The price of bandit information in multiclass online classification

Amit Daniely and Tom Halbertal

The Hebrew University of Jerusalem

COLT 2013

Online multiclass classification: Full info vs Bandits

The Scenarios

- *The adversary chooses* $(x_t, y_t) \in \mathcal{X} \times [k]$ *and exposes* x_t.
- *The algorithm predicts* $\hat{y}_t$ *and receives* **feedback:**
  - **Full-info:** y_t *is revealed.*
  - **Bandit:** *Only an indication whether* $y_t = \hat{y}_t$ *is revealed.*

Online multiclass classification: Full info vs Bandits

The Scenarios

- *The adversary chooses* $(x_t, y_t) \in \mathcal{X} \times [k]$ *and exposes* x_t.
- *The algorithm predicts* $\hat{y}_t$ *and receives* **feedback:**
  - **Full-info:** y_t *is revealed.*
  - **Bandit:** *Only an indication whether* $y_t = \hat{y}_t$ *is revealed.*

- Clear motivation: Internet advertising, Recommendation Systems, ...
- Studied extensively in recent years.

Online multiclass classification: Full info vs Bandits

The Scenarios

- *The adversary chooses* $(x_t, y_t) \in \mathcal{X} \times [k]$ *and exposes* x_t.
- *The algorithm predicts* $\hat{y}_t$ *and receives* **feedback:**
  - **Full-info:** y_t *is revealed.*
  - **Bandit:** *Only an indication whether* $y_t = \hat{y}_t$ *is revealed.*

- Clear motivation: Internet advertising, Recommendation Systems, ...
- Studied extensively in recent years.

Question

How harder is it to predict in the (online) bandit scenario?

Online multiclass classification: Full info vs Bandits

The Scenarios

- *The adversary chooses* $(x_t, y_t) \in \mathcal{X} \times [k]$ *and exposes* x_t.
- *The algorithm predicts* $\hat{y}_t$ *and receives* **feedback:**
  - **Full-info:** y_t *is revealed.*
  - **Bandit:** *Only an indication whether* $y_t = \hat{y}_t$ *is revealed.*

- Clear motivation: Internet advertising, Recommendation Systems, ...
- Studied extensively in recent years.

Question

How harder is it to predict in the (online) bandit scenario?

Disclaimer: We focus on **information theoretic** bounds – non-efficient algorithms are legitimate.

Outline

Hypothesis classes based online learning

- To obtain meaningful learning scenario, we assume the sequence is realizable by some **hypothesis class** $\mathcal{H}$

$$\exists h \in \mathcal{H}, \quad \forall t, y_t = h(x_t)$$

Hypothesis classes based online learning

- To obtain meaningful learning scenario, we assume the sequence is realizable by some **hypothesis class** $\mathcal{H}$

$$\exists h \in \mathcal{H}, \;\; \forall t, y_t = h(x_t)$$

- Similar result holds for the non-realizable case (come to the poster!).

Setting – Mistake bounds

- The **full info error rate** of $\mathcal{H}$ is

$$\mathrm{Err}_{\mathcal{H}} \quad = \quad \text{num. of errors can be forced by full info adversary.}$$

Setting – Mistake bounds

- The **full info error rate** of $\mathcal{H}$ is

$$\mathrm{Err}_{\mathcal{H}} \quad = \quad \text{num. of errors can be forced by full info adversary.}$$

- The **bandit error rate** of $\mathcal{H}$ is

$$\text{B-}\mathrm{Err}_{\mathcal{H}} \quad = \quad \text{num. of errors can be forced by bandit adversary.}$$

Setting – Mistake bounds

- The **full info error rate** of $\mathcal{H}$ is

$$\mathrm{Err}_{\mathcal{H}} \quad = \quad \text{num. of errors can be forced by full info adversary.}$$

- The **bandit error rate** of $\mathcal{H}$ is

$$\text{B-}\mathrm{Err}_{\mathcal{H}} \quad = \quad \text{num. of errors can be forced by bandit adversary.}$$

Question

How large can the ratio $\mathrm{POB}_{\mathcal{H}} = \frac{\text{B-}\mathrm{Err}_{\mathcal{H}}}{\mathrm{Err}_{\mathcal{H}}}$ *be?*

A toy example

- Finite $\mathcal{X}$, $\mathcal{H} = [k]^{\mathcal{X}}$.

A toy example

- Finite $\mathcal{X}$, $\mathcal{H} = [k]^{\mathcal{X}}$.
- **Full info analysis:**
  - An upper bound: no reason to err twice on the same $x \in \mathcal{X}$.
    - $\Rightarrow \mathrm{Err}_{\mathcal{H}} \leq |\mathcal{X}|$.

A toy example

- Finite $\mathcal{X}$, $\mathcal{H} = [k]^{\mathcal{X}}$.
- **Full info analysis:**
  - An upper bound: no reason to err twice on the same $x \in \mathcal{X}$.
    - $\Rightarrow \mathrm{Err}_{\mathcal{H}} \leq |\mathcal{X}|$.
  - A lower bound: adversary can force an error on every $x \in \mathcal{X}$.
    - $\Rightarrow \mathrm{Err}_{\mathcal{H}} \geq |\mathcal{X}|$.

A toy example

- Finite $\mathcal{X}$, $\mathcal{H} = [k]^{\mathcal{X}}$.
- **Full info analysis:**
  - An upper bound: no reason to err twice on the same $x \in \mathcal{X}$.
    - $\Rightarrow \mathrm{Err}_{\mathcal{H}} \leq |\mathcal{X}|$.
  - A lower bound: adversary can force an error on every $x \in \mathcal{X}$.
    - $\Rightarrow \mathrm{Err}_{\mathcal{H}} \geq |\mathcal{X}|$.
- **Bandit analysis:**
  - An upper bound: no reason to err $\geq k-1$ times on the same $x \in \mathcal{X}$.
    - $\Rightarrow \text{B-}\mathrm{Err}_{\mathcal{H}} \leq (k-1) \cdot |\mathcal{X}|$.

A toy example

- Finite $\mathcal{X}$, $\mathcal{H} = [k]^{\mathcal{X}}$.
- **Full info analysis:**
  - An upper bound: no reason to err twice on the same $x \in \mathcal{X}$.
    - $\Rightarrow \mathrm{Err}_{\mathcal{H}} \leq |\mathcal{X}|$.
  - A lower bound: adversary can force an error on every $x \in \mathcal{X}$.
    - $\Rightarrow \mathrm{Err}_{\mathcal{H}} \geq |\mathcal{X}|$.
- **Bandit analysis:**
  - An upper bound: no reason to err $\geq k-1$ times on the same $x \in \mathcal{X}$.
    - $\Rightarrow \text{B-}\mathrm{Err}_{\mathcal{H}} \leq (k-1) \cdot |\mathcal{X}|$.
  - A lower bound: adversary can force $k-1$ errors on every $x \in \mathcal{X}$.
    - $\Rightarrow \text{B-}\mathrm{Err}_{\mathcal{H}} \geq (k-1) \cdot |\mathcal{X}|$.

A toy example

- Finite $\mathcal{X}$, $\mathcal{H} = [k]^{\mathcal{X}}$.
- **Full info analysis:**
  - An upper bound: no reason to err twice on the same $x \in \mathcal{X}$.
    - $\Rightarrow \mathrm{Err}_{\mathcal{H}} \leq |\mathcal{X}|$.
  - A lower bound: adversary can force an error on every $x \in \mathcal{X}$.
    - $\Rightarrow \mathrm{Err}_{\mathcal{H}} \geq |\mathcal{X}|$.
- **Bandit analysis:**
  - An upper bound: no reason to err $\geq k-1$ times on the same $x \in \mathcal{X}$.
    - $\Rightarrow \text{B-}\mathrm{Err}_{\mathcal{H}} \leq (k-1) \cdot |\mathcal{X}|$.
  - A lower bound: adversary can force $k-1$ errors on every $x \in \mathcal{X}$.
    - $\Rightarrow \text{B-}\mathrm{Err}_{\mathcal{H}} \geq (k-1) \cdot |\mathcal{X}|$.
- It follows that $\mathrm{POB}_{\mathcal{H}} = k-1$.

Main Result

Theorem

$\mathrm{POB}_{\mathcal{H}} \leq 4k \log(k).$

Main Result

Theorem

$\mathrm{POB}_{\mathcal{H}} \leq 4k \log(k)$.

- By the toy example, the theorem is tight, up to the $\log(k)$ factor.

Previous Results

$$\mathrm{POB}_{\mathcal{H}} = \frac{\text{B-Err}_{\mathcal{H}}}{\mathrm{Err}_{\mathcal{H}}}$$

Previous Results

$$\mathrm{POB}_{\mathcal{H}} = \frac{\text{B-Err}_{\mathcal{H}}}{\mathrm{Err}_{\mathcal{H}}}$$

- Known lower bounds on $\mathrm{Err}_{\mathcal{H}}$:
  - (Littlestone 89): In the *binary* case $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$.
  - (Daniely et al 11): $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$ also for multiclass classification.

Previous Results

$$\mathrm{POB}_{\mathcal{H}} = \frac{\text{B-Err}_{\mathcal{H}}}{\mathrm{Err}_{\mathcal{H}}}$$

- Known lower bounds on $\mathrm{Err}_{\mathcal{H}}$:
  - (Littlestone 89): In the *binary* case $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$.
  - (Daniely et al 11): $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$ also for multiclass classification.
- Known upper bounds on $\text{B-Err}_{\mathcal{H}}$:
  - (Halving Algorithm): $\text{B-Err}_{\mathcal{H}} \leq k \log(|\mathcal{H}|)$.
  - (Auer et al 03): Extension to the non-realizable case.
  - (Kakade et al 08): For the class of halfspaces with margin γ, $\text{B-Err}_{\mathcal{H}} = \tilde{O}\left(\frac{k^2}{\gamma}\right)$.
  - (Daniely et al 11): $\text{B-Err}_{\mathcal{H}} \leq \text{BL-Dim}(\mathcal{H})$.

Previous Results

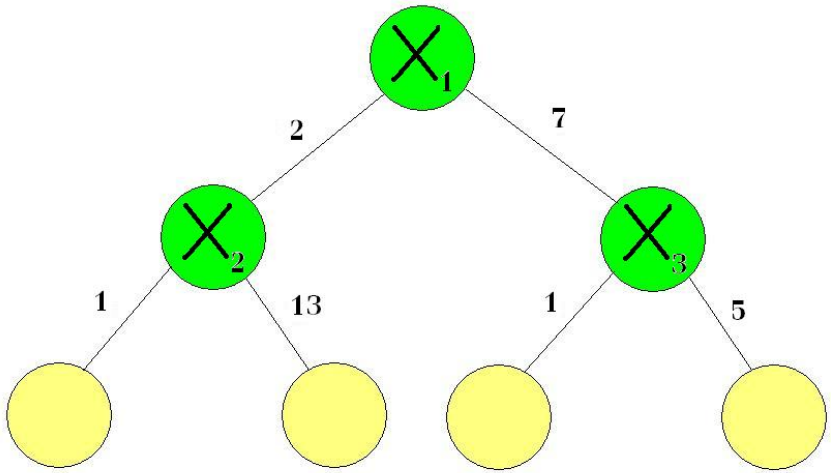
$$\mathrm{POB}_{\mathcal{H}} = \frac{\text{B-Err}_{\mathcal{H}}}{\mathrm{Err}_{\mathcal{H}}}$$

- Known lower bounds on $\mathrm{Err}_{\mathcal{H}}$:
  - (Littlestone 89): In the *binary* case $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$.
  - (Daniely et al 11): $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$ also for multiclass classification.
- Known upper bounds on $\text{B-Err}_{\mathcal{H}}$:
  - (Halving Algorithm): $\text{B-Err}_{\mathcal{H}} \leq k \log(|\mathcal{H}|)$.
  - (Auer et al 03): Extension to the non-realizable case.
  - (Kakade et al 08): For the class of halfspaces with margin γ, $\text{B-Err}_{\mathcal{H}} = \tilde{O}\left(\frac{k^2}{\gamma}\right)$.
  - (Daniely et al 11): $\text{B-Err}_{\mathcal{H}} \leq \text{BL-Dim}(\mathcal{H})$.
- None of the above yields a general upper bound on $\mathrm{POB}_{\mathcal{H}}$!

Outline

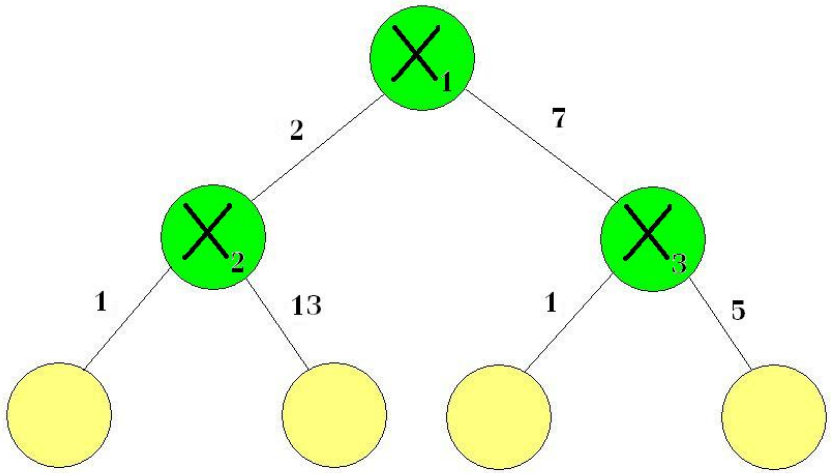
The Littlestone dimension (Littlestone, 89, Daniely et al, 11)

- Let $\mathcal{T}$ be a rooted full binary tree whose
  - Internal nodes are labelled by instances and
  - Whose edges are labelled by labels.



The Littlestone dimension (Littlestone, 89, Daniely et al, 11)

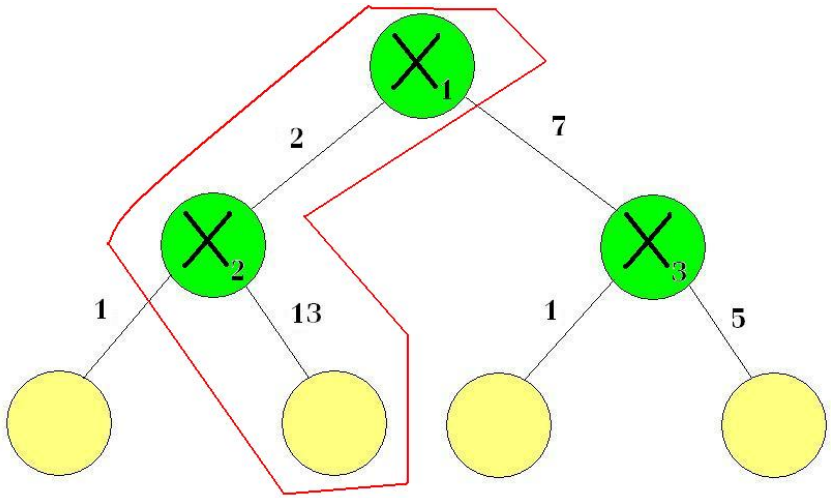
- Let $\mathcal{T}$ be a rooted full binary tree whose
  - Internal nodes are labelled by instances and
  - Whose edges are labelled by labels.



- A root to leaf path $x_1, \ldots, x_t$ **is realized** if there exists $h \in \mathcal{H}$ for which $h(x_i) =$ label of $x_i x_{i+1}$.

The Littlestone dimension (Littlestone, 89, Daniely et al, 11)

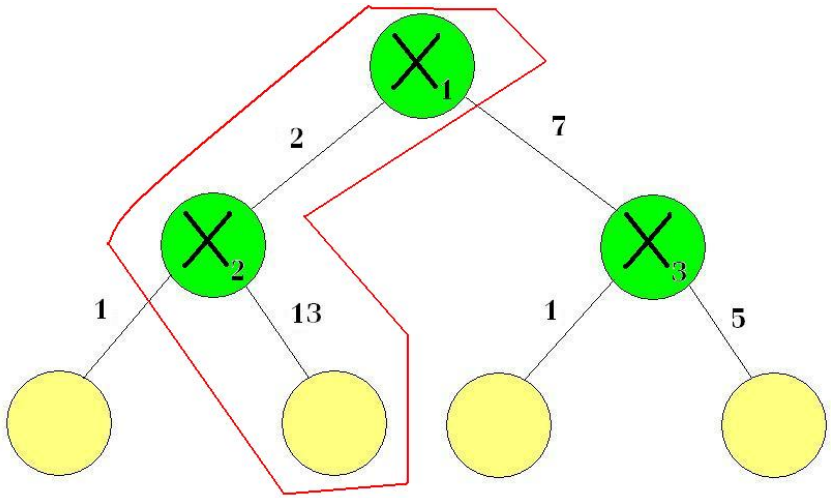
- Let $\mathcal{T}$ be a rooted full binary tree whose
  - Internal nodes are labelled by instances and
  - Whose edges are labelled by labels.



- A root to leaf path $x_1, \ldots, x_t$ **is realized** if there exists $h \in \mathcal{H}$ for which $h(x_i) =$ label of $x_i x_{i+1}$.

The Littlestone dimension (Littlestone, 89, Daniely et al, 11)

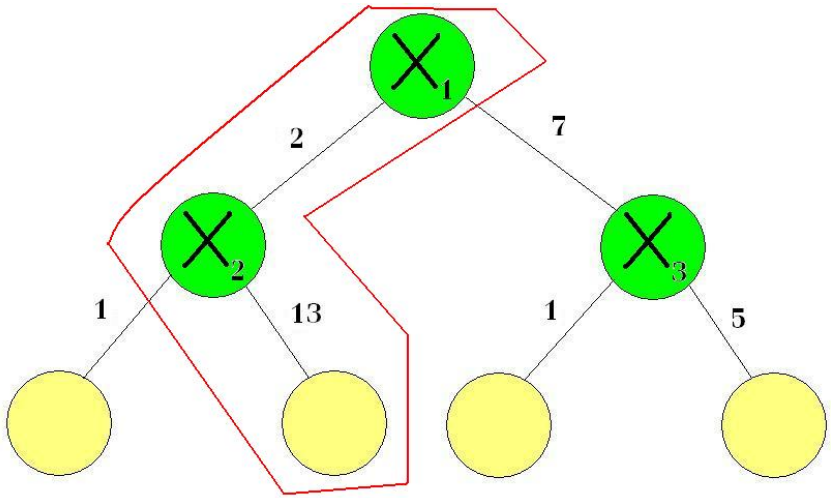
- Let $\mathcal{T}$ be a rooted full binary tree whose
  - Internal nodes are labelled by instances and
  - Whose edges are labelled by labels.



- A root to leaf path $x_1, \ldots, x_t$ **is realized** if there exists $h \in \mathcal{H}$ for which $h(x_i) =$ label of $x_i x_{i+1}$.
- The tree $\mathcal{T}$ is **shattered** is all its root to leaf paths are realizable.

The Littlestone dimension (Littlestone, 89, Daniely et al, 11)

- Let $\mathcal{T}$ be a rooted full binary tree whose
  - Internal nodes are labelled by instances and
  - Whose edges are labelled by labels.



- A root to leaf path $x_1, \ldots, x_t$ **is realized** if there exists $h \in \mathcal{H}$ for which $h(x_i) =$ label of $x_i x_{i+1}$.
- The tree $\mathcal{T}$ is **shattered** is all its root to leaf paths are realizable.
- The **Littlestone dimension**, $\text{L-Dim}(\mathcal{H})$, of $\mathcal{H}$ is the maximal depth of a shattered tree.

The Littlestone dimension (Littlestone, 89)

Theorem (Littlestone, 89)

$$\mathrm{Err}_{\mathcal{H}} = \text{L-Dim}(\mathcal{H})$$

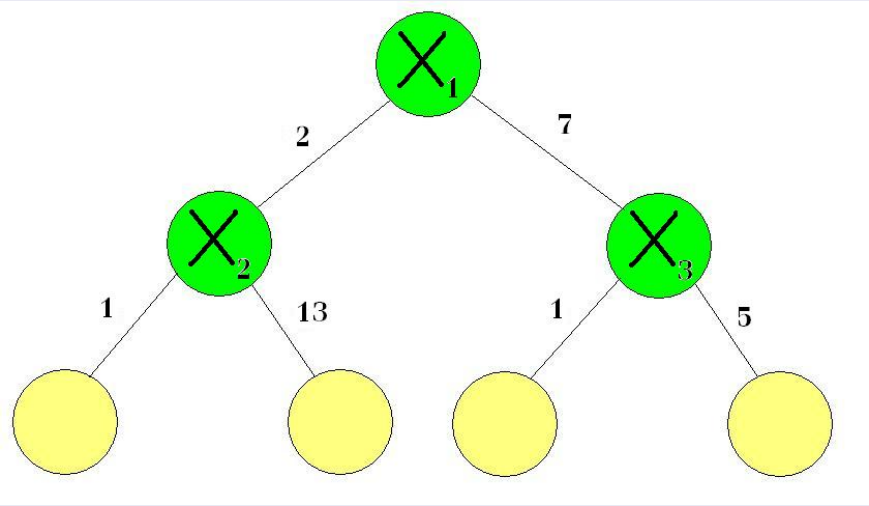
The Littlestone dimension (Littlestone, 89)

Theorem (Littlestone, 89)

$\mathrm{Err}_{\mathcal{H}} = \text{L-Dim}(\mathcal{H})$

Proof. $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$.

- Using a shattered tree of depth L, the adversary can easily force L mistakes.



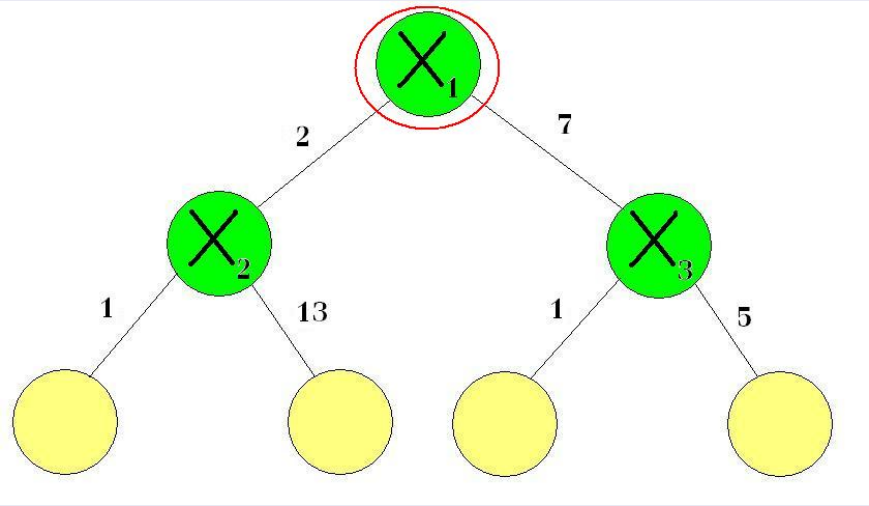
The Littlestone dimension (Littlestone, 89)

Theorem (Littlestone, 89)

$\mathrm{Err}_{\mathcal{H}} = \text{L-Dim}(\mathcal{H})$

Proof. $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$.

- Using a shattered tree of depth L, the adversary can easily force L mistakes.



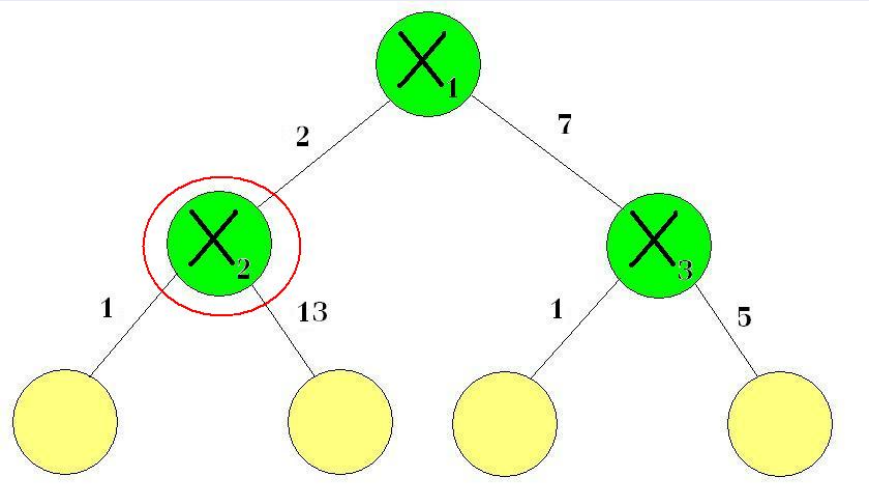
The Littlestone dimension (Littlestone, 89)

Theorem (Littlestone, 89)

$\mathrm{Err}_{\mathcal{H}} = \text{L-Dim}(\mathcal{H})$

Proof. $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$.

- Using a shattered tree of depth L, the adversary can easily force L mistakes.



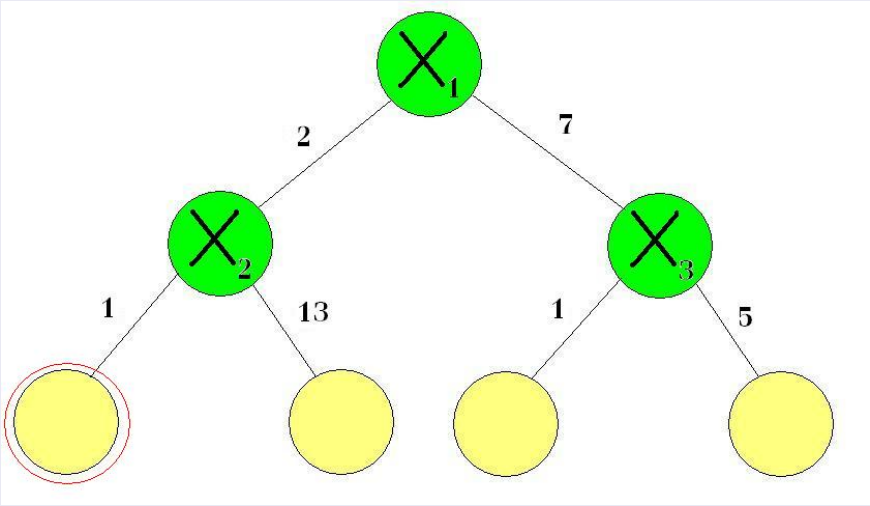
The Littlestone dimension (Littlestone, 89)

Theorem (Littlestone, 89)

$\mathrm{Err}_{\mathcal{H}} = \text{L-Dim}(\mathcal{H})$

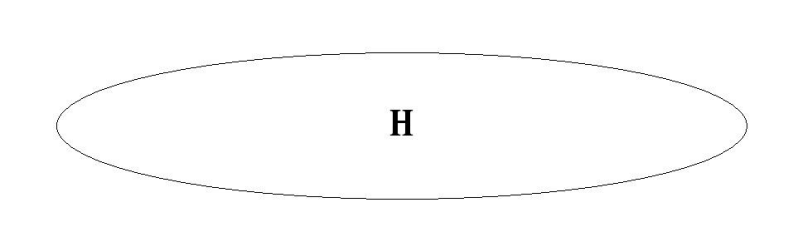
Proof. $\mathrm{Err}_{\mathcal{H}} \geq \text{L-Dim}(\mathcal{H})$.

- Using a shattered tree of depth L, the adversary can easily force L mistakes.



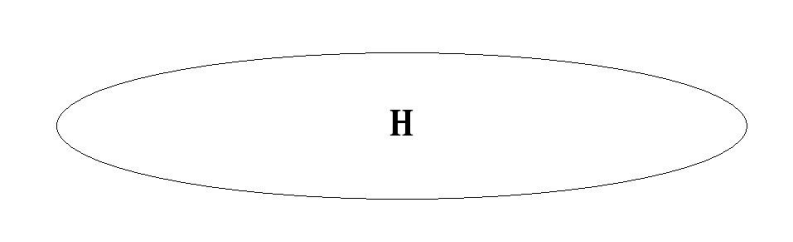
The Littlestone dimension (Littlestone, 89)

Proof. $\mathrm{Err}_{\mathcal{H}} \leq \text{L-Dim}(\mathcal{H})$.



The Littlestone dimension (Littlestone, 89)

Proof. $\mathrm{Err}_{\mathcal{H}} \leq \text{L-Dim}(\mathcal{H})$.

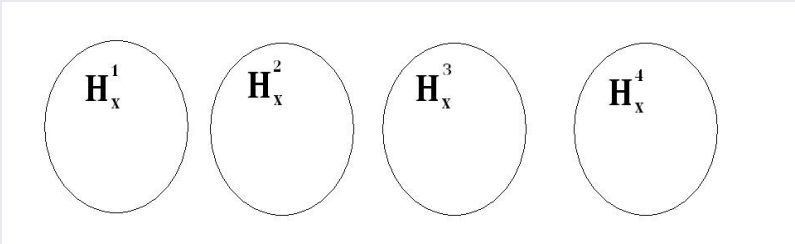


- Given $x \in \mathcal{X}$ chosen by the adversary, denote for every $y \in [k]$,

$$\mathcal{H}_x^y = \{h \in \mathcal{H} \mid h(x) = y\}$$

The Littlestone dimension (Littlestone, 89)

Proof. $\mathrm{Err}_{\mathcal{H}} \leq \text{L-Dim}(\mathcal{H})$.

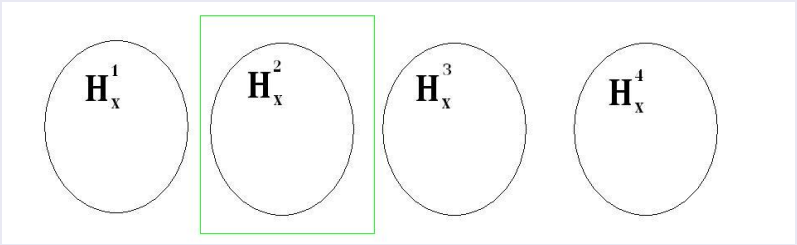


- Given $x \in \mathcal{X}$ chosen by the adversary, denote for every $y \in [k]$,

$$\mathcal{H}_x^y = \{h \in \mathcal{H} \mid h(x) = y\}$$

The Littlestone dimension (Littlestone, 89)

Proof. $\mathrm{Err}_{\mathcal{H}} \leq \text{L-Dim}(\mathcal{H})$.



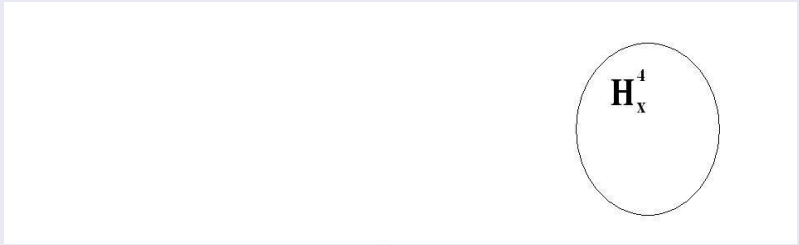
- Given $x \in \mathcal{X}$ chosen by the adversary, denote for every $y \in [k]$,

$$\mathcal{H}_x^y = \{h \in \mathcal{H} \mid h(x) = y\}$$

- The crucial point is that there is at most a single y for which $\text{L-Dim}(\mathcal{H}_x^y) = \text{L-Dim}(\mathcal{H})$, and the algorithm will choose that one.

The Littlestone dimension (Littlestone, 89)

Proof. $\mathrm{Err}_{\mathcal{H}} \leq \text{L-Dim}(\mathcal{H})$.



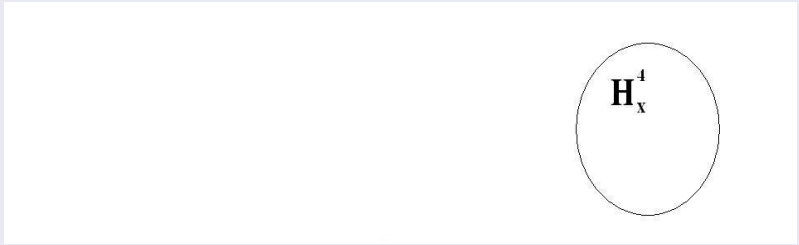
- Given $x \in \mathcal{X}$ chosen by the adversary, denote for every $y \in [k]$,

$$\mathcal{H}_x^y = \{h \in \mathcal{H} \mid h(x) = y\}$$

- The crucial point is that there is at most a single y for which $\text{L-Dim}(\mathcal{H}_x^y) = \text{L-Dim}(\mathcal{H})$, and the algorithm will choose that one.
- If it errs, the Littlestone dimension decreases by 1.

The Littlestone dimension (Littlestone, 89)

Proof. $\mathrm{Err}_{\mathcal{H}} \leq \text{L-Dim}(\mathcal{H})$.



- Given $x \in \mathcal{X}$ chosen by the adversary, denote for every $y \in [k]$,

$$\mathcal{H}_x^y = \{h \in \mathcal{H} \mid h(x) = y\}$$

- The crucial point is that there is at most a single y for which $\text{L-Dim}(\mathcal{H}_x^y) = \text{L-Dim}(\mathcal{H})$, and the algorithm will choose that one.
- If it errs, the Littlestone dimension decreases by 1.
- After $\text{L-Dim}(\mathcal{H})$ mistakes, $\text{L-Dim} = 0 \Rightarrow$ only a single consistent hypothesis is left.

Proof of the main theorem – a preface

Theorem

For every class $\mathcal{H}$, $\frac{\text{B-Err}_{\mathcal{H}}}{\text{Err}_{\mathcal{H}}} \leq 4k\log(k)$.

Equivalently,

Proof of the main theorem – a preface

Theorem

For every class $\mathcal{H}$, $\frac{\text{B-Err}_{\mathcal{H}}}{\text{Err}_{\mathcal{H}}} \leq 4k \log(k)$.

Equivalently,

Theorem

$\text{B-Err}(\mathcal{H}) \leq 4k \log(k) \, \text{L-Dim}(\mathcal{H})$

Proof of the main theorem – a preface

Theorem

For every class $\mathcal{H}$, $\frac{\text{B-Err}_{\mathcal{H}}}{\text{Err}_{\mathcal{H}}} \leq 4k\log(k)$.

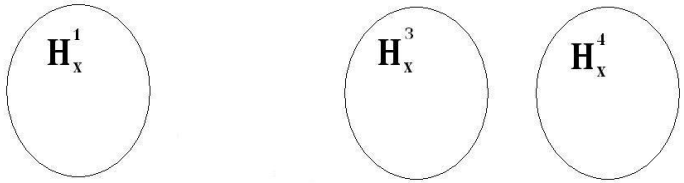
Equivalently,

Theorem

$\text{B-Err}(\mathcal{H}) \leq 4k\log(k)\,\text{L-Dim}(\mathcal{H})$

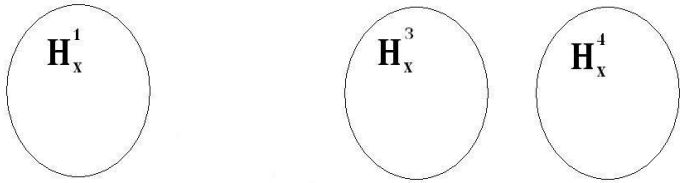
- Proceeding as before won't help – after an error, we are left with a union of $k-1$ classes of dimension $\text{L-Dim}(\mathcal{H})-1$.
  - The union might be of dimension $\text{L-Dim}(\mathcal{H})$!

Proof of the main theorem – a preface



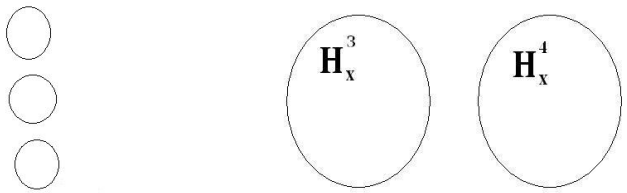
- But! $k-1$ classes of dimension $\text{L-Dim}(\mathcal{H})-1$ are, in some sense, indeed smaller than a class of dimension $\text{L-Dim}(\mathcal{H})$

Proof of the main theorem – a preface



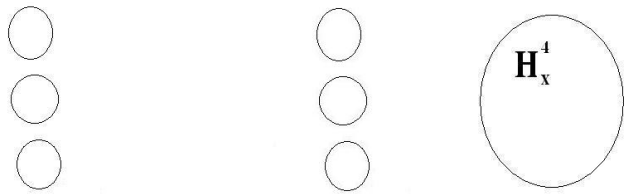
- But! $k-1$ classes of dimension $\text{L-Dim}(\mathcal{H})-1$ are, in some sense, indeed smaller than a class of dimension $\text{L-Dim}(\mathcal{H})$
  - In each of the next $k-1$ erroneous steps, we can split one of these classes into $k-1$ classes of Littlestone dimension $\text{L-Dim}(\mathcal{H})-2$.

Proof of the main theorem – a preface



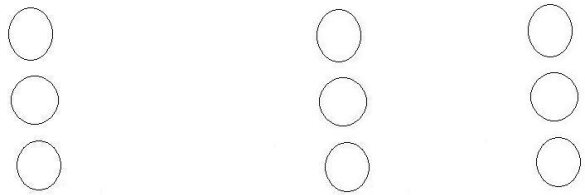
- But! $k-1$ classes of dimension $\text{L-Dim}(\mathcal{H})-1$ are, in some sense, indeed smaller than a class of dimension $\text{L-Dim}(\mathcal{H})$
  - In each of the next $k-1$ erroneous steps, we can split one of these classes into $k-1$ classes of Littlestone dimension $\text{L-Dim}(\mathcal{H})-2$.

Proof of the main theorem – a preface



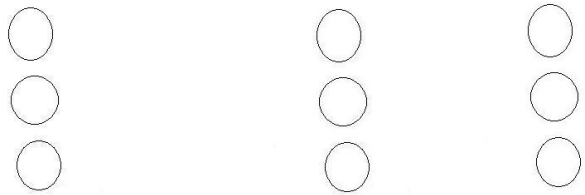
- But! $k-1$ classes of dimension $\text{L-Dim}(\mathcal{H})-1$ are, in some sense, indeed smaller than a class of dimension $\text{L-Dim}(\mathcal{H})$
  - In each of the next $k-1$ erroneous steps, we can split one of these classes into $k-1$ classes of Littlestone dimension $\text{L-Dim}(\mathcal{H})-2$.

Proof of the main theorem – a preface



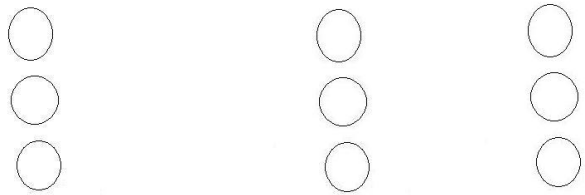
- But! $k-1$ classes of dimension $\text{L-Dim}(\mathcal{H})-1$ are, in some sense, indeed smaller than a class of dimension $\text{L-Dim}(\mathcal{H})$
  - In each of the next $k-1$ erroneous steps, we can split one of these classes into $k-1$ classes of Littlestone dimension $\text{L-Dim}(\mathcal{H})-2$.

Proof of the main theorem – a preface



- But! $k-1$ classes of dimension $\text{L-Dim}(\mathcal{H})-1$ are, in some sense, indeed smaller than a class of dimension $\text{L-Dim}(\mathcal{H})$
  - In each of the next $k-1$ erroneous steps, we can split one of these classes into $k-1$ classes of Littlestone dimension $\text{L-Dim}(\mathcal{H})-2$.
  - Eventually, we will be left with a single function!
    - However, the number of steps would be **exponential in** $\text{L-Dim}(\mathcal{H})$.

Proof of the main theorem – a preface



- But! $k-1$ classes of dimension $\text{L-Dim}(\mathcal{H})-1$ are, in some sense, indeed smaller than a class of dimension $\text{L-Dim}(\mathcal{H})$
  - In each of the next $k-1$ erroneous steps, we can split one of these classes into $k-1$ classes of Littlestone dimension $\text{L-Dim}(\mathcal{H})-2$.
  - Eventually, we will be left with a single function!
    - However, the number of steps would be **exponential in** $\text{L-Dim}(\mathcal{H})$.
- Nevertheless, the idea of the algorithm would be to maintain a collection of subclasses of $\mathcal{H}$, and shrink it more carefully.

Proof of the main theorem – the splitting mechanism

- Define the **capacity** of $\mathcal{V} \subset \mathcal{H}$ by

$$C(\mathcal{V}) = k^{2\,\text{L-Dim}(\mathcal{V})}$$

Proof of the main theorem – the splitting mechanism

- Define the **capacity** of $\mathcal{V} \subset \mathcal{H}$ by

$$C(\mathcal{V}) = k^{2\,\text{L-Dim}(\mathcal{V})}$$

- The capacity of a *collection* Ψ of subclasses of $\mathcal{H}$ is

$$C(\Psi) = \sum_{\mathcal{V} \in \Psi} C(\mathcal{V})$$

Proof of the main theorem – the splitting mechanism

- Define the **capacity** of $\mathcal{V} \subset \mathcal{H}$ by

$$C(\mathcal{V}) = k^{2\,\text{L-Dim}(\mathcal{V})}$$

- The capacity of a *collection* Ψ of subclasses of $\mathcal{H}$ is

$$C(\Psi) = \sum_{\mathcal{V} \in \Psi} C(\mathcal{V})$$

- Define a splitting operator:

$$\text{split}\,(\mathcal{V}, x, y) = \left\{ \mathcal{V}_x^{y'} \mid y' \neq y, \mathcal{V}_x^{y'} \neq \emptyset \right\}$$

Proof of the main theorem – the splitting mechanism

- Define the **capacity** of $\mathcal{V} \subset \mathcal{H}$ by

$$C(\mathcal{V}) = k^{2\,\text{L-Dim}(\mathcal{V})}$$

- The capacity of a *collection* Ψ of subclasses of $\mathcal{H}$ is

$$C(\Psi) = \sum_{\mathcal{V} \in \Psi} C(\mathcal{V})$$

- Define a splitting operator:

$$\text{split}\left(\mathcal{V}, x, y\right) = \left\{ \mathcal{V}_x^{y'} \mid y' \neq y, \mathcal{V}_x^{y'} \neq \emptyset \right\}$$

$$\text{split}\left(\Psi, x, y\right) = \cup_{\mathcal{V} \in \Psi} \text{split}\left(\mathcal{V}, x, y\right)$$

Proof of the main theorem – the algorithm

- Initialize $\Psi = \{\mathcal{H}\}$.
- For $t = 1, 2 \ldots$
  - Receive x_t
  - Predict the label $\hat{y}_t$ that minimizes $C(\text{split}(\Psi, x_t, \hat{y}_t))$
  - If $\hat{y}_t \neq y_t$, update $\Psi = \text{split}(\Psi, x_t, \hat{y}_t)$.

Proof of the main theorem – analysis

- **The crucial point:** If $\mathcal{V}$ is splitted into $k-1$ class of smaller Littlestone dimension, it capacity shrinks by a factor of

$$\frac{1}{k}$$

Proof of the main theorem – analysis

- **The crucial point:** If $\mathcal{V}$ is splitted into $k-1$ class of smaller Littlestone dimension, it capacity shrinks by a factor of

$$\frac{1}{k}$$

- $\Rightarrow$ *Some* $\hat{y}$ shrinks the capacity of $1/k$ of the classes by $1/k$. The *total* capacity is therefore shrinked by

$$1-\frac{1}{2k}$$

Proof of the main theorem – analysis

- After M mistakes, the capacity shrinks by

$$\left(1 - \frac{1}{2k}\right)^M$$

Proof of the main theorem – analysis

- After M mistakes, the capacity shrinks by

$$\left(1 - \frac{1}{2k}\right)^M$$

- The initial capacity is $k^{2\,\text{L-Dim}(\mathcal{H})}$
  - $\Rightarrow$ After $\leq 4k\log(k)\,\text{L-Dim}(\mathcal{H})$ mistakes, the capacity shrinks to 1!
  - Only a *single* consistent hypothesis is left!

Conclusion and open questions

- **Main result:** The price of bandit information in online classification is $\leq 4k \log(k)$.

Conclusion and open questions

- **Main result:** The price of bandit information in online classification is $\leq 4k \log(k)$.
  - A similar result holds for the non realizable case (come to the poster!).

Conclusion and open questions

- **Main result:** The price of bandit information in online classification is $\leq 4k\log(k)$.
  - A similar result holds for the non realizable case (come to the poster!).
  - An application: calculating the bandit error rate of large margin multiclass linear classifiers (come to the poster!).

Conclusion and open questions

- **Main result:** The price of bandit information in online classification is $\leq 4k\log(k)$.
  - A similar result holds for the non realizable case (come to the poster!).
  - An application: calculating the bandit error rate of large margin multiclass linear classifiers (come to the poster!).

Some open questions:

- Is the $\log(k)$ factor necessary?

Conclusion and open questions

- **Main result:** The price of bandit information in online classification is $\leq 4k\log(k)$.
  - A similar result holds for the non realizable case (come to the poster!).
  - An application: calculating the bandit error rate of large margin multiclass linear classifiers (come to the poster!).

Some open questions:

- Is the $\log(k)$ factor necessary?
- What if we restrict to *efficient* algorithms?