Mixed strategies

	q_1	q_2	q_3	q_4	q_5
p_1	1	0	1	0	1
p_2	-1	0	0	1	1
p_3	1	0	-1	1	0

- pure strategies: each player chooses a single action.
- mixed strategies: each player chooses a distribution over actions.
- Expeted gain/loss: $\vec{p}M\vec{q}^T$

The Minimax theorem

John Von-Neumann, 1928

$$\max_{\vec{p}} \min_{\vec{q}} \vec{p} M \vec{q}^{\,T} = \min_{\vec{q}} \max_{\vec{p}} \vec{p} M \vec{q}^{\,T}$$

- Unlike pure strategies, the order of choice of mixed strategies does not matter.
- Optimal mixed strateigies: the strategies that achieve the minimax.
- Value of the game: the value of the minimax.
- Finding the minimax strategies when the matrix is known = Linear Programming.

A matrix corresponding to online learning.

$t =$	1	2	3	4	...
expert 1	1	0	1	0	...
expert 2	-1	0	0	1	...
expert 3	1	0	-1	1	...

- The columns are revealed one at a time. strategies does not matter.
- Using Hedge or NormalHedge the row player chooses a mixed strategy over the rows that is almost as good as the best single row in hind-sight.
- The best single row in hind-site is at least as good as any mixed strategy in hind-sight.

A matrix corresponding to online learning.

$t =$	1	2	3	4	...
expert 1	1	0	1	0	...
expert 2	-1	0	0	1	...
expert 3	1	0	-1	1	...

- If the adversary playes optimally, then the row distribution coverges to a minimax optimal mixed strategy.
- But adversary might not play optimally - minimizing regret is a stronger criterion than converging to minimax optimal mixed strategy.

A matrix corresponding to boosting

	ex. 1	ex. 2	ex. 3	ex. 4	...
base rule 1	1	0	1	0	...
base rule 2	0	0	0	1	...
base rule 3	1	0	0	1	...

- 0 mistake, 1 correct.
- A weak learning algorithm: can find a base rule whose weighted error is smaller than $1/2 - \gamma$ or any distribution over the examples.
- There is a distribution over the base rules such that for any example the expected error is smaller $1/2 - \gamma$.
- Implies that the majority vote wrt this distribution over base rules is correct on **all** examples.
- Moreover - the weight of the majority is at least $1/2 + \gamma$, the minority is at most $1/2 - \gamma$.

$h_1(x_1)=y_1$ means 1 is correct, 0 if incorrect

	h_1	h_2	h_3
x_1	$h_1(x_1)=y_1$	$h_2(x_1)=y_1$	$h_3(x_1)=y_1$
x_2	$h_1(x_2)=y_2$	$h_2(x_2)=y_2$	$h_3(x_2)=y_2$
x_3	$h_1(x_3)=y_3$	$h_2(x_3)=y_3$	$h_3(x_3)=y_3$
x_4	$h_1(x_4)=y_4$	$h_2(x_4)=y_4$	$h_3(x_4)=y_4$

Boosting and the Minmax Theorem

- γ-weak learning assumption:
  - for every distribution on examples
  - can find weak classifier with weighted error $\leq \frac{1}{2} - \gamma$
- equivalent to:

$$(\text{value of game } \mathbf{M}) \geq \frac{1}{2} + \gamma$$

- by minmax theorem, implies that:
  - $\exists$ some weighted majority classifier that correctly classifies all training examples with margin $\geq 2\gamma$
  - further, weights are given by maxmin strategy of game $\mathbf{M}$

Idea for Boosting

- maxmin strategy of $\mathbf{M}$ has perfect (training) accuracy and large margins
- find approximately using earlier algorithm for solving a game
  - i.e., apply MW to $\mathbf{M}$
- yields (variant of) AdaBoost

AdaBoost and Game Theory

- summarizing:
  - weak learning assumption implies maxmin strategy for $\mathbf{M}$ defines large-margin classifier
  - AdaBoost finds maxmin strategy by applying general algorithm for solving games through repeated play
- consequences:
  - weights on weak classifiers converge to (approximately) maxmin strategy for game $\mathbf{M}$
  - (average) of distributions D_t converges to (approximately) minmax strategy
  - margins and edges connected via minmax theorem
  - explains why AdaBoost maximizes margins
- different instantiation of game-playing algorithm gives online learning algorithms (such as weighted majority algorithm)

Boosting as a Game

- Mindy (row player) $\leftrightarrow$ booster
- Max (column player) $\leftrightarrow$ weak learner
- matrix $\mathbf{M}$:
  - row $\leftrightarrow$ training example
  - column $\leftrightarrow$ weak classifier
  - $\mathbf{M}(i,j) =$
    $\begin{cases} 1 & \text{if } j\text{-th weak classifier correct on } i\text{-th training example} \\ 0 & \text{else} \end{cases}$
  - encodes which weak classifiers correct on which examples
  - huge # of columns — one for every possible weak classifier

A weighted training set

Feature vectors

Binary labels {-1,+1}

Positive weights

$$(x_1, y_1, w_1), (x_2, y_2, w_2), \ldots, (x_m, y_m, w_m)$$

A weak learner

Weighted training set

$(x_1, y_1, w_1), (x_2, y_2, w_2), \ldots, (x_m, y_m, w_m)$

Weak Learner

A weak rule

h

instances

$x_1, x_2, \ldots, x_m$

h

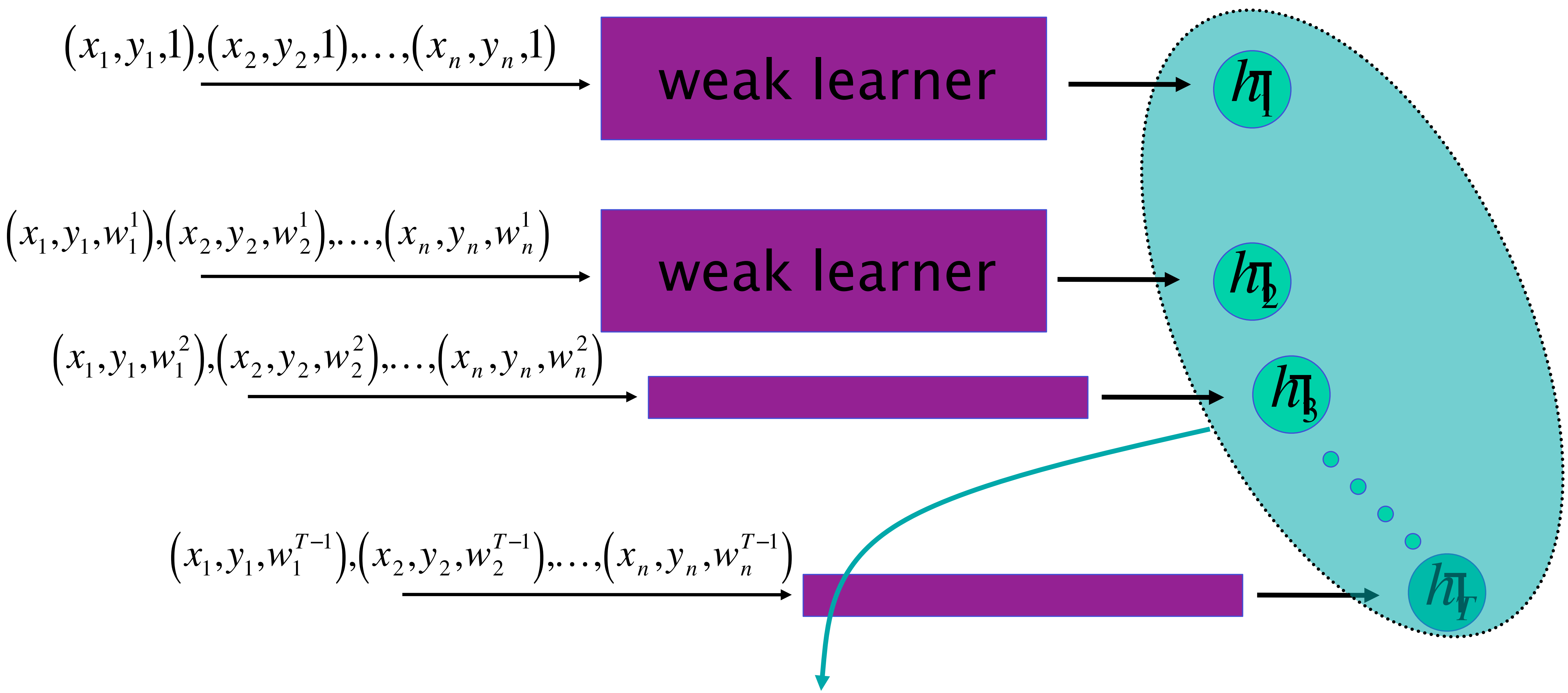
predictions

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m; \quad \hat{y}_i \in \{0,1\}$

The weak requirement:

$$\left| \frac{\sum_{i=1}^{m} y_i \hat{y}_i w_i}{\sum_{i=1}^{m} w_i} \right| > \gamma > 0$$

The boosting process



$$F_T(x) = \alpha_1 h_1(x) + \alpha_2 h_2(x) + \dots + \alpha_T h_T(x)$$

Adaboost

Freund, Schapire 1997

$$F_0(x) \equiv 0$$

$$\text{for } \boxed{t} = 1..T$$

$$\boxed{w_i^t} = \exp\left(-y_i F_{t-1}(x_i)\right)$$

$$\text{Get } \boxed{h_t} \text{ from } \textit{weak} - \textit{learner}$$

$$\boxed{\alpha_t} = \tfrac{1}{2}\ln\left(\sum_{i:h_t(x_i)=1, y_i=1} w_i^t \Big/ \sum_{i:h_t(x_i)=1, y_i=-1} w_i^t\right)$$

$$\boxed{F_{t+1}} = F_t + \alpha_t h_t$$

Main property of Adaboost

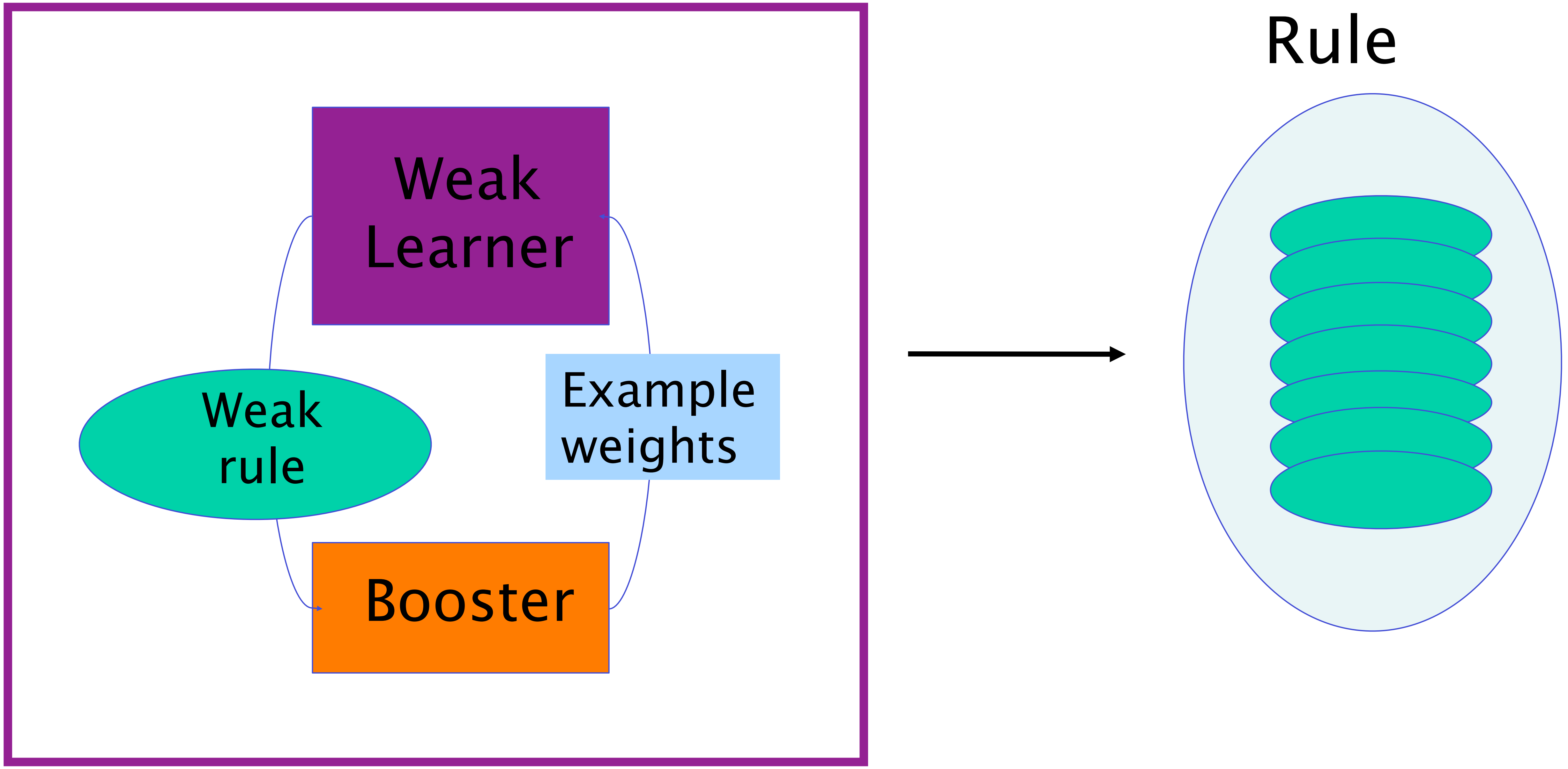
If advantages of weak rules over random guessing are: $\gamma_1, \gamma_2, .., \gamma_T$ then training error of final rule is at most

$$\hat{\varepsilon}(f_T) \leq \exp\left(-\sum_{t=1}^{T} \gamma_t^2\right)$$

Boosting block diagram

Strong Learner

Accurate Rule

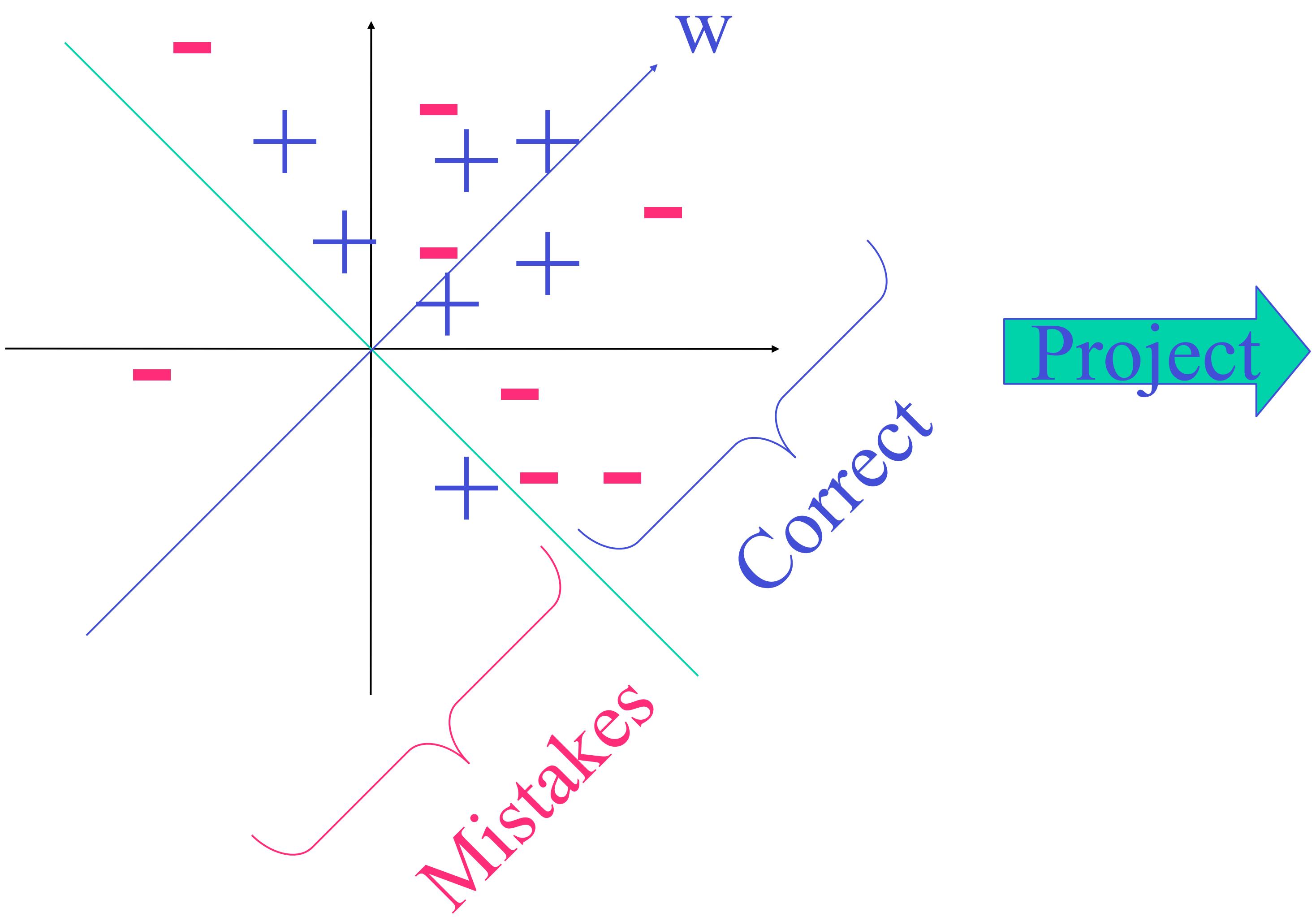


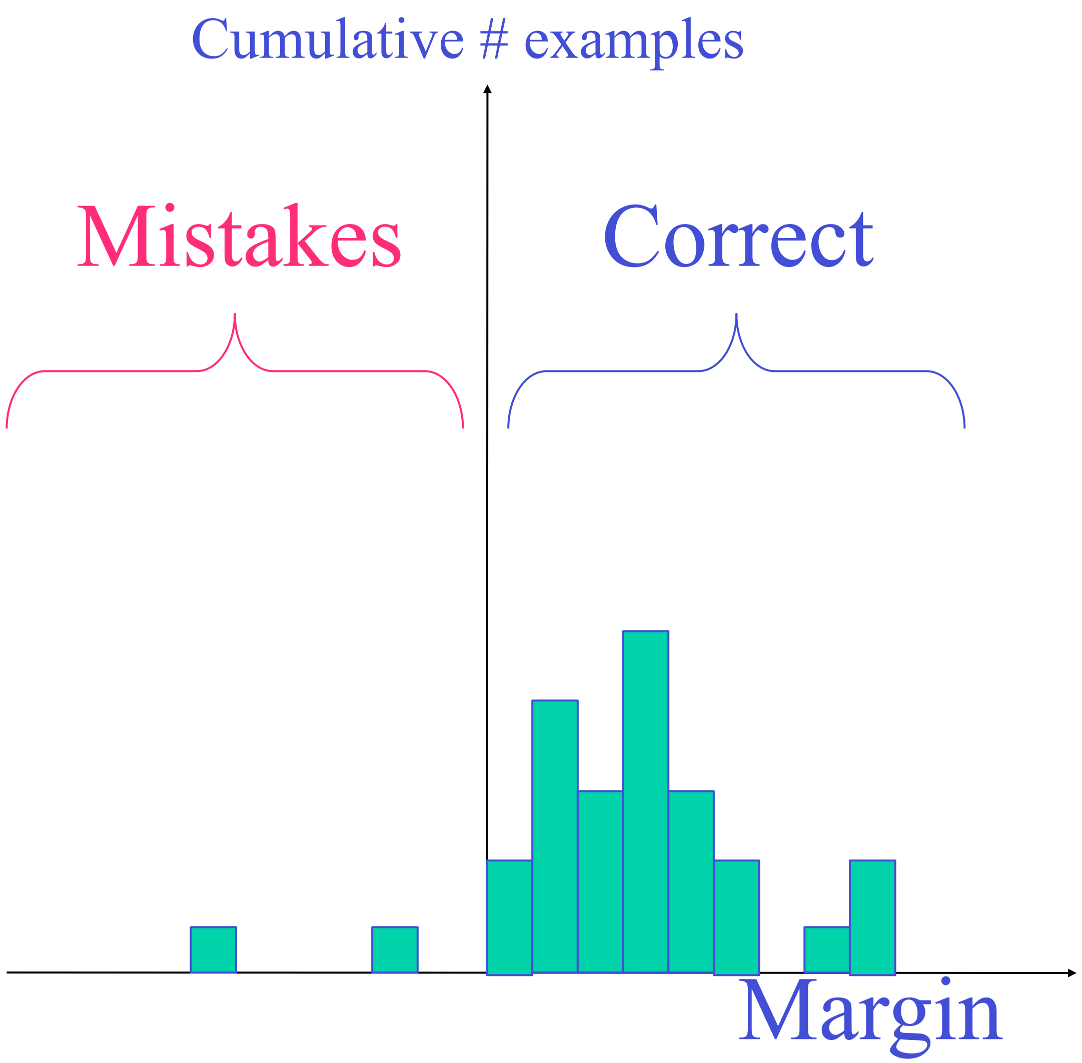
Margins view

$$x, w \in R^n; y \in \{-1,+1\}$$

Prediction = $sign(w \bullet x)$

$$\text{Margin} = \frac{y(w \bullet x)}{\|w\|_p \times \|x\|_q}$$



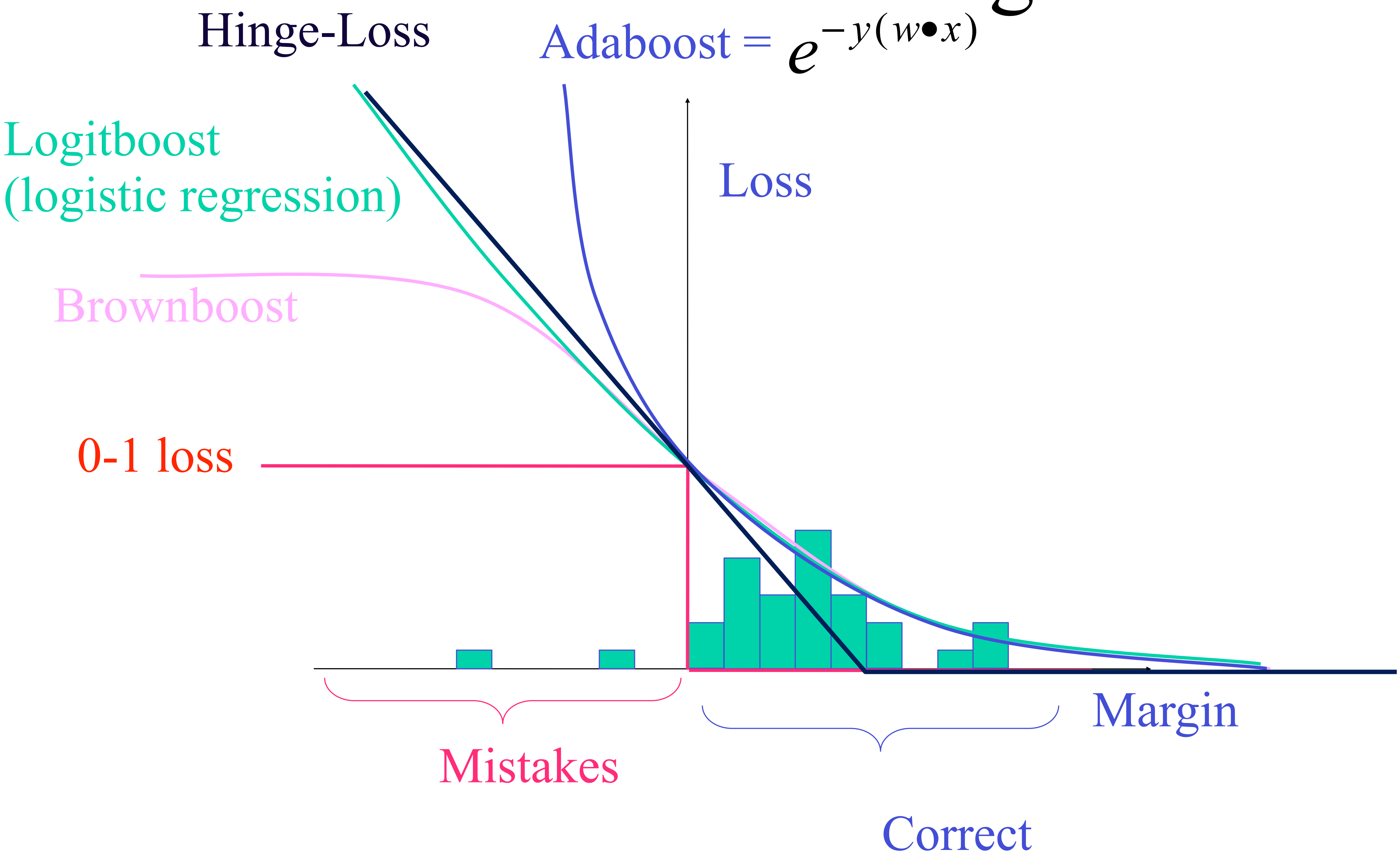


SVM vs Adaboost

$$\text{Margin} = \frac{y(w \bullet x)}{\|w\|_p \times \|x\|_q}$$

	Norms	Algorithm
SVM:	$p = q = \frac{1}{2}$	Quadratic optimization
Adaboost:	$p = 1, \quad q = \infty$	Coordinate-wise descent

Minimizing error using loss functions on margin



What is a typical weak learner?

- Define a large set of features $(x_1, x_2, \ldots, x_n)$
- Decision stumps:

$$h(x) = \begin{cases} +1 & \text{if } x_i \geq \theta \\ -1 & \text{otherwise} \end{cases}$$

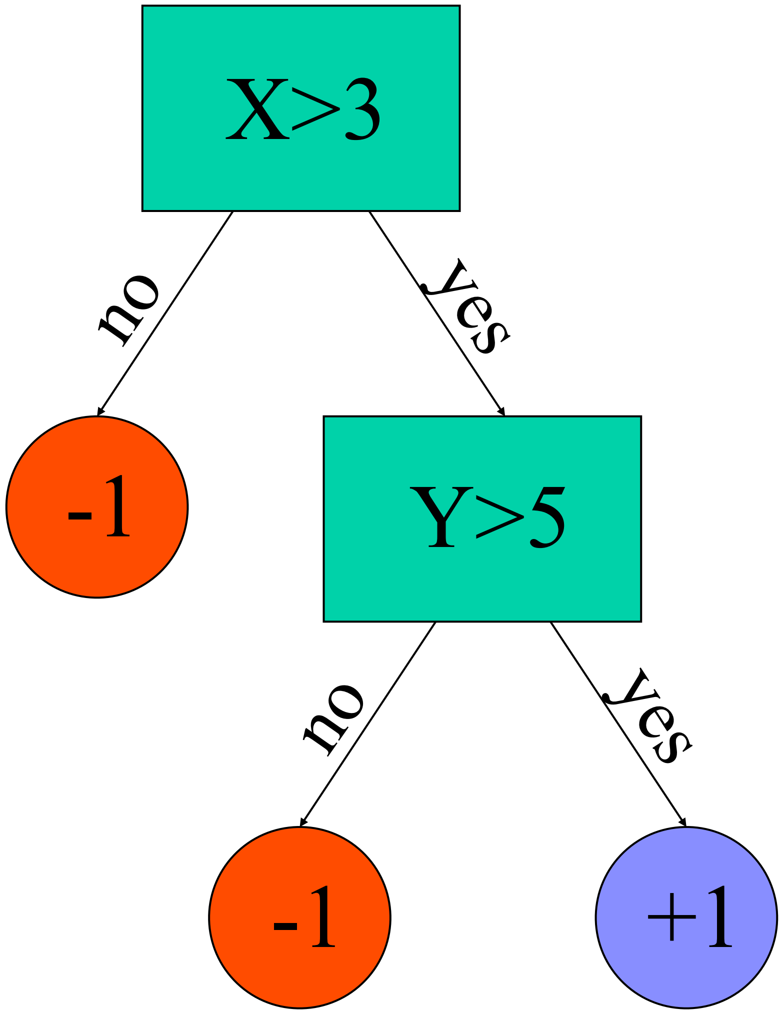
- Speciaists (particularly for multi-class):

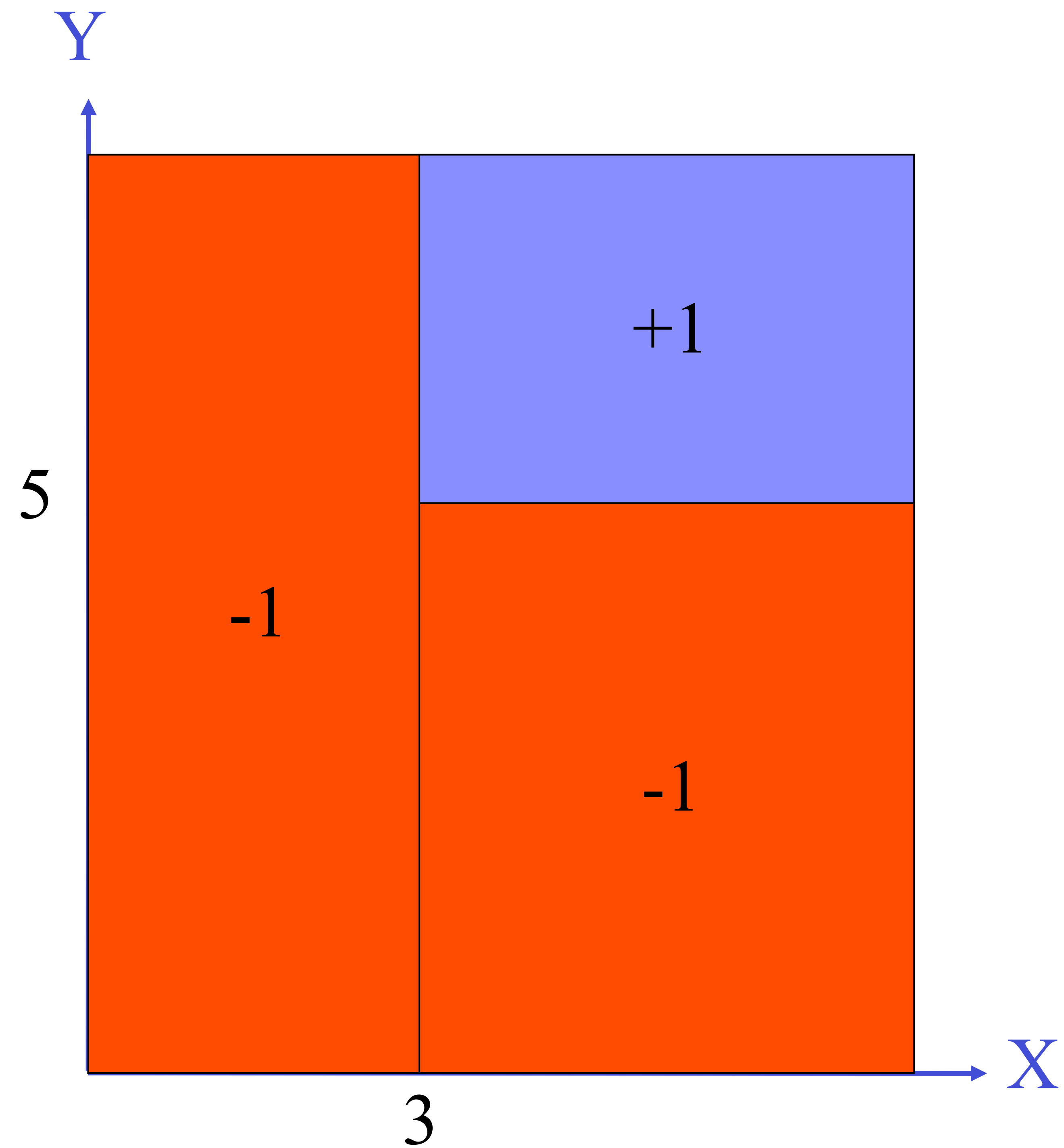
$$h(x) = \begin{cases} \text{class } j & \text{if } x_i \geq \theta \ (x_i < \theta) \\ 0 & \text{otherwise} \end{cases}$$

Alternating Trees

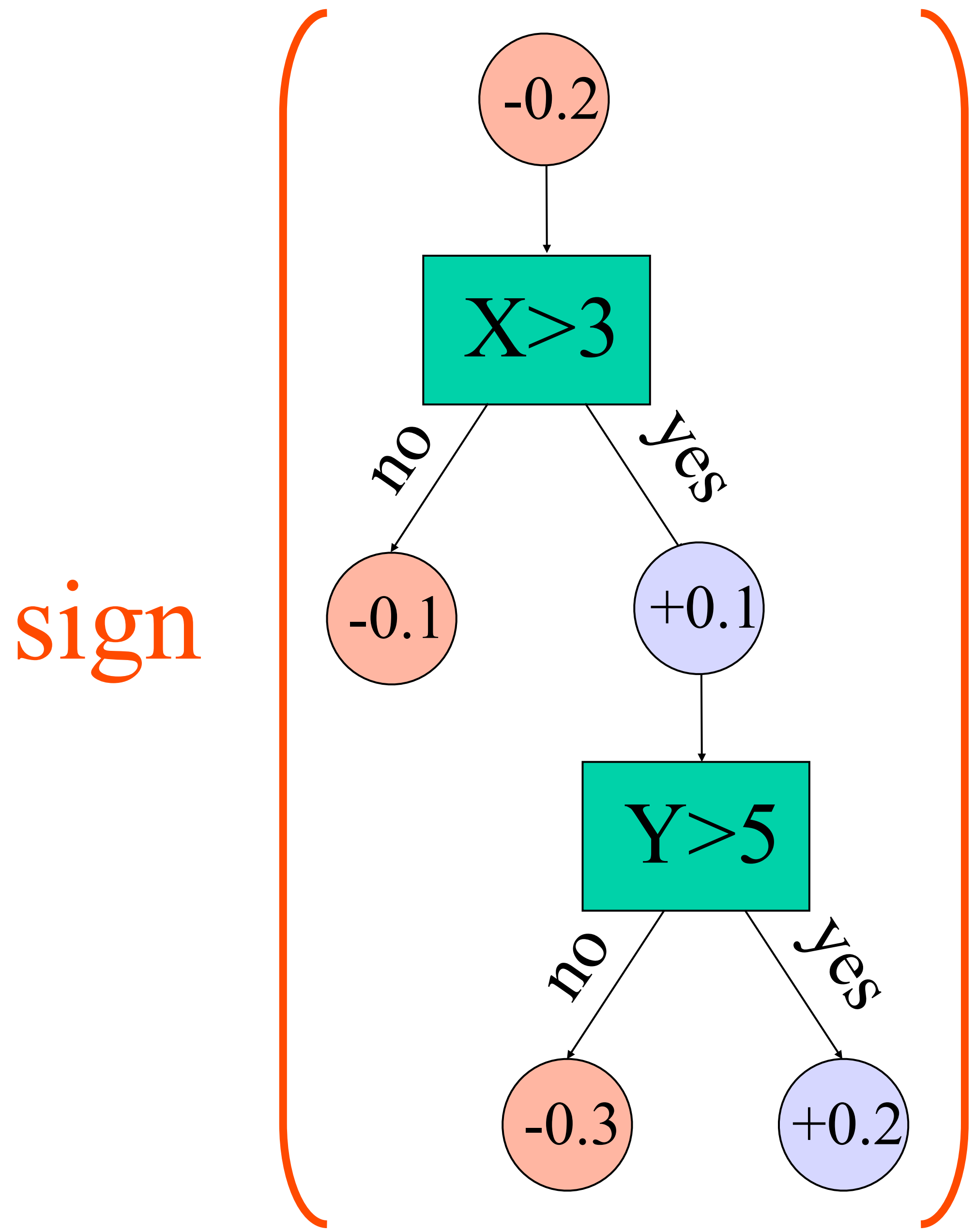
Joint work with Llew Mason

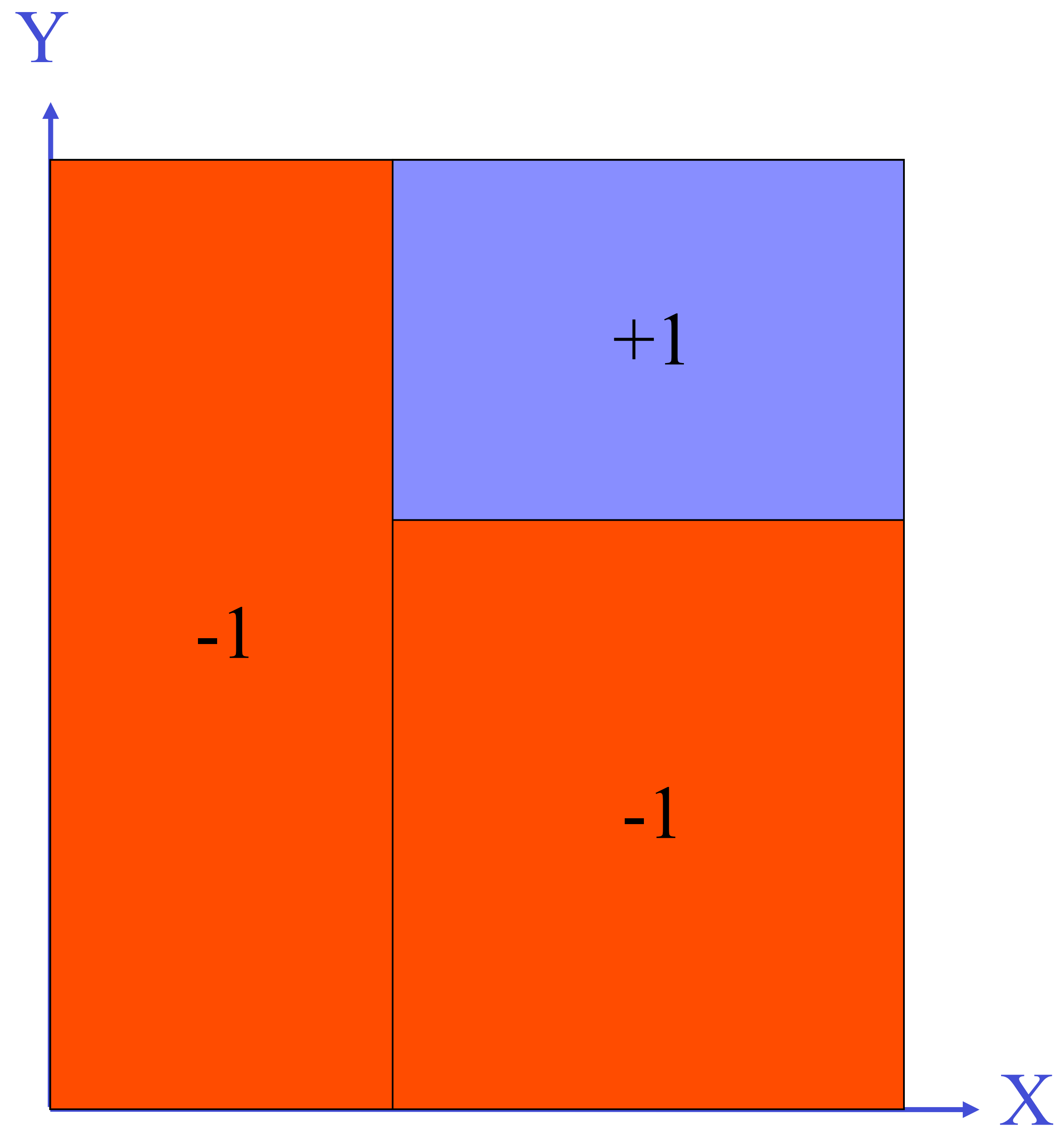
Decision Trees



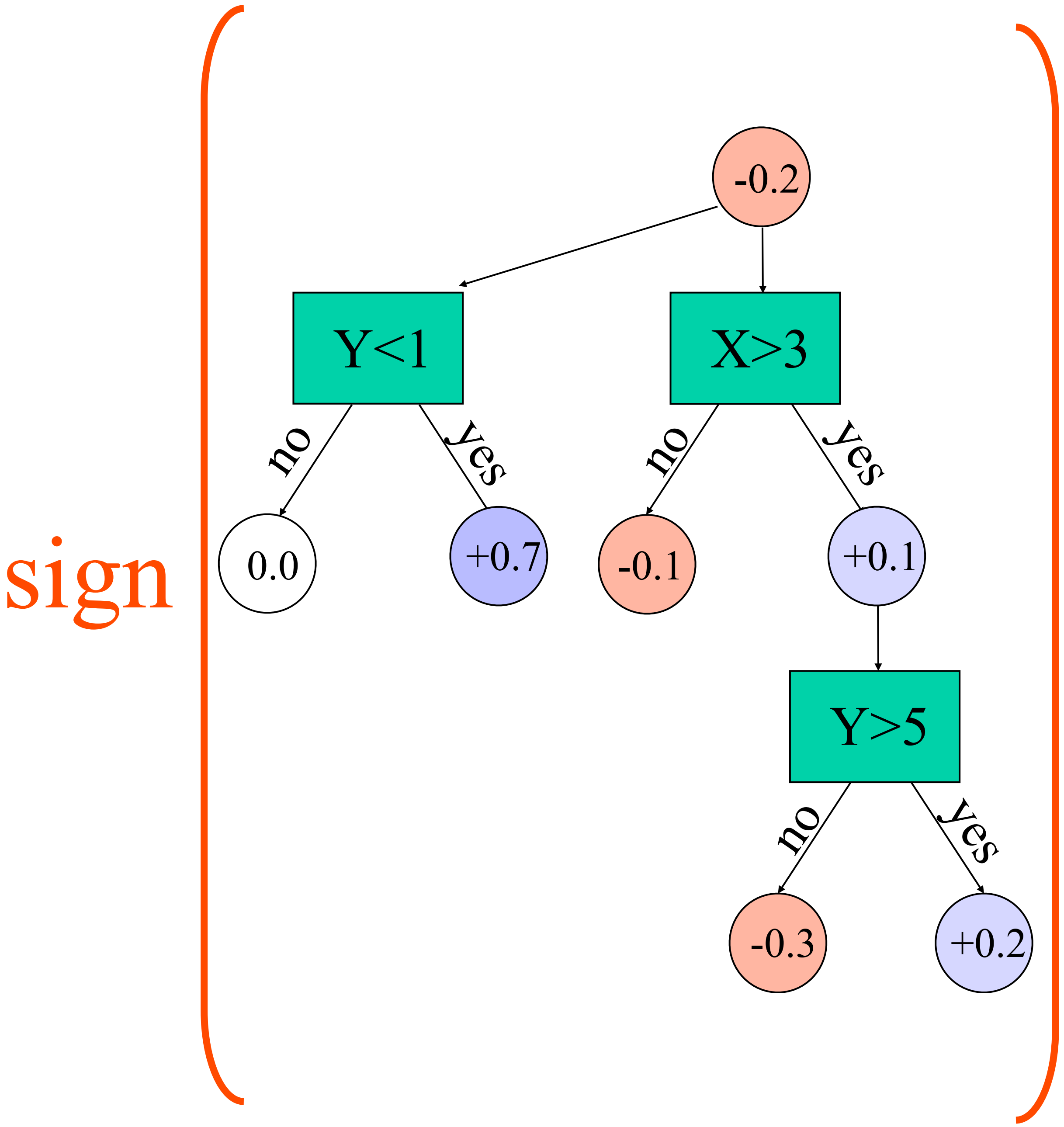


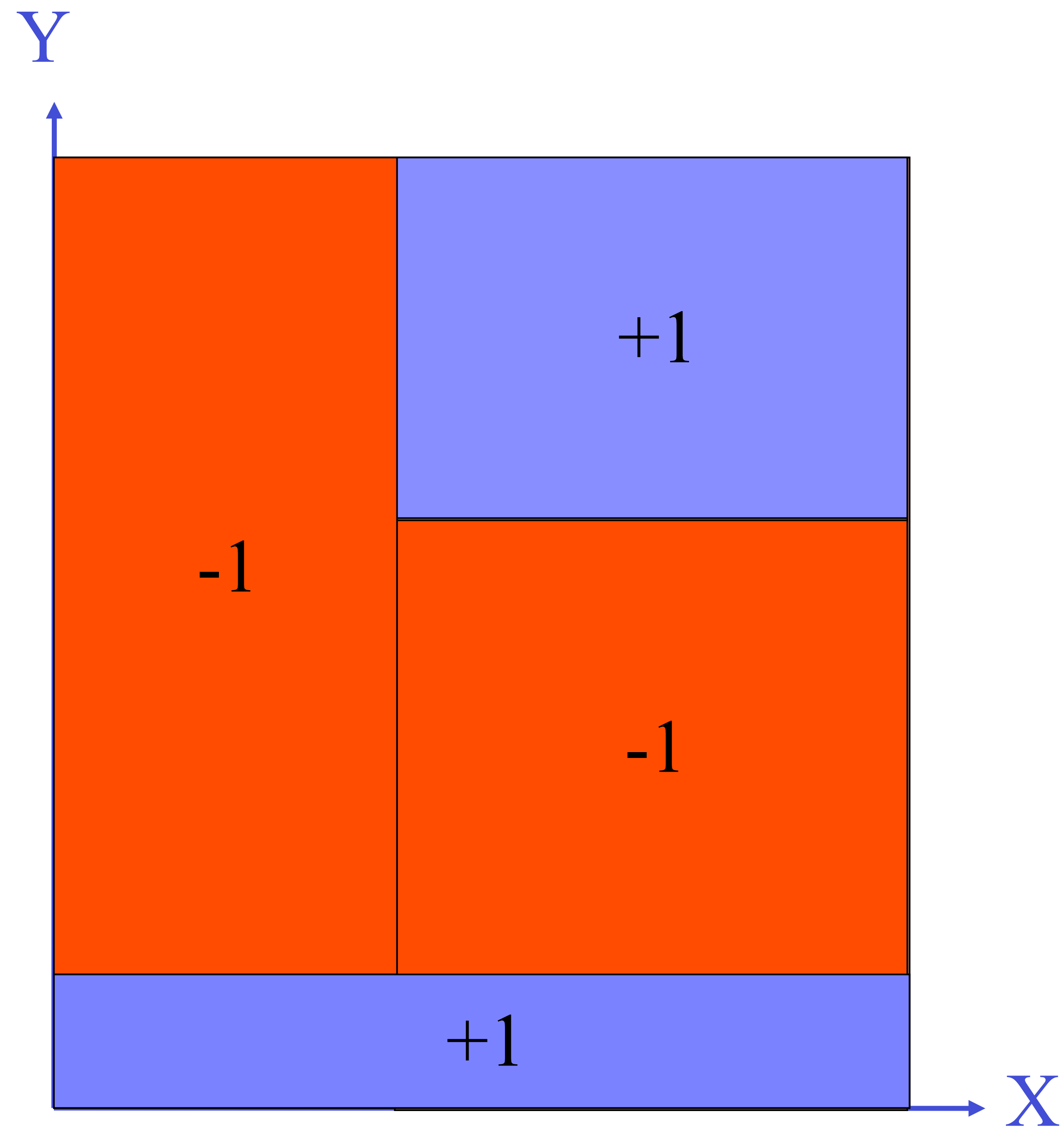
Decision tree as a sum





An alternating decision tree

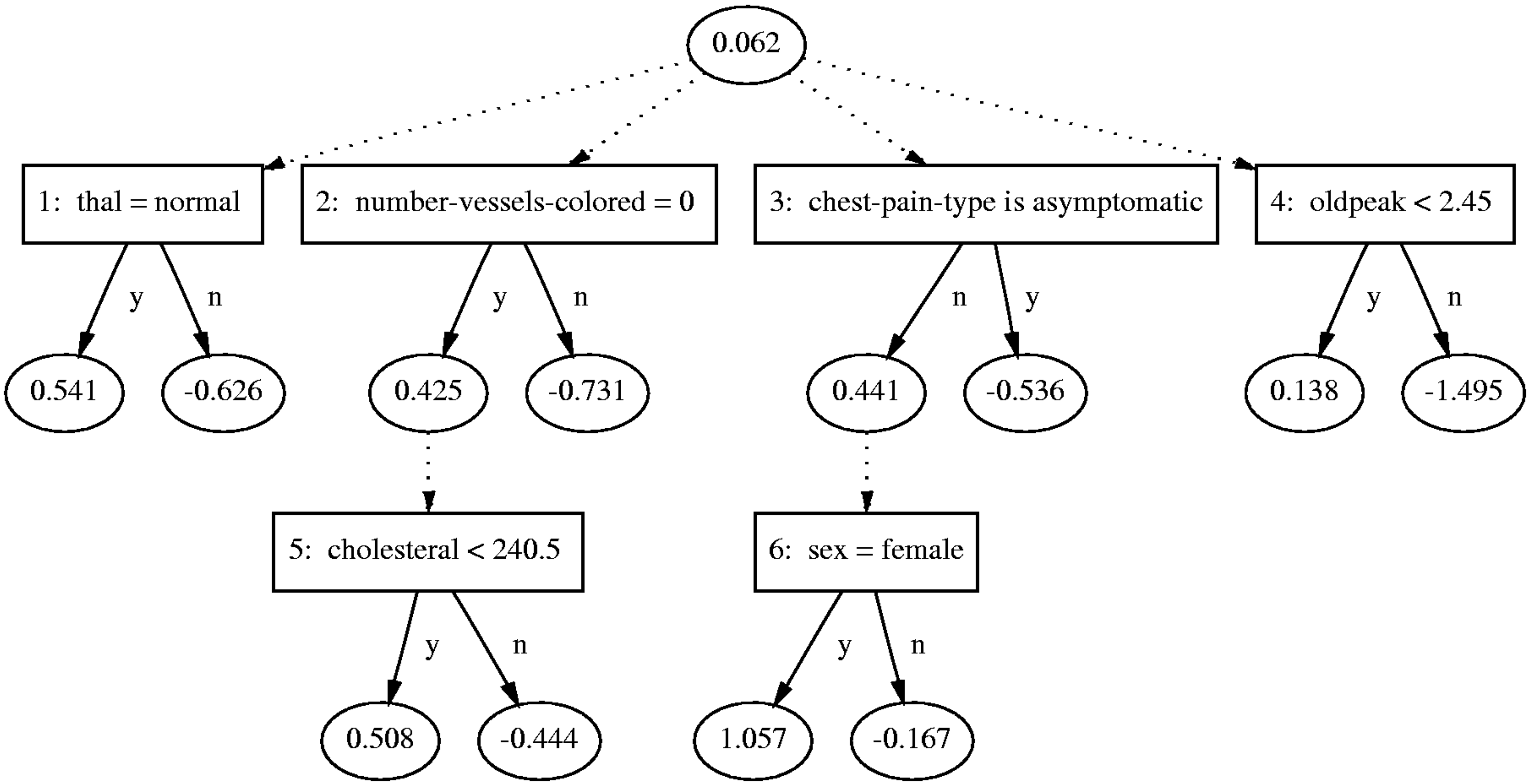




Example: Medical Diagnostics

- `Cleve` dataset from UC Irvine database.
- Heart disease diagnostics (+1=healthy,-1=sick)
- 13 features from tests (real valued and discrete).
- 303 instances.

Adtree for Cleveland heart-disease diagnostics problem



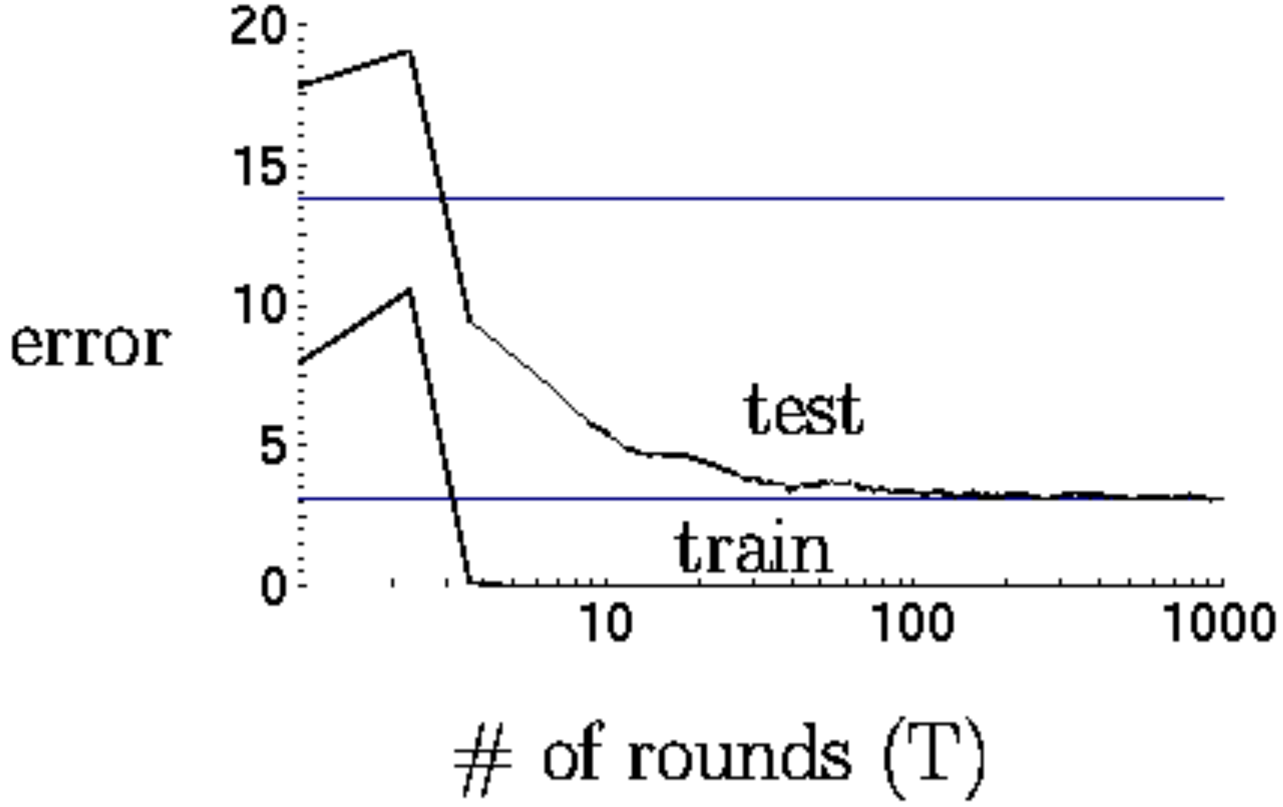
Cross-validated accuracy

Learning algorithm	Number of splits	Average test error	Test error variance
ADtree	6	17.0%	0.6%
C5.0	27	27.2%	0.5%
C5.0 + boosting	446	20.2%	0.5%
Boost Stumps	16	16.5%	0.8%

Boosting and over-fitting

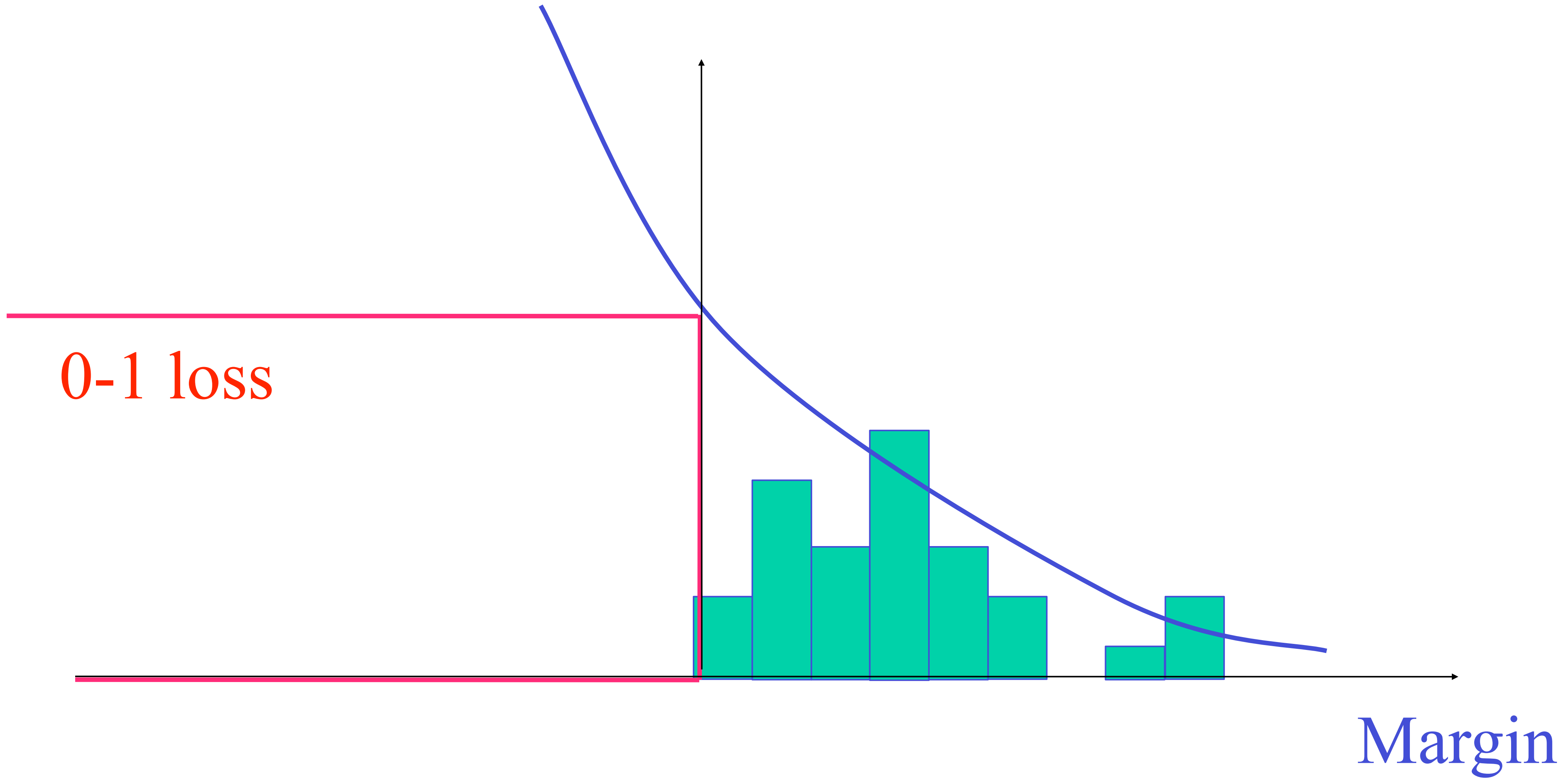
Curious phenomenon

Boosting decision trees

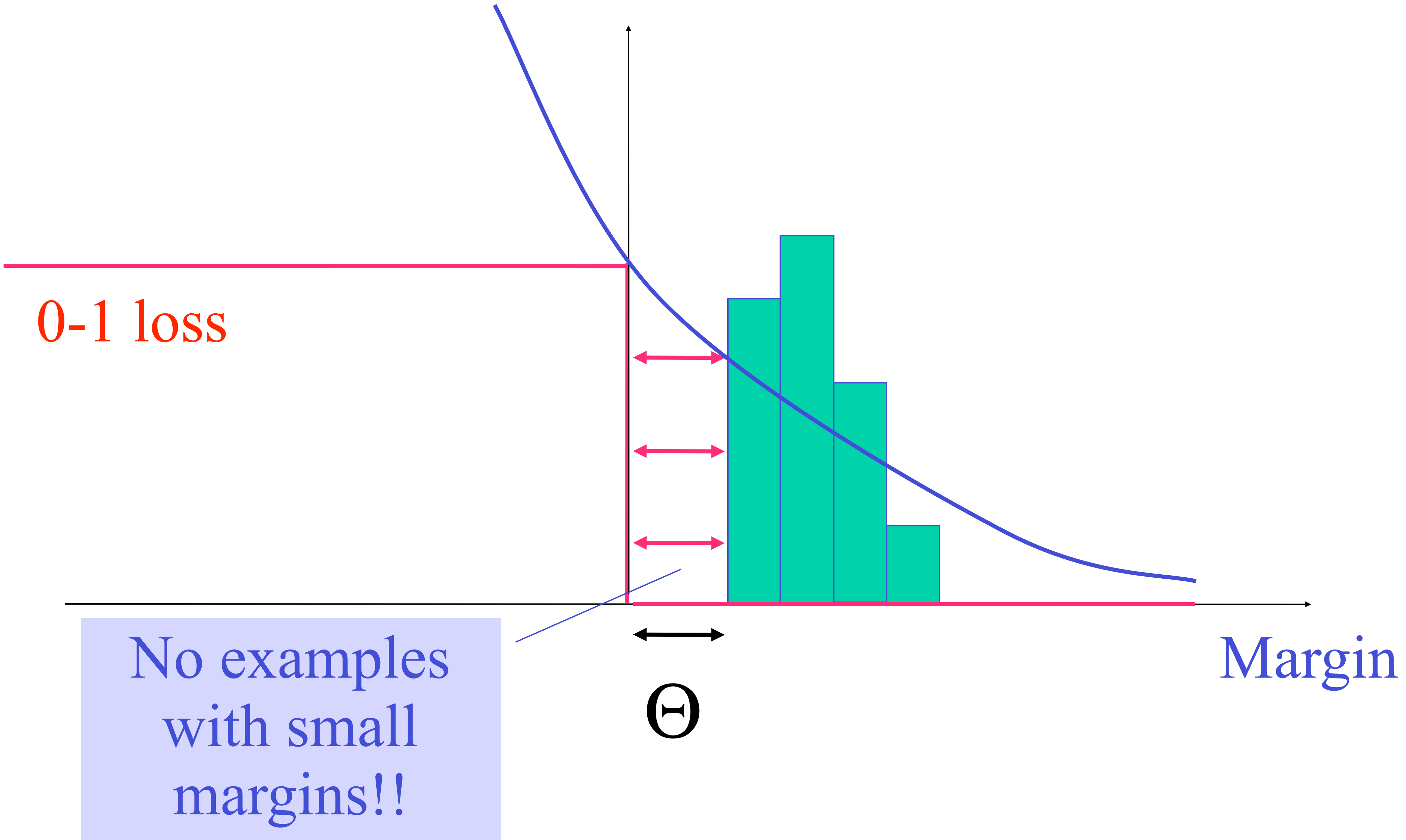


Using <10,000 training examples we fit >2,000,000 parameters

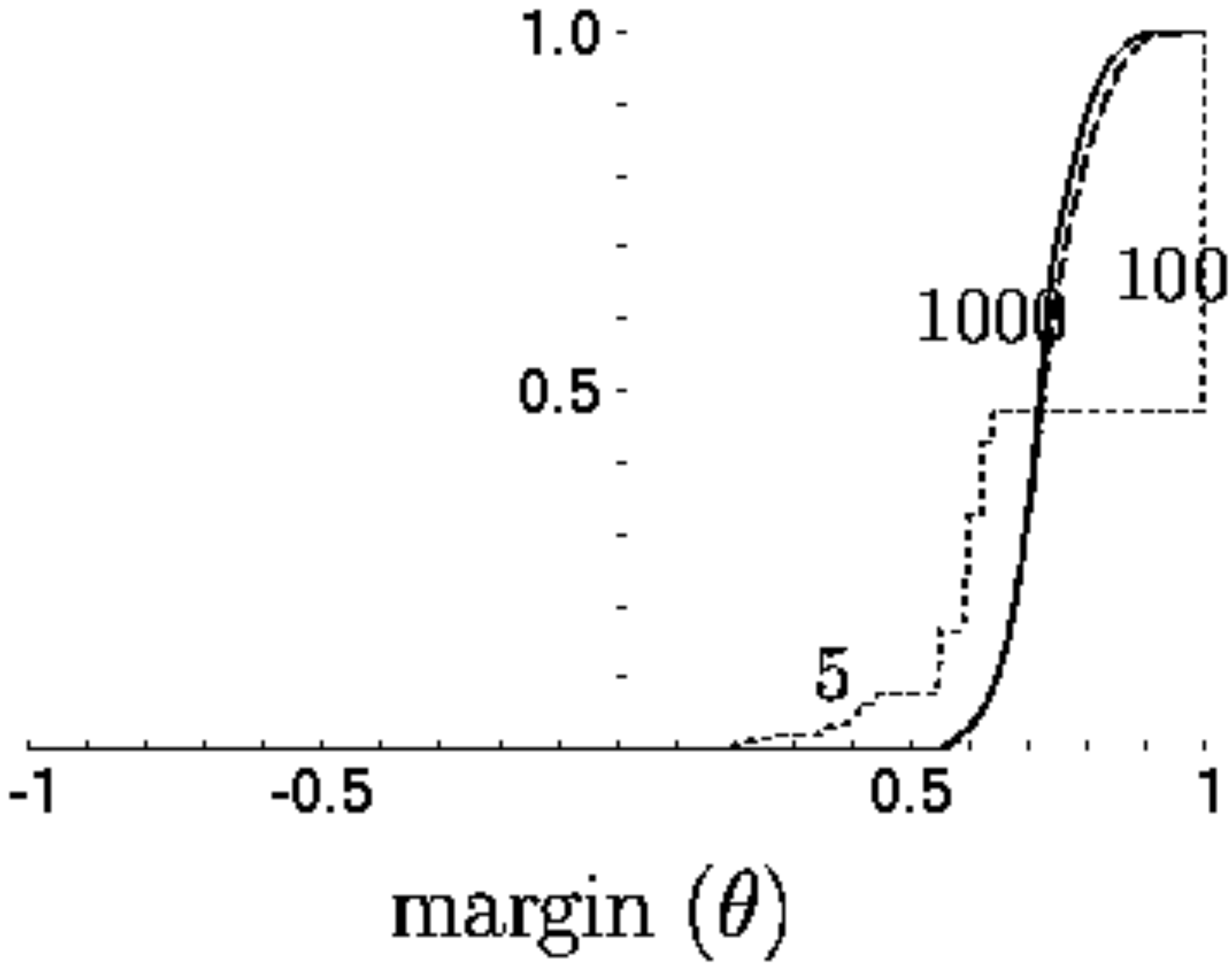
Explanation using margins



Explanation using margins



Experimental Evidence



Theorem

Schapire, Freund, Bartlett & Lee
Annals of stat. 98

For any convex combination and any threshold $\forall f \in C, \forall \theta > 0:$

$$\underbrace{P_{(x,y)\sim D}\left[\mathrm{sign}(f(x)) \neq y\right]}_{\text{Probability of mistake}} \leq \underbrace{P_{(x,y)\sim S}\left[\mathit{margin}_f(x,y) \leq \theta\right]}_{\text{Fraction of training example with small margin}}$$

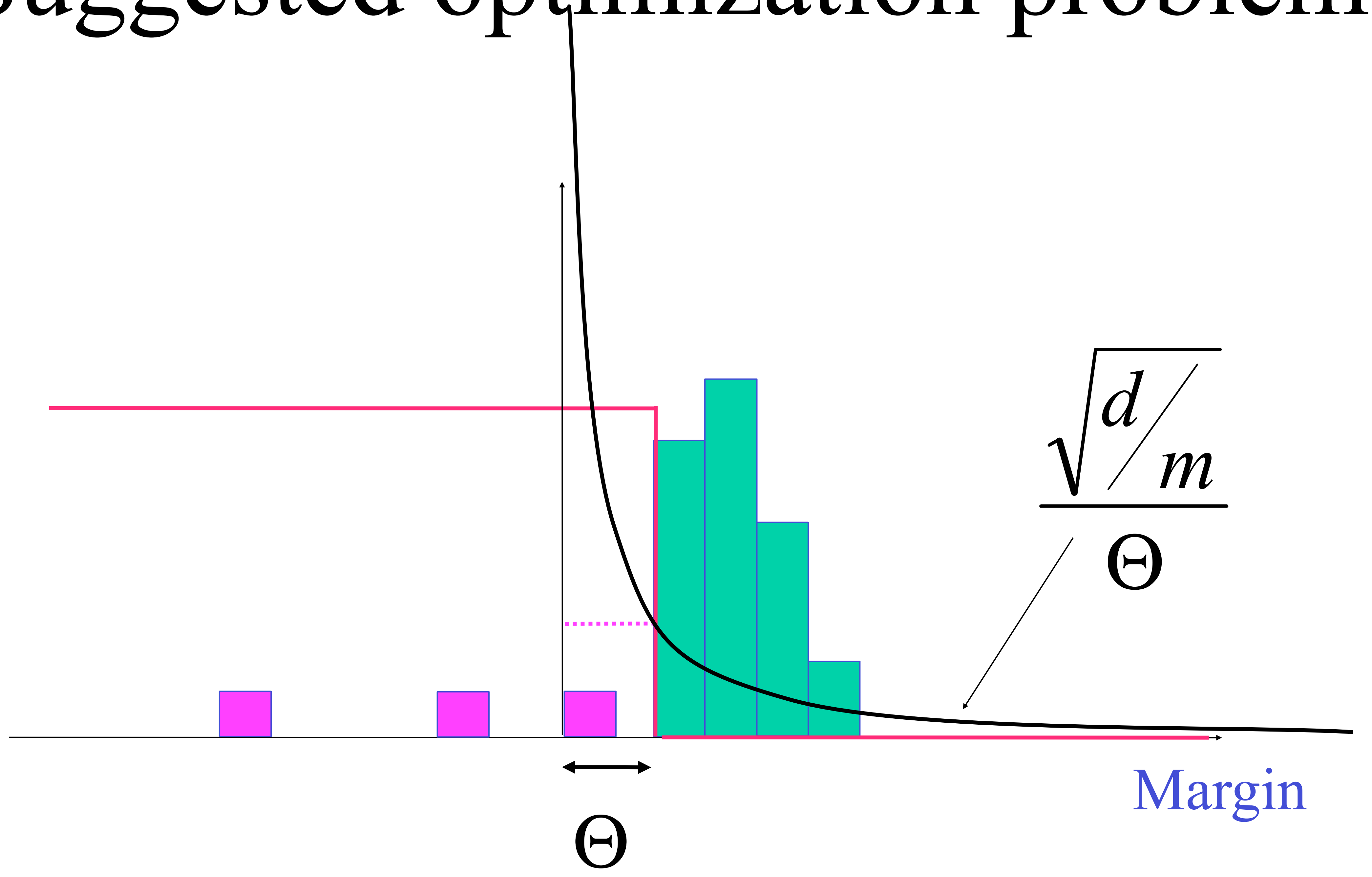
Size of training sample

$$+\ \tilde{O}\left(\frac{\sqrt{d/m}}{\theta}\right)$$

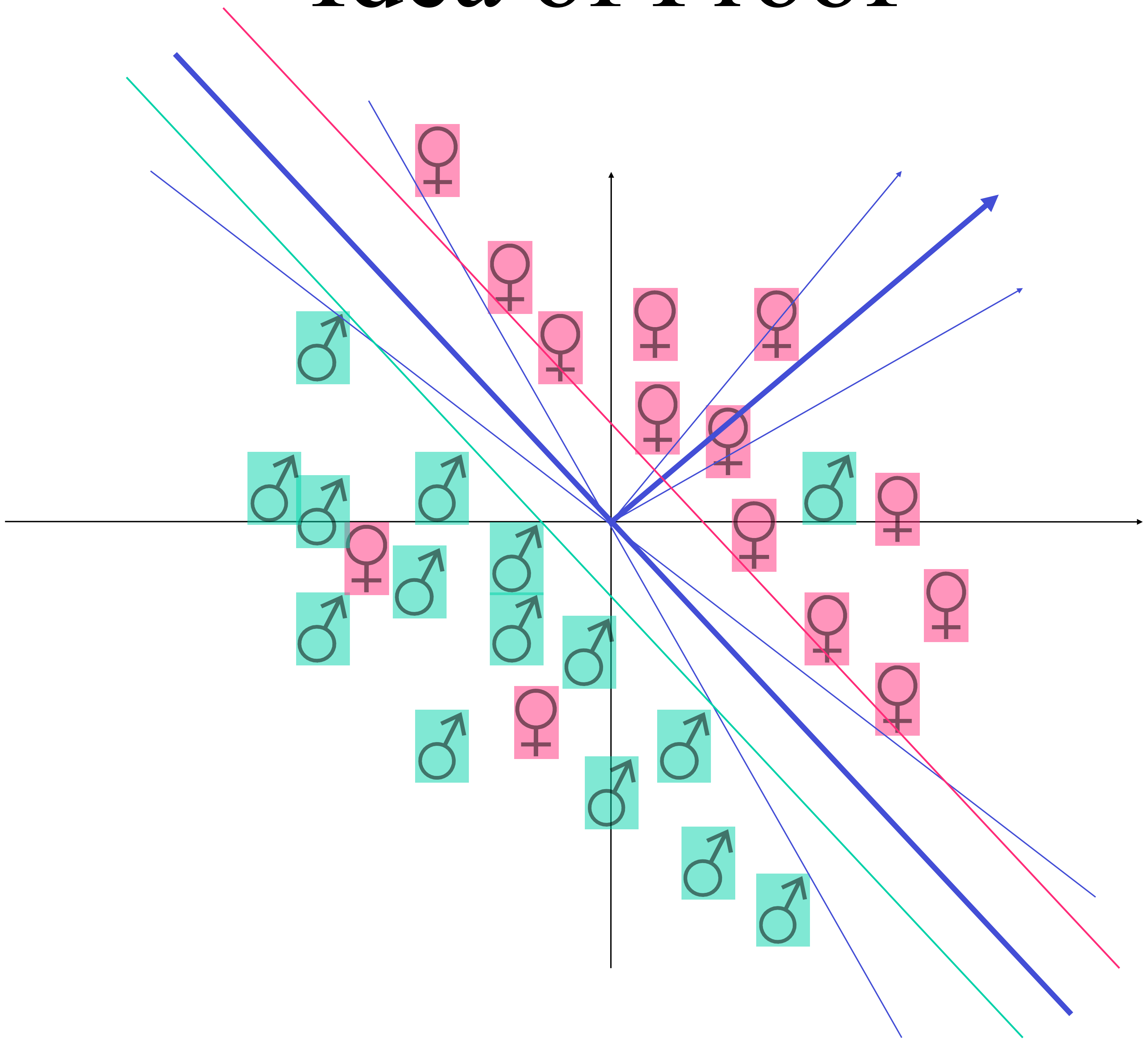
VC dimension of weak rules

No dependence on number of weak rules that are combined!!!

Suggested optimization problem



Idea of Proof

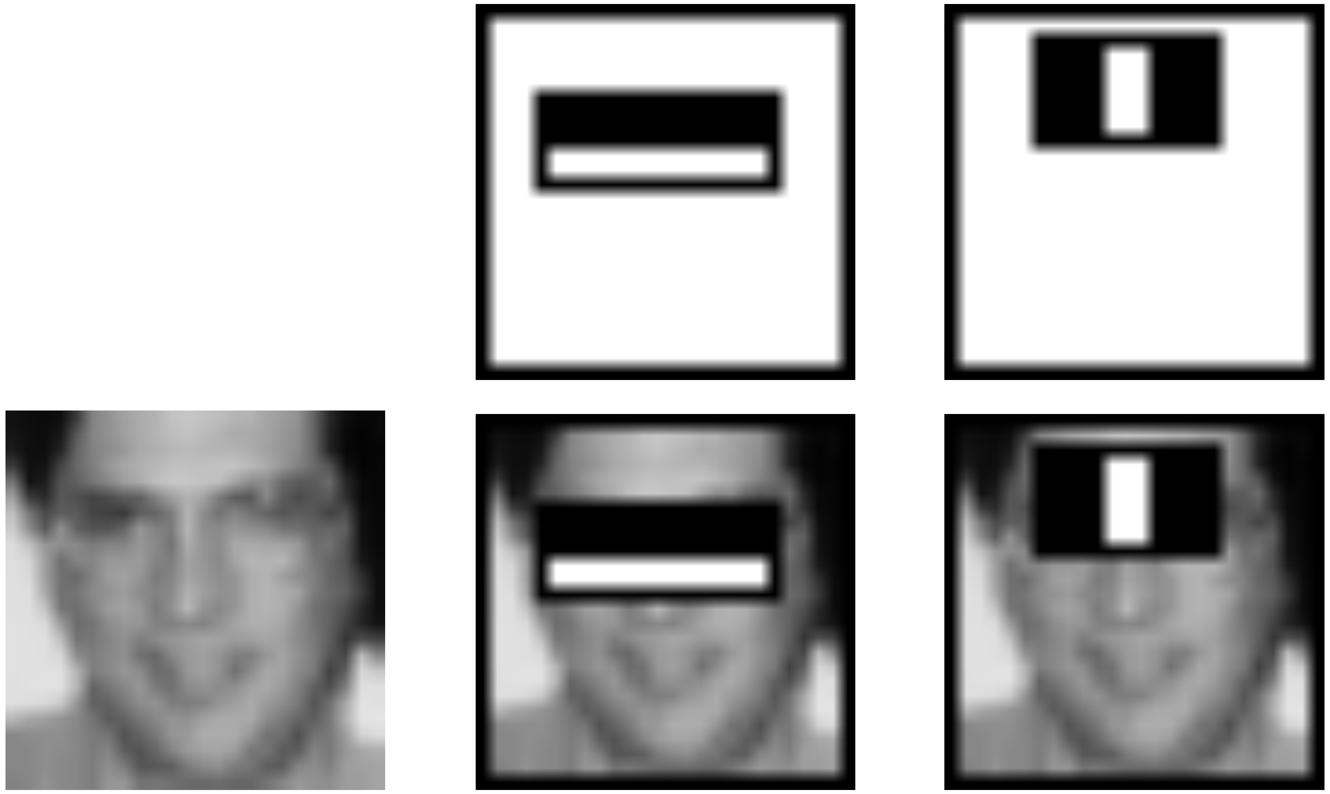


Applications

Application: Detecting Faces

[Viola & Jones]

- problem: find faces in photograph or movie
- weak classifiers: detect light/dark rectangles in image



- many clever tricks to make extremely fast and accurate

Viola and Jones (~1996)

Fundamental Perspectives

- game theory
- loss minimization
- an information-geometric view

Just a Game

- can view boosting as a game, a formal interaction between booster and weak learner
- on each round t:
  - booster chooses distribution D_t
  - weak learner responds with weak classifier h_t
- game theory: studies interactions between all sorts of "players"

Games

- game defined by matrix $\mathbf{M}$:

	Rock	Paper	Scissors
Rock	$1/2$	1	0
Paper	0	$1/2$	1
Scissors	1	0	$1/2$

- row player ("Mindy") chooses row i
- column player ("Max") chooses column j (simultaneously)
- Mindy's goal: minimize her loss $\mathbf{M}(i, j)$
- assume (wlog) all entries in $[0, 1]$

Randomized Play

- usually allow randomized play:
  - Mindy chooses distribution $\mathbf{P}$ over rows
  - Max chooses distribution $\mathbf{Q}$ over columns (simultaneously)
- Mindy's (expected) loss

$$
\begin{aligned}
&= \sum_{i,j} \mathbf{P}(i)\mathbf{M}(i,j)\mathbf{Q}(j) \\
&= \mathbf{P}^\top \mathbf{M}\mathbf{Q} \equiv \mathbf{M}(\mathbf{P},\mathbf{Q})
\end{aligned}
$$

- i, j = "pure" strategies
- $\mathbf{P}$, $\mathbf{Q}$ = "mixed" strategies
- $m = \#$ rows of $\mathbf{M}$
- also write $\mathbf{M}(i,\mathbf{Q})$ and $\mathbf{M}(\mathbf{P},j)$ when one side plays pure and other plays mixed

Sequential Play

- say Mindy plays before Max
- if Mindy chooses $\mathbf{P}$ then Max will pick $\mathbf{Q}$ to maximize $\mathbf{M}(\mathbf{P},\mathbf{Q})$ $\Rightarrow$ loss will be

$$L(\mathbf{P}) \equiv \max_{\mathbf{Q}} \mathbf{M}(\mathbf{P},\mathbf{Q})$$

- so Mindy should pick $\mathbf{P}$ to minimize $L(\mathbf{P})$ $\Rightarrow$ loss will be

$$\min_{\mathbf{P}} L(\mathbf{P}) = \min_{\mathbf{P}} \max_{\mathbf{Q}} \mathbf{M}(\mathbf{P},\mathbf{Q})$$

- similarly, if Max plays first, loss will be

$$\max_{\mathbf{Q}} \min_{\mathbf{P}} \mathbf{M}(\mathbf{P},\mathbf{Q})$$

Minmax Theorem

- playing second (with knowledge of other player's move) cannot be worse than playing first, so:

$$\underbrace{\min_{\mathbf{P}} \max_{\mathbf{Q}} \mathbf{M}(\mathbf{P}, \mathbf{Q})}_{\text{Mindy plays first}} \geq \underbrace{\max_{\mathbf{Q}} \min_{\mathbf{P}} \mathbf{M}(\mathbf{P}, \mathbf{Q})}_{\text{Mindy plays second}}$$

- von Neumann's minmax theorem:

$$\min_{\mathbf{P}} \max_{\mathbf{Q}} \mathbf{M}(\mathbf{P}, \mathbf{Q}) = \max_{\mathbf{Q}} \min_{\mathbf{P}} \mathbf{M}(\mathbf{P}, \mathbf{Q})$$

- in words: no advantage to playing second

Optimal Play

- minmax theorem:

$$\min_{\mathbf{P}} \max_{\mathbf{Q}} \mathbf{M}(\mathbf{P}, \mathbf{Q}) = \max_{\mathbf{Q}} \min_{\mathbf{P}} \mathbf{M}(\mathbf{P}, \mathbf{Q}) = \text{value } v \text{ of game}$$

- optimal strategies:
  - $\mathbf{P}^* = \arg\min_{\mathbf{P}} \max_{\mathbf{Q}} \mathbf{M}(\mathbf{P}, \mathbf{Q})$ = minmax strategy
  - $\mathbf{Q}^* = \arg\max_{\mathbf{Q}} \min_{\mathbf{P}} \mathbf{M}(\mathbf{P}, \mathbf{Q})$ = maxmin strategy
- in words:
  - Mindy's minmax strategy $\mathbf{P}^*$ guarantees loss $\leq v$ (regardless of Max's play)
  - optimal because Max has maxmin strategy $\mathbf{Q}^*$ that can force loss $\geq v$ (regardless of Mindy's play)
- e.g.: in RPS, $\mathbf{P}^* = \mathbf{Q}^* =$ uniform
- solving game = finding minmax/maxmin strategies

Weaknesses of Classical Theory

- seems to fully answer how to play games — just compute minmax strategy (e.g., using linear programming)
- weaknesses:
  - game **M** may be unknown
  - game **M** may be extremely large
  - opponent may not be fully adversarial
    - may be possible to do better than value v
    - e.g.:

      Lisa (thinks):

      Poor predictable Bart, always takes Rock.

      Bart (thinks):

      Good old Rock, nothing beats that.

Repeated Play

- if only playing once, hopeless to overcome ignorance of game $\mathbf{M}$ or opponent
- but if game played repeatedly, may be possible to learn to play well
- goal: play (almost) as well as if knew game and how opponent would play ahead of time

Repeated Play (cont.)

- $\mathbf{M}$ unknown
- for $t = 1, \ldots, T$:
  - Mindy chooses $\mathbf{P}_t$
  - Max chooses $\mathbf{Q}_t$ (possibly depending on $\mathbf{P}_t$)
  - Mindy's loss $= \mathbf{M}(\mathbf{P}_t, \mathbf{Q}_t)$
  - Mindy observes loss $\mathbf{M}(i, \mathbf{Q}_t)$ of each pure strategy i
- want:

$$\underbrace{\frac{1}{T}\sum_{t=1}^{T}\mathbf{M}(\mathbf{P}_t,\mathbf{Q}_t)}_{\text{actual average loss}} \le \underbrace{\min_{\mathbf{P}}\frac{1}{T}\sum_{t=1}^{T}\mathbf{M}(\mathbf{P},\mathbf{Q}_t)}_{\text{best loss (in hindsight)}} + [\text{“small amount”}]$$

Multiplicative-weights Algorithm (MW)

- choose $\eta > 0$
- initialize: $\mathbf{P}_1 =$ uniform
- on round t:

$$\mathbf{P}_{t+1}(i) = \frac{\mathbf{P}_t(i)\ \exp\left(-\eta\, \mathbf{M}(i, \mathbf{Q}_t)\right)}{\text{normalization}}$$

- idea: decrease weight of strategies suffering the most loss
- directly generalizes [Littlestone & Warmuth]
- other algorithms:
  - [Hannan'57]
  - [Blackwell'56]
  - [Foster & Vohra]
  - [Fudenberg & Levine]

  ⋮

Analysis

- Theorem: can choose η so that, for any game $\mathbf{M}$ with m rows, and any opponent,

$$\underbrace{\frac{1}{T}\sum_{t=1}^{T}\mathbf{M}(\mathbf{P}_t,\mathbf{Q}_t)}_{\text{actual average loss}} \leq \underbrace{\min_{\mathbf{P}}\frac{1}{T}\sum_{t=1}^{T}\mathbf{M}(\mathbf{P},\mathbf{Q}_t)}_{\text{best average loss } (\leq v)} + \Delta_T$$

where $\Delta_T = O\left(\sqrt{\frac{\ln m}{T}}\right) \to 0$

- regret Δ_T is:
  - logarithmic in # rows m
  - independent of # columns
- therefore, can use when working with very large games

Solving a Game

- suppose game $\mathbf{M}$ played repeatedly
  - Mindy plays using MW
  - on round t, Max chooses best response:

$$\mathbf{Q}_t = \arg\max_{\mathbf{Q}} \mathbf{M}(\mathbf{P}_t, \mathbf{Q})$$

- let

$$\overline{\mathbf{P}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{P}_t, \quad \overline{\mathbf{Q}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{Q}_t$$

- can prove that $\overline{\mathbf{P}}$ and $\overline{\mathbf{Q}}$ are Δ_T-approximate minmax and maxmin strategies:

$$\max_{\mathbf{Q}} \mathbf{M}(\overline{\mathbf{P}}, \mathbf{Q}) \leq v + \Delta_T$$

  and

$$\min_{\mathbf{P}} \mathbf{M}(\mathbf{P}, \overline{\mathbf{Q}}) \geq v - \Delta_T$$

Fundamental Perspectives

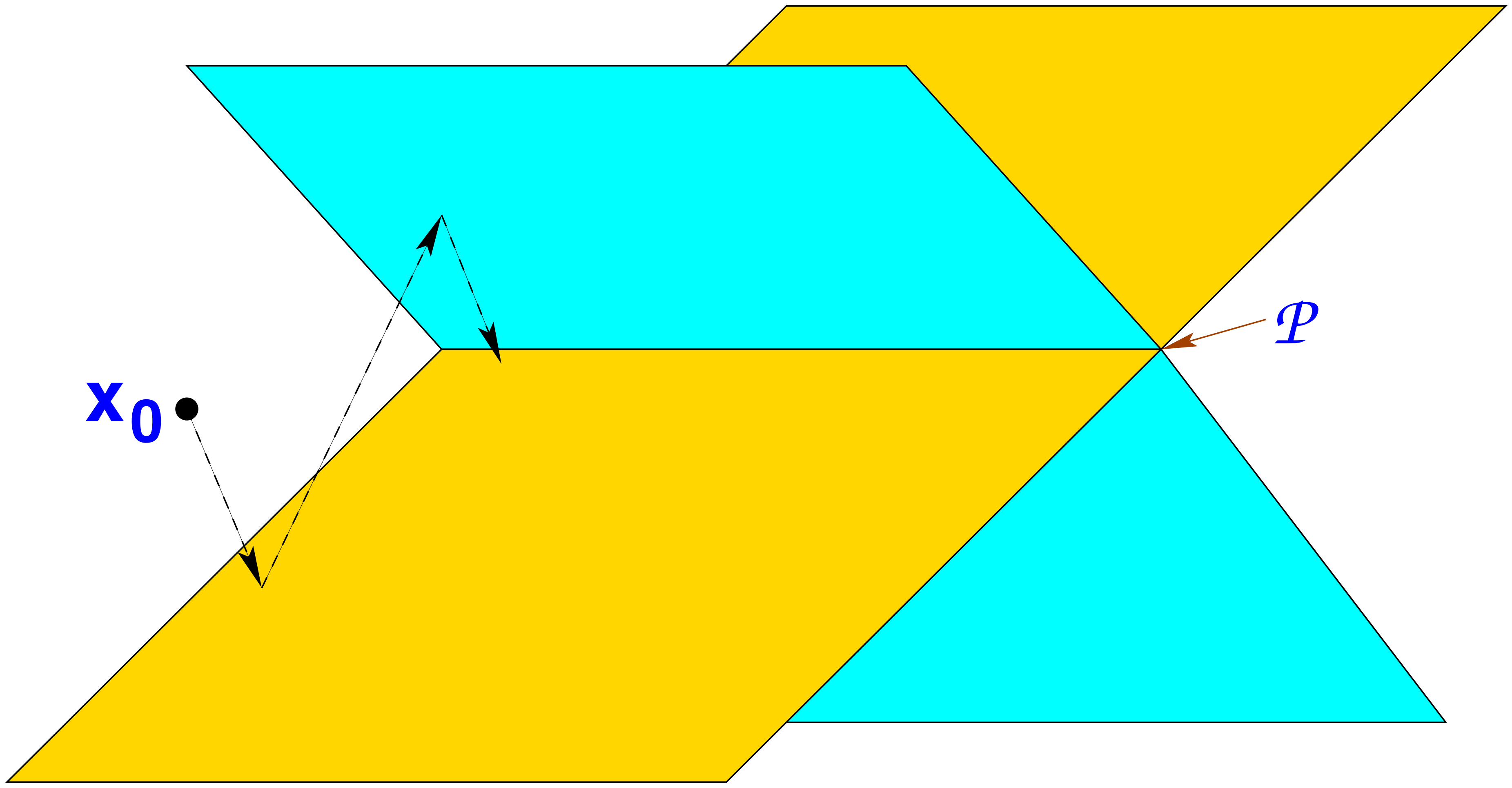
- game theory
- loss minimization
- an information-geometric view

A Dual Information-Geometric Perspective

- loss minimization focuses on function computed by AdaBoost (i.e., weights on weak classifiers)
- dual view: instead focus on distributions D_t (i.e., weights on examples)
- dual perspective combines geometry and information theory
- exposes underlying mathematical structure
- basis for proving convergence

An Iterative-Projection Algorithm

- say want to find point closest to $\mathbf{x}_0$ in set $\mathcal{P} = \{$ intersection of N hyperplanes $\}$
- algorithm: [Bregman; Censor & Zenios]
  - start at $\mathbf{x}_0$
  - repeat: pick a hyperplane and project onto it



- if $\mathcal{P} \neq \emptyset$, under general conditions, will converge correctly

AdaBoost is an Iterative-Projection Algorithm

[Kivinen & Warmuth]

- points = distributions D_t over training examples
- distance = relative entropy:

$$\mathrm{RE}\left(P \parallel Q\right) = \sum_i P(i) \ln\left(\frac{P(i)}{Q(i)}\right)$$

- reference point $\mathbf{x}_0 =$ uniform distribution
- hyperplanes defined by all possible weak classifiers g_j:

$$\sum_i D(i) y_i g_j(x_i) = 0 \Leftrightarrow \Pr_{i \sim D}\left[g_j(x_i) \neq y_i\right] = \tfrac{1}{2}$$

  - intuition: looking for "hardest" distribution

AdaBoost as Iterative Projection (cont.)

- algorithm:
  - start at $D_1 =$ uniform
  - for $t = 1, 2, \ldots$:
    - pick hyperplane/weak classifier $h_t \leftrightarrow g_j$
    - $D_{t+1} =$ (entropy) projection of D_t onto hyperplane

$$= \arg \min_{D : \sum_i D(i) y_i g_j(x_i) = 0} \mathrm{RE}(D \parallel D_t)$$

- claim: equivalent to AdaBoost
- further: choosing h_t with minimum error $\equiv$ choosing farthest hyperplane

Boosting as Maximum Entropy

- corresponding optimization problem:

$$\min_{D \in \mathcal{P}} \mathrm{RE}(D \parallel \text{uniform}) \leftrightarrow \max_{D \in \mathcal{P}} \text{entropy}(D)$$

- where

$$\begin{aligned} \mathcal{P} &= \text{feasible set} \\ &= \left\{ D : \sum_i D(i) y_i g_j(x_i) = 0 \ \ \forall j \right\} \end{aligned}$$

- $\mathcal{P} \neq \emptyset \Leftrightarrow$ weak learning assumption does not hold
  - in this case, $D_t \rightarrow$ (unique) solution
- if weak learning assumption does hold then
  - $\mathcal{P} = \emptyset$
  - D_t can never converge
  - dynamics are fascinating but unclear in this case

Unifying the Two Cases

[Schapire Collins & Singer]

- two distinct cases:
  - weak learning assumption holds
    - $\mathcal{P} = \emptyset$
    - dynamics unclear
  - weak learning assumption does not hold
    - $\mathcal{P} \neq \emptyset$
    - can prove convergence of D_t's
- to unify: work instead with unnormalized versions of D_t's
  - standard AdaBoost: $D_{t+1}(i) = \dfrac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{\text{normalization}}$
  - instead:

$$d_{t+1}(i) = d_t(i) \exp(-\alpha_t y_i h_t(x_i))$$

$$D_{t+1}(i) = \frac{d_{t+1}(i)}{\text{normalization}}$$

  - algorithm is unchanged

Reformulating AdaBoost as Iterative Projection

- points $=$ nonnegative vectors $\mathbf{d}_t$
- distance $=$ unnormalized relative entropy:

$$\mathrm{RE}\left(\mathbf{p} \;\|\; \mathbf{q}\right) = \sum_i \left[p(i) \ln\left(\frac{p(i)}{q(i)}\right) + q(i) - p(i) \right]$$

- reference point $\mathbf{x}_0 = \mathbf{1}$ (all 1's vector)
- hyperplanes defined by weak classifiers g_j:

$$\sum_i d(i) y_i g_j(x_i) = 0$$

- resulting iterative-projection algorithm is again equivalent to AdaBoost

Reformulated Optimization Problem

- optimization problem:

$$\min_{\mathbf{d} \in \mathcal{P}} \mathrm{RE}\left(\mathbf{d} \;\middle\|\; \mathbf{1}\right)$$

- where

$$\mathcal{P} = \left\{ \mathbf{d} : \sum_i d(i) y_i g_j(x_i) = 0 \;\; \forall j \right\}$$

- note: feasible set $\mathcal{P}$ never empty (since $\mathbf{0} \in \mathcal{P}$)

Exponential Loss as Entropy Optimization

- all vectors $\mathbf{d}_t$ created by AdaBoost have form:

$$d(i) = \exp\left(-y_i \sum_j \lambda_j g_j(x_i)\right)$$

- let $\mathcal{Q} = \{$ all vectors $\mathbf{d}$ of this form $\}$
- can rewrite exponential loss:

$$\begin{aligned}
\inf_{\boldsymbol{\lambda}} \sum_i \exp\left(-y_i \sum_j \lambda_j g_j(x_i)\right) &= \inf_{\mathbf{d}\in\mathcal{Q}} \sum_i d(i) \\
&= \min_{\mathbf{d}\in\overline{\mathcal{Q}}} \sum_i d(i) \\
&= \min_{\mathbf{d}\in\overline{\mathcal{Q}}} \mathrm{RE}\left(\mathbf{0} \ \| \ \mathbf{d}\right)
\end{aligned}$$

- $\overline{\mathcal{Q}}$ = closure of $\mathcal{Q}$

Duality

[Della Pietra, Della Pietra & Lafferty]

- presented two optimization problems:
  - $\min_{\mathbf{d} \in \mathcal{P}} \mathrm{RE}(\mathbf{d} \parallel \mathbf{1})$
  - $\min_{\mathbf{d} \in \overline{\mathcal{Q}}} \mathrm{RE}(\mathbf{0} \parallel \mathbf{d})$
- which is AdaBoost solving? Both!
- problems have same solution
- moreover: solution given by unique point in $\mathcal{P} \cap \overline{\mathcal{Q}}$
- problems are convex duals of each other

Convergence of AdaBoost

- can use to prove AdaBoost converges to common solution of both problems:
  - can argue that $\mathbf{d}^* = \lim \mathbf{d}_t$ is in $\mathcal{P}$
  - vectors $\mathbf{d}_t$ are in $\mathcal{Q}$ always $\Rightarrow \mathbf{d}^* \in \overline{\mathcal{Q}}$

  $\therefore$ $\mathbf{d}^* \in \mathcal{P} \cap \overline{\mathcal{Q}}$

  $\therefore$ $\mathbf{d}^*$ solves both optimization problems
- so:
  - AdaBoost minimizes exponential loss
  - exactly characterizes limit of unnormalized "distributions"
  - likewise for normalized distributions when weak learning assumption does not hold
- also, provides additional link to logistic regression
  - only need slight change in optimization problem

[Schapire, Collins, Singer; Lebannon & Lafferty]

Conclusions

- from different perspectives, AdaBoost can be interpreted as:
  - a method for boosting the accuracy of a weak learner
  - a procedure for maximizing margins
  - an algorithm for playing repeated games
  - a numerical method for minimizing exponential loss
  - an iterative-projection algorithm based on an information-theoretic geometry
- none is entirely satisfactory by itself, but each useful in its own way
- taken together, create rich theoretical understanding
  - connect boosting to other learning problems and techniques
  - provide foundation for versatile set of methods with many extensions, variations and applications

References

Coming soon:

- Robert E. Schapire and Yoav Freund.
  Boosting: Foundations and Algorithms.
  MIT Press, 2012.

Survey articles:

- Ron Meir and Gunnar Rätsch.
  An Introduction to Boosting and Leveraging.
  In *Advanced Lectures on Machine Learning (LNAI2600)*, 2003.
  http://www.boosting.org/papers/MeiRae03.pdf
- Robert E. Schapire.
  The boosting approach to machine learning: An overview.
  In *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.
  http://www.cs.princeton.edu/∼schapire/boost.html