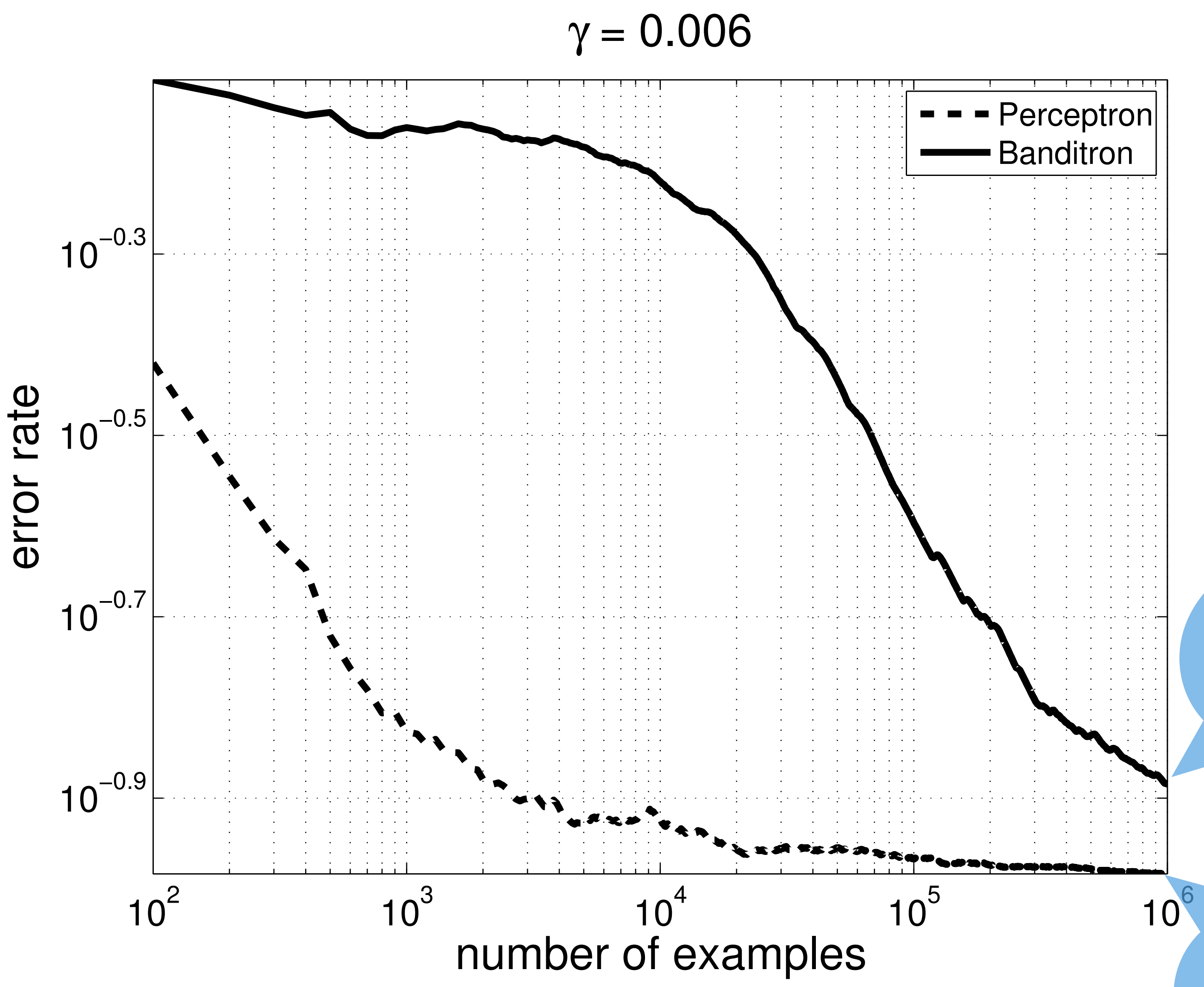


Similar Slope

Experimental Results – Separable Data



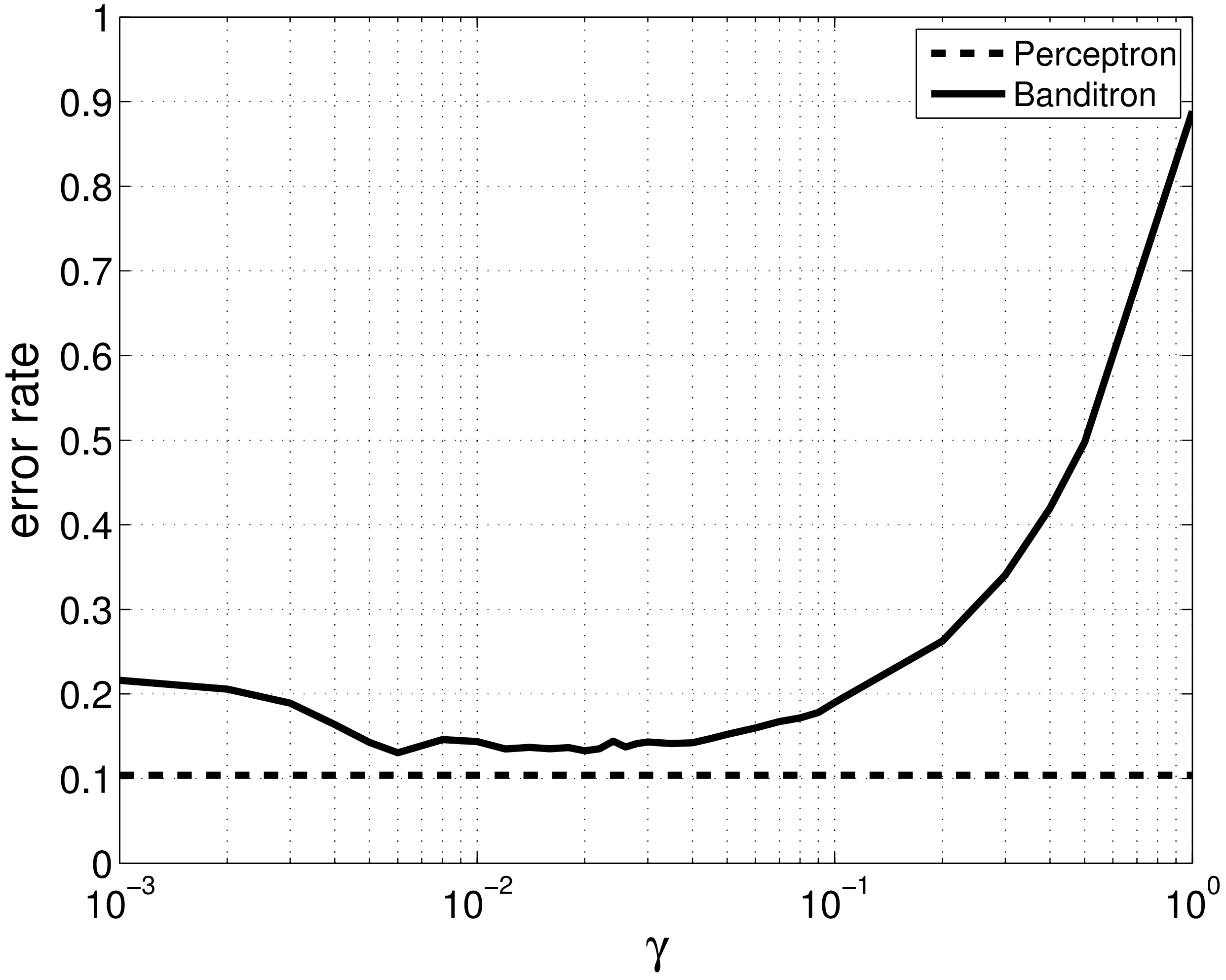
Experimental Results – 5% label noise

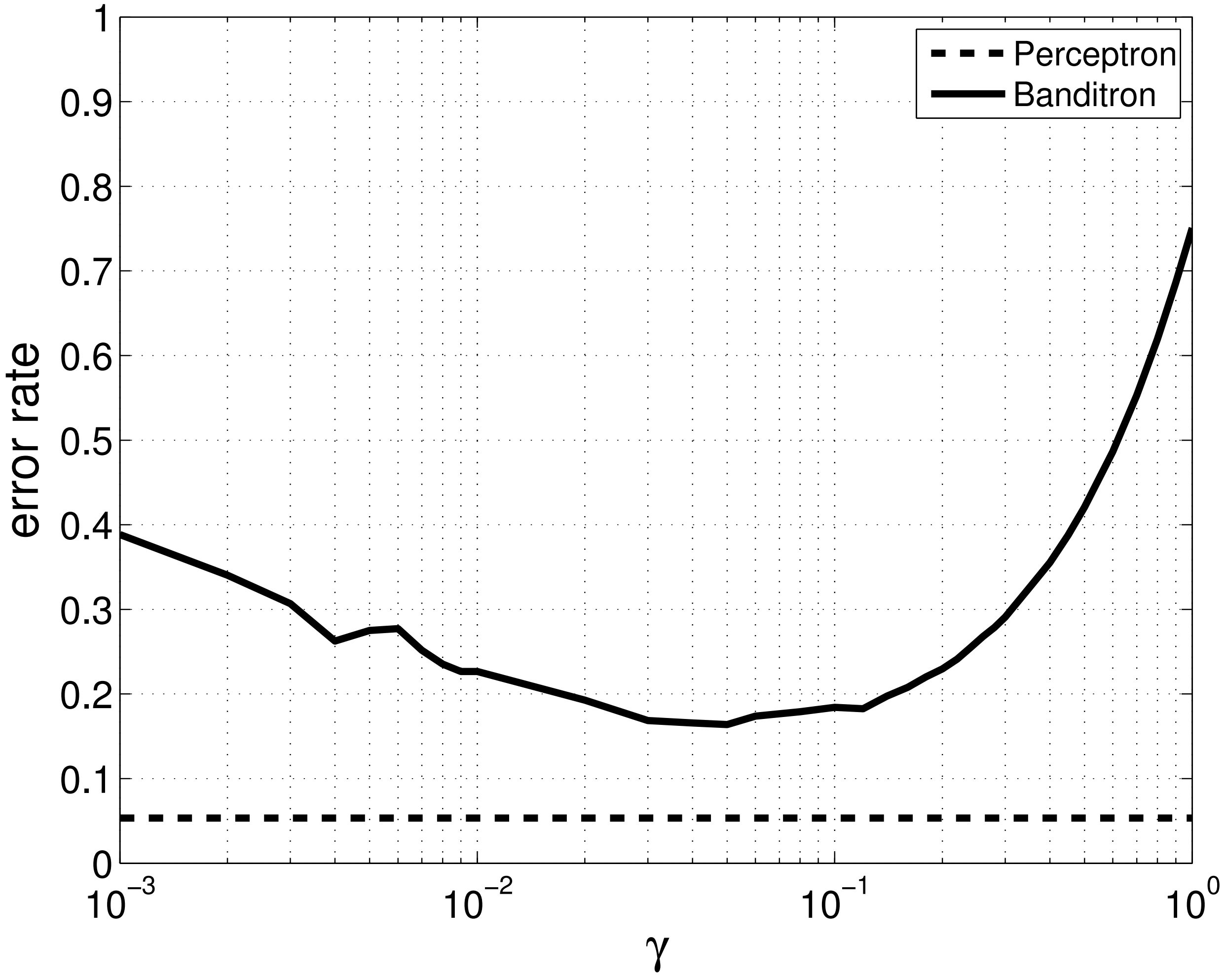


13%

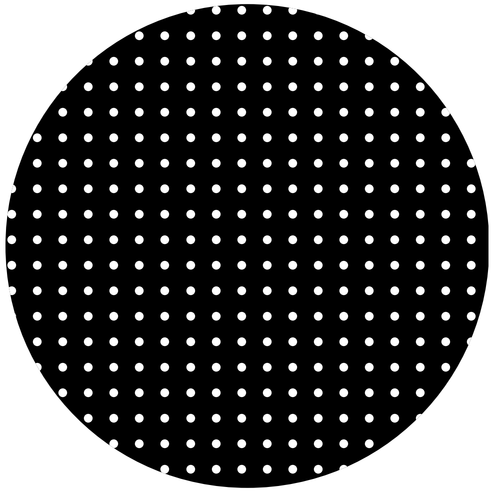
10%

Exploration-Exploitation Parameter


5% label noise


Reuters

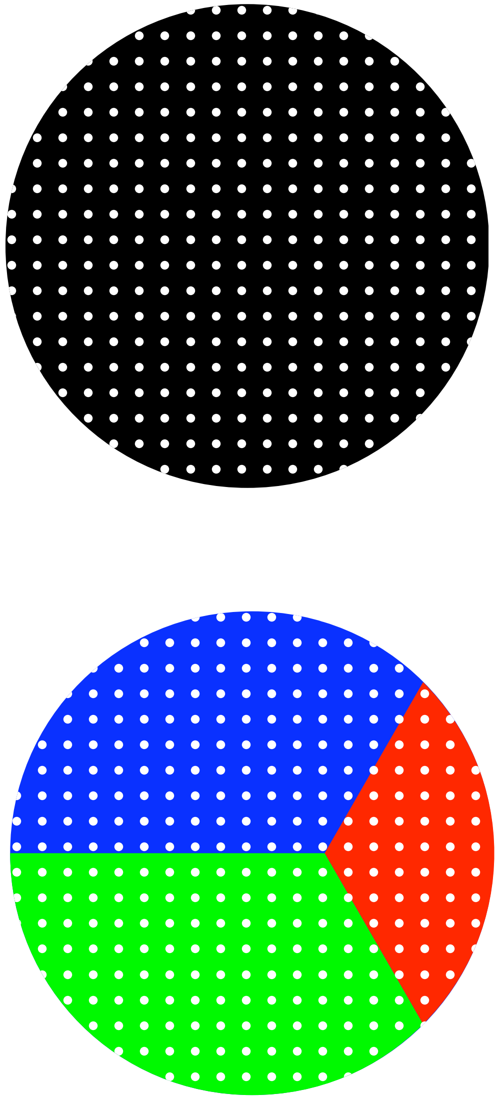
The Separable Case



Halving

- Discretized hypothesis space
- Predict by majority vote
- Remove 'wrong' hypotheses
- Note: can be applied in Bandit setting
- Mistake Bound $O(k^2\, d \log(D\, d))$
- Using JL lemma we can also obtain $O(k^2\, D \log(\frac{T+k}{\delta}) \log(D))$

The Separable Case



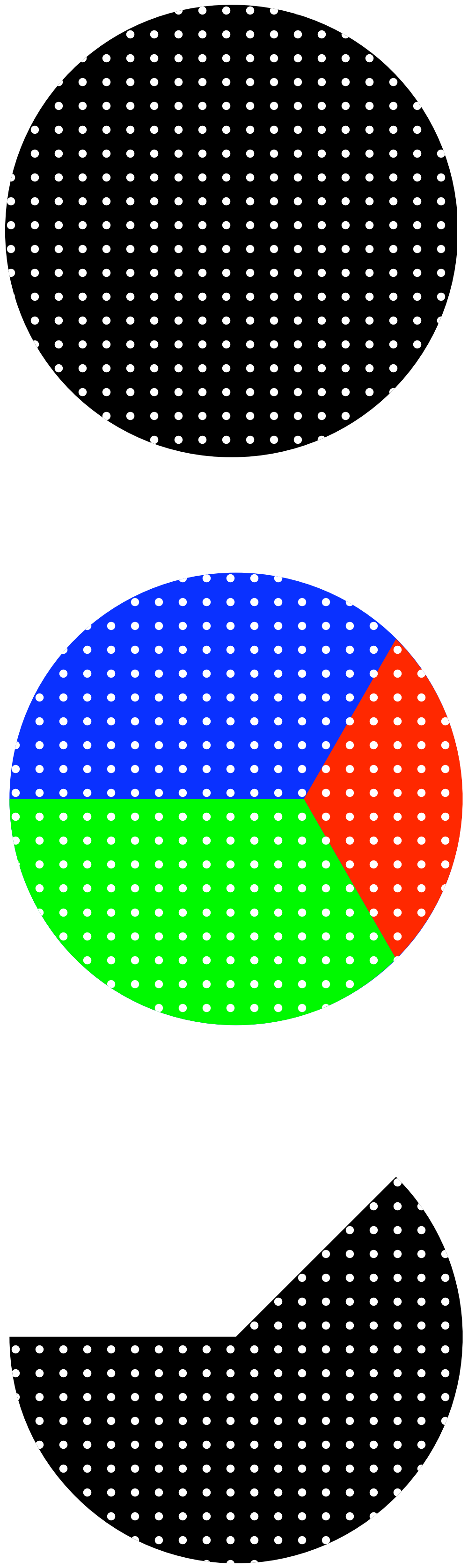
Halving

- Discretized hypothesis space
- Predict by majority vote
- Remove 'wrong' hypotheses
- Note: can be applied in Bandit setting
- Mistake Bound $O(k^2\, d \log(D\, d))$
- Using JL lemma we can also obtain $O(k^2\, D \log(\frac{T+k}{\delta}) \log(D))$

The Separable Case

Halving

- Discretized hypothesis space
- Predict by majority vote
- Remove 'wrong' hypotheses
- Note: can be applied in Bandit setting
- Mistake Bound $O(k^2\, d \log(D\, d))$
- Using JL lemma we can also obtain $O(k^2\, D \log(\frac{T+k}{\delta}) \log(D))$



Extensions and Open Problems

- Label Ranking
  - Predicting a “label ranking”
  - How to interpret feedback ?
- Multiplicative and Margin-based updates
  - Bandit versions of “Winnow” and “Passive-Aggressive”
- Deterministic vs. Randomized strategies
- Achievable rates ?
  - Efficient algorithms for the separable case ?