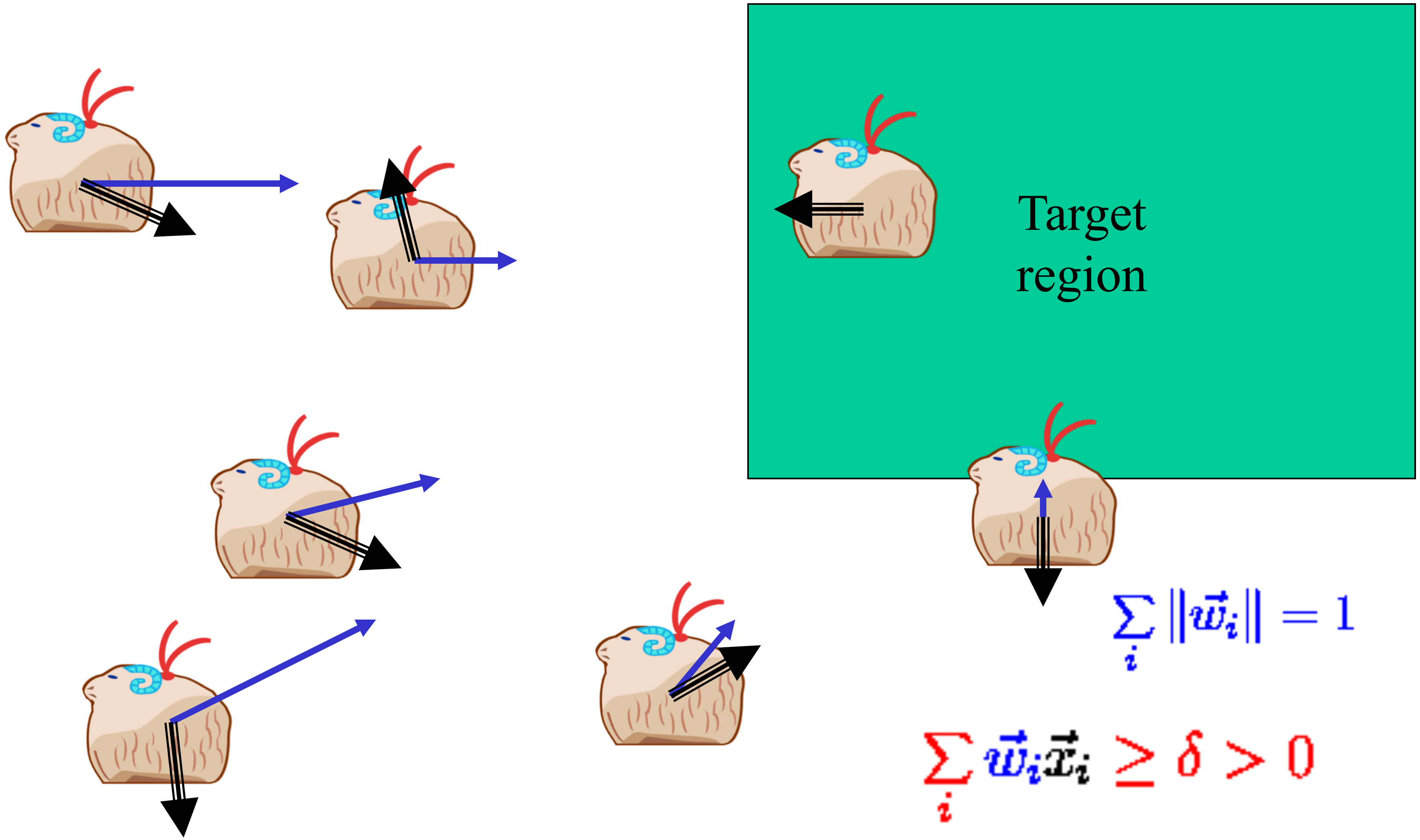
Learning Games and Brownian Motion

Yoav Freund

Center for Computational Learning Systems (CLASS)

Columbia University

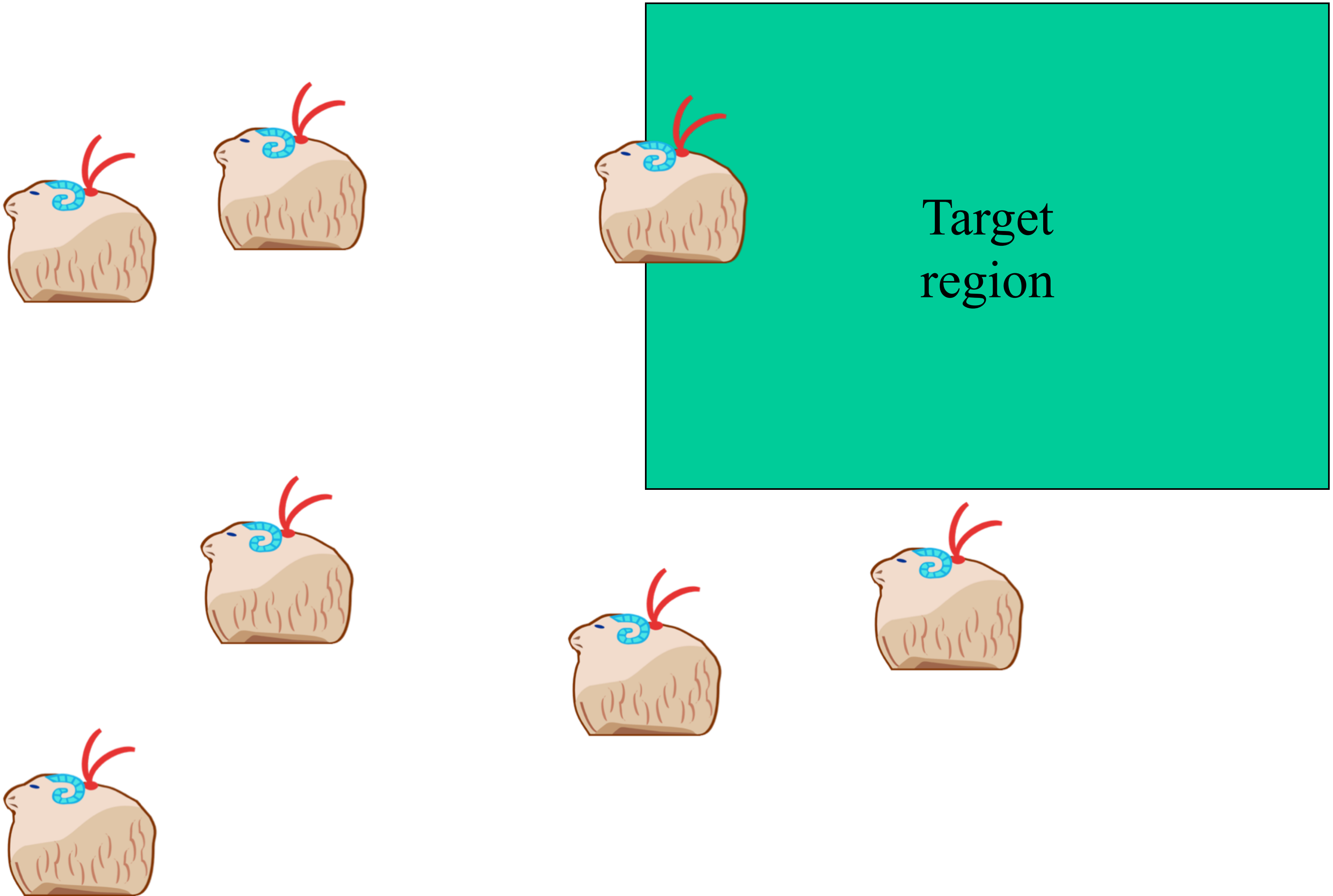
Drifting games (in 2d)



$$\sum_i \|\vec{w}_i\| = 1$$

$$\sum_i \vec{w}_i \vec{x}_i \geq \delta > 0$$

Drifting games (in 2d)



Target region

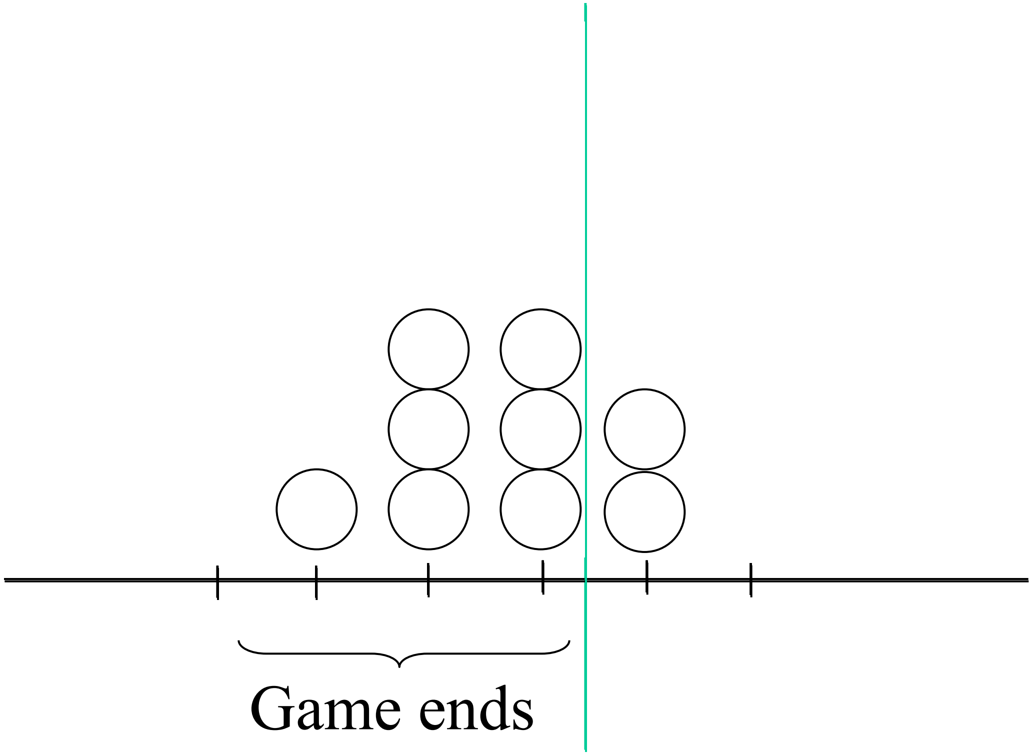
Plan of talk

- The one dimensional case
- Applications (in learning and elsewhere)
- Solution of the one-dimensional case
- Generalization to higher dimensions
- The continuous time limit (Brownian motion)
- Other applications and open problems

Simple 1d drifting games

- Chips (formerly sheep) start from origin.
- Goal of shepherd: move all chips above origin.
- Sheep strategist divided into Splitter and Chooser.
- GAME: repeat
  1) Shepherd set a weight for each chip, weights sum to 1.
  2) Splitter splits the chips into two sets.
  3) Chooser selects one of the two sets which is then moved one step up.

The 1d Chip game

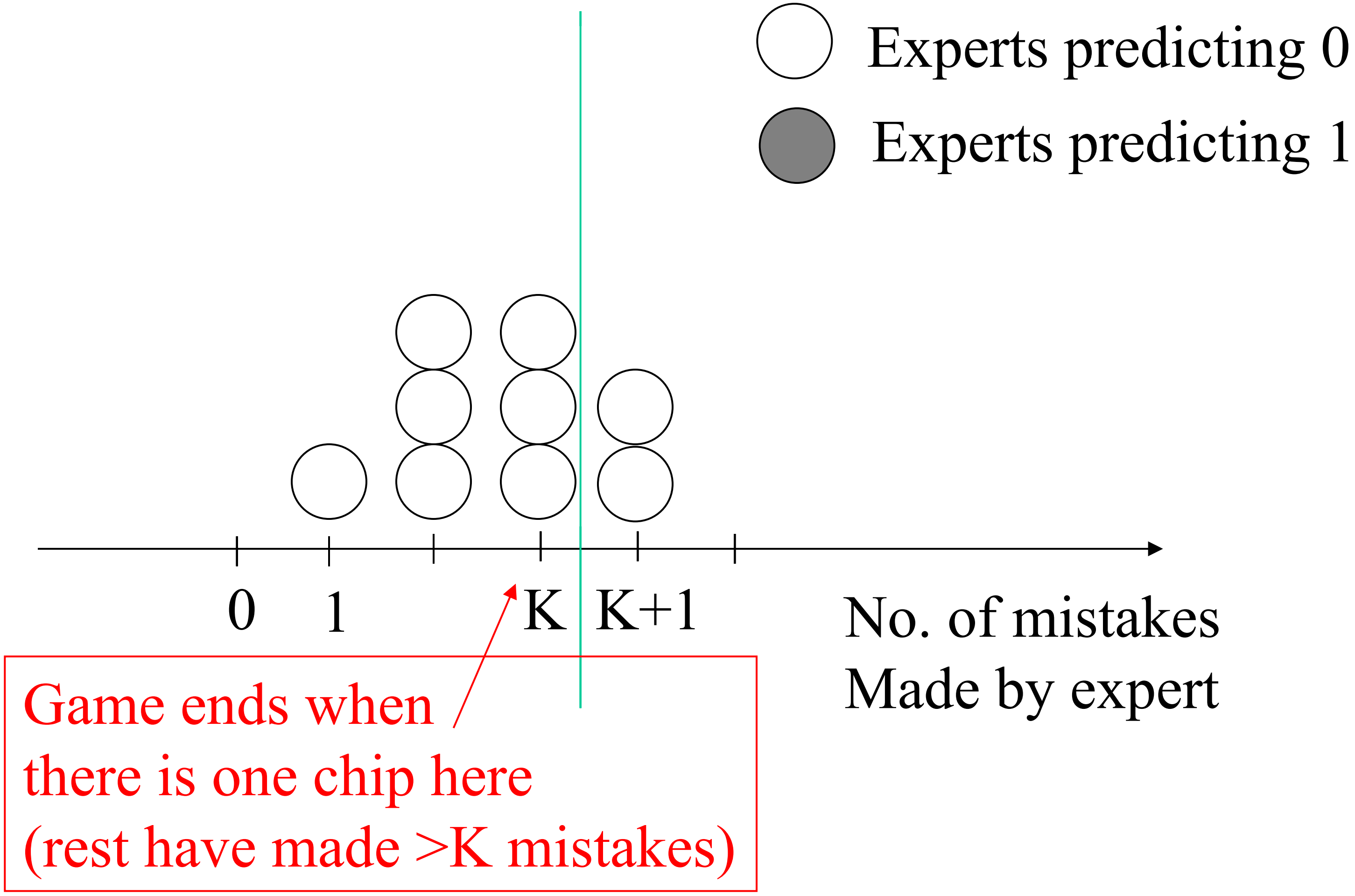


Game ends
when this
is empty

Combining expert advice

- Binary sequence: 1,0,0,1,1,0,?
- N experts make predictions:
- expert 1: 1,0,1,0,1,1,1
- expert 2: 0,0,0,1,1,0,1
- …
- expert N: 1,0,0,1,1,1,0
- Algorithm makes prediction: 1,0,1,1,0,1,1
- Assumption: there exists an expert which makes at most k mistakes
- Goal: make least mistakes under the assumption (no statistics!)
- Chip = expert, bin = number of mistakes made
- Game step = algorithm’s prediction is incorrect.

Chip game for expert advice



○ Experts predicting 0

● Experts predicting 1

No. of mistakes
Made by expert

Game ends when
there is one chip here
(rest have made >K mistakes)

Boosting

- A method for improving accuracy of learning algorithms for classification problems.
- Weak Learner: a learning algorithm generating rules that are only slightly better than random guessing.
- Basic idea: re-weight training examples to force weak learner into different parts of the space.
- Combine weak rules by a majority vote.

batch learning for binary classification

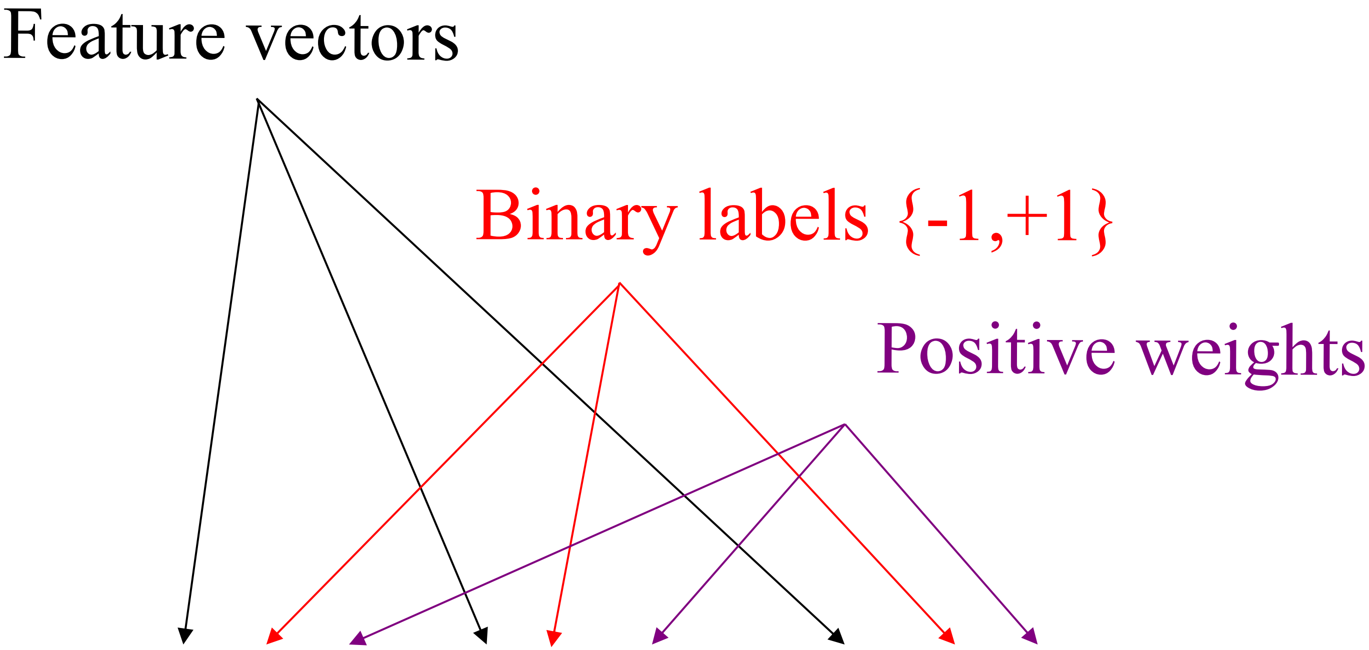
Data distribution:

Generalization error:

Training set:

Training error:

A weighted training set

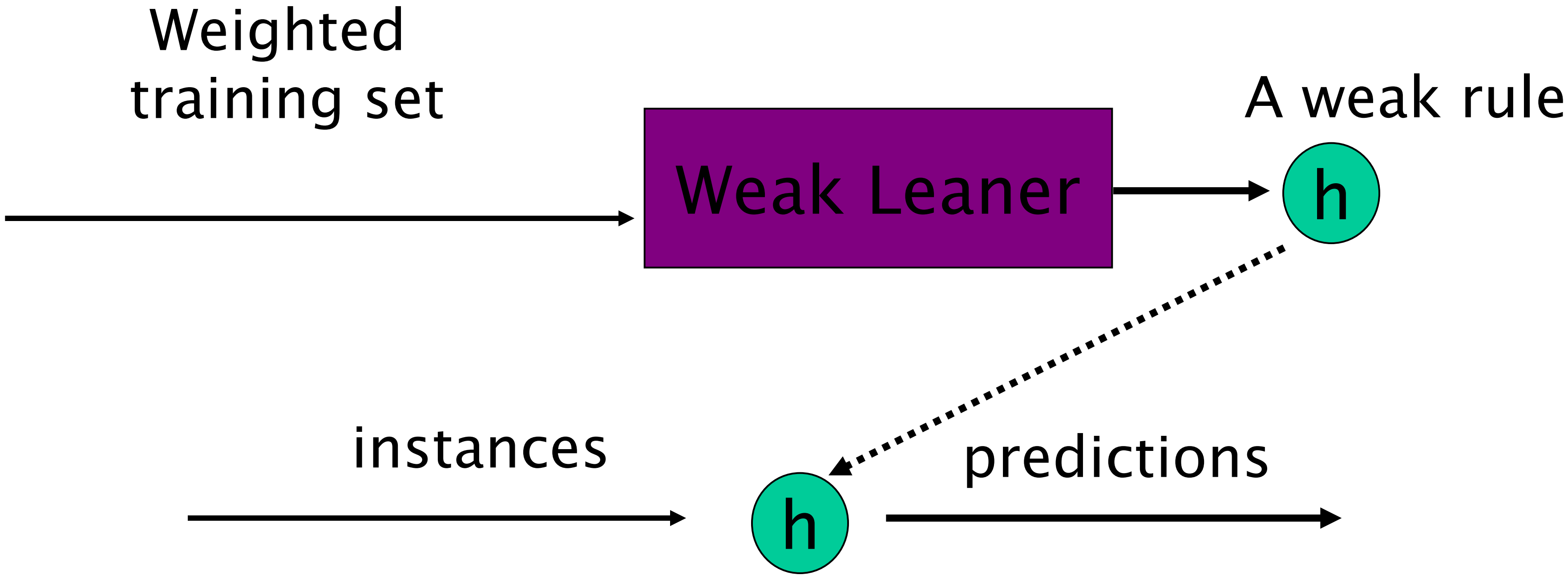


Feature vectors

Binary labels {-1,+1}

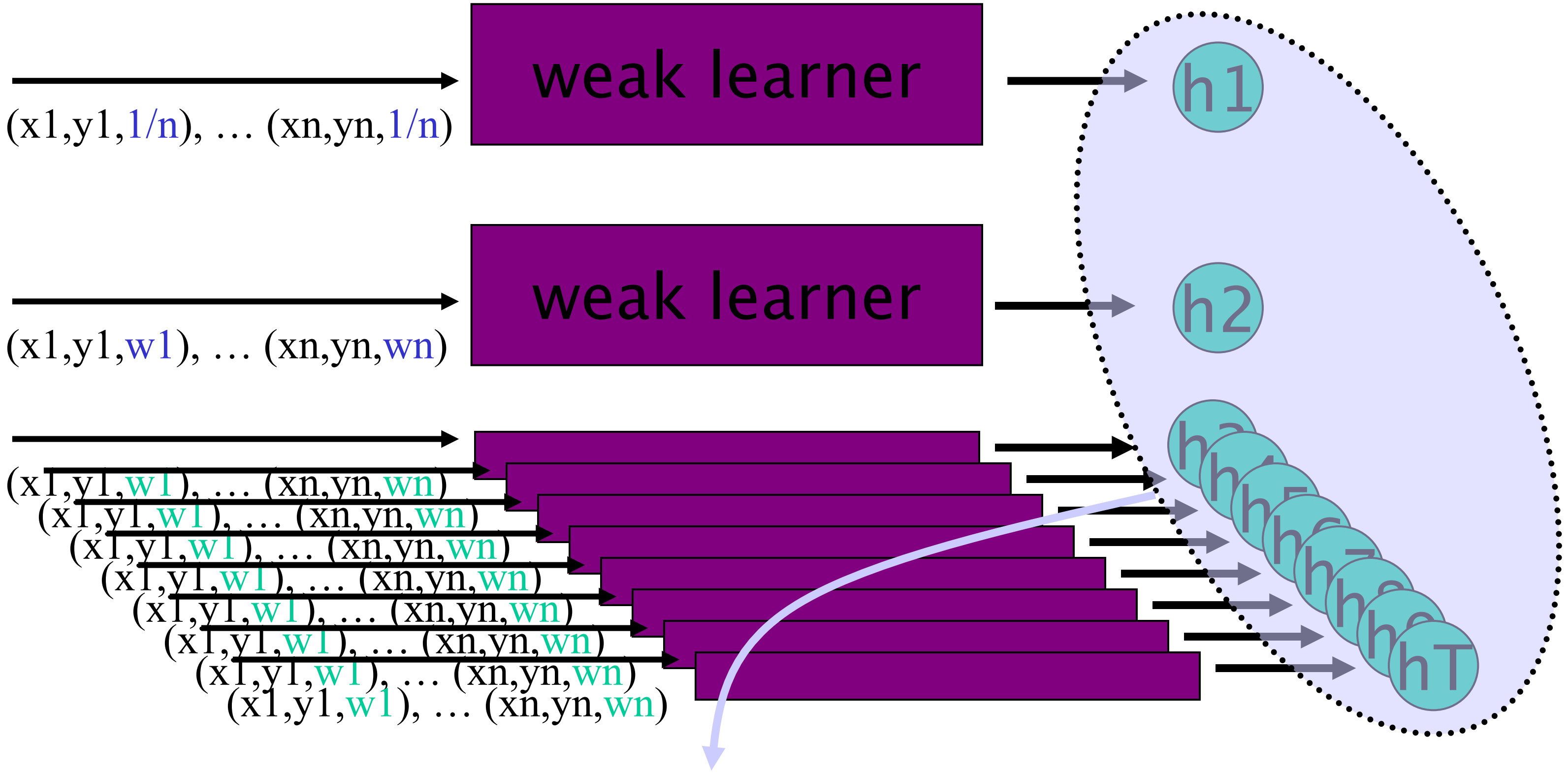
Positive weights

A weak learner



The weak requirement:

The boosting process

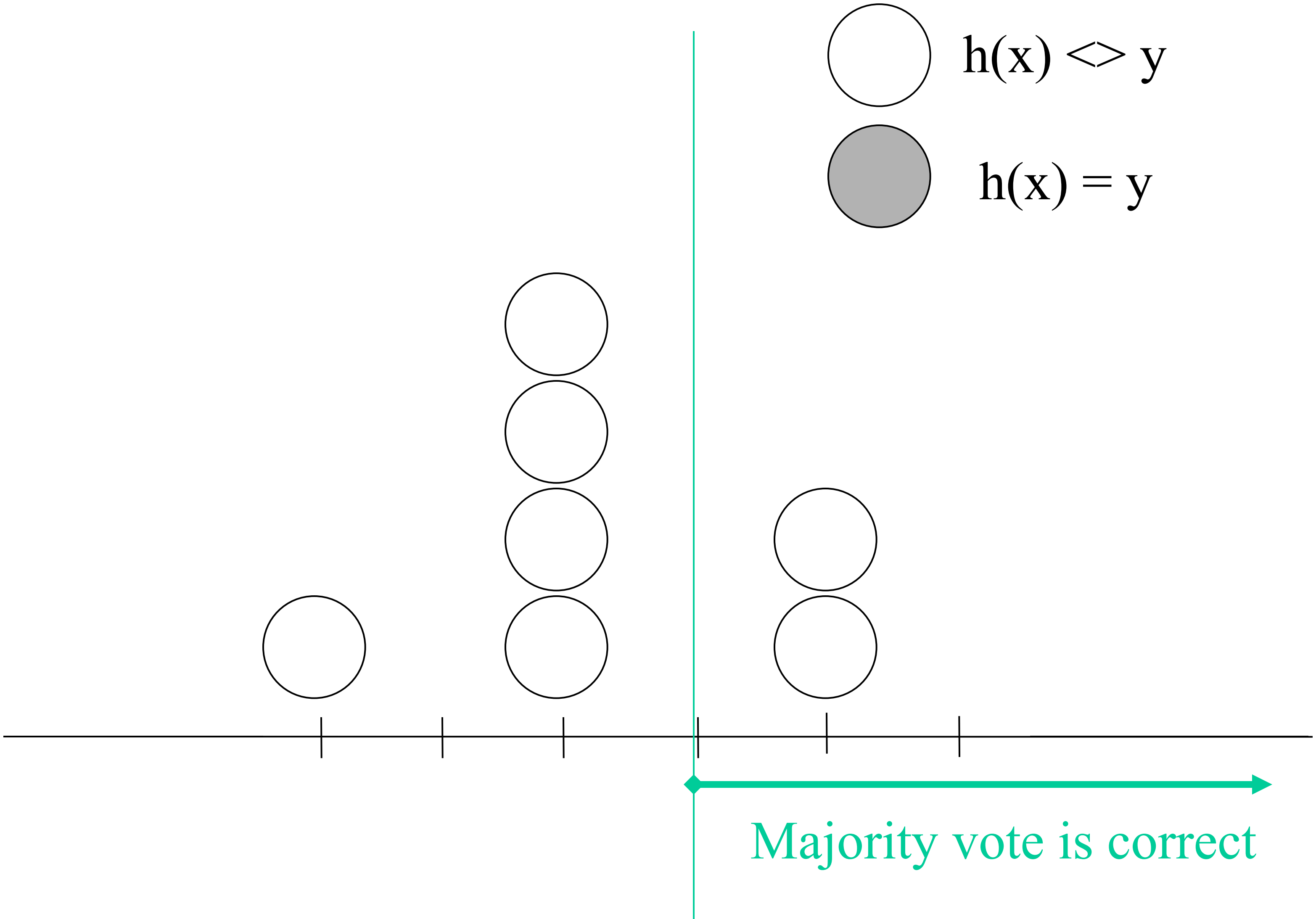


Final rule:

Boosting as a chip game

- Chip = training example
- Booster assigns a weight to each chip, weights sum to 1.
- Learner selects a subset with weight $\geq \frac{1}{2} + \gamma$
  - Selected set moves a step right (correct)
  - Unselected set moves a step left (incorrect)
- Booster wins examples on right of origin.
  - Implies that majority vote is correct.

The boosting chip game



○ h(x) <> y

● h(x) = y

Majority vote is correct

Ulam’s game with k lies

- Player 1 chooses secret x from S1,S2,…,SN
- Player 2 asks “is x in the set A?”
- Player 1 can lie k times.
- Game ends when player 2 can determine x.
- A smart player 1 aims to have more than one consistent x for as long as possible.
- Chip = element (Si)
- Bin = number of lies made regarding element

Ron Rivest’s game

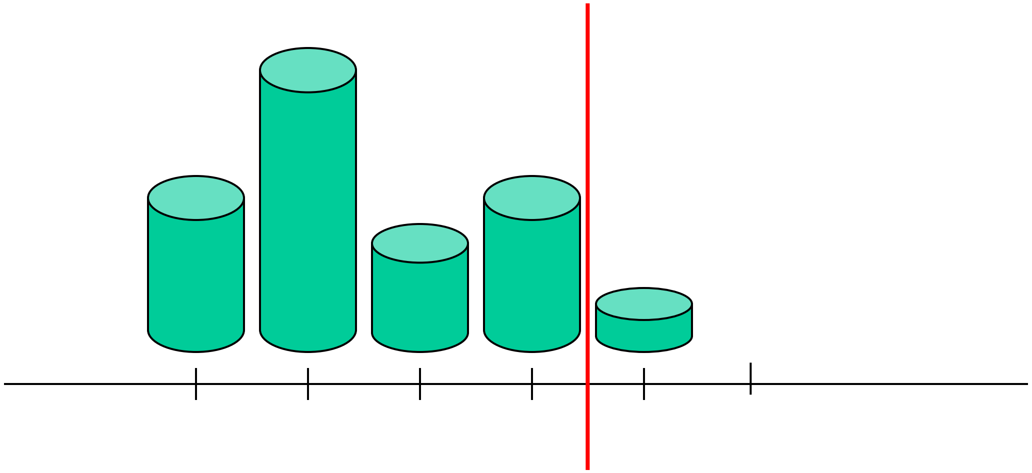
- You are given a bag with 1000 fair coins.
- Your goal is to have at least 700 “heads”
- You repeatedly choose one of two actions:

1) Take a single coin out of the bag and flip it.
2) Put all the coins back into the bag.

- Find strategy that minimizes the expected time to success?

Taking number of chips to infinity

Replace individual chips by chip mass



Each bin can be divided arbitrarily

Notation

loss of unit mass in bin *i* at game's end

Simple case:

Fraction of mass that is in bin *i* at time *t*

Goal of shepherd is to minimize

More Notation

Fraction of mass in bin *i* at time *t* that moves up

Weights assigned by Shepherd:

allowed movement given weighting

Backwards recursion

The Potential of a unit mass in bin *i* at time *t*

Given we express the min/max value for *t-1*

Matching min/max strategies

Shepherd's strategy

implies

Sheep's strategy

Recursive definition of potential

Definition:

Yields:

min-max value of the game
if at t=0 mass is concentrated in origin

Probabilistic interpretation

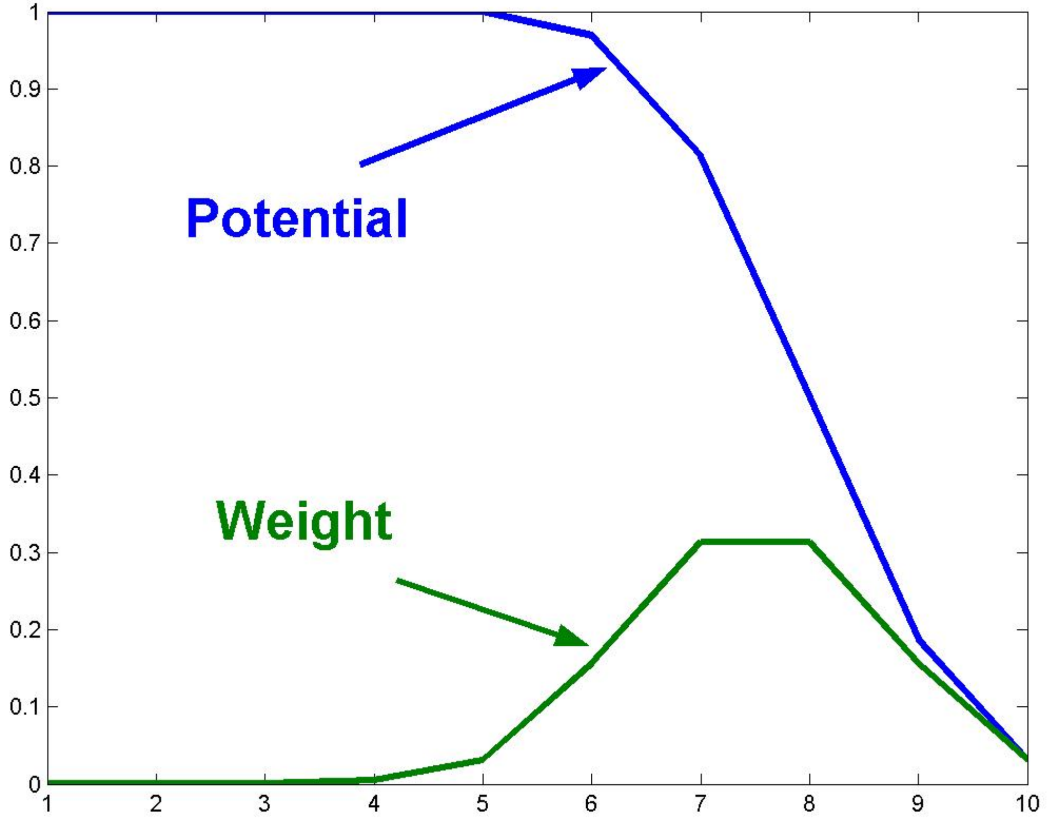
Simplifying assumption (not critical)

Each sheep moves up independently with probability

The expected final loss of a chip performing
a biased random walk starting from bin *i* at time *t*

Solution for boost-by-majority

Example potential and weight



Generalization to general vector spaces

General drifting game notation

	Sheep mass in 1d	Discrete Sheep in d>1
location	Bin no. i	Sheep j location
Sheep steps		bounded set
Shepherd’s choice		

The min/max solution

[Schapire99]

- A potential defined by a min/max recursion

$$\phi_T(\mathbf{s}) = L(\mathbf{s})$$

$$\phi_{t-1}(\mathbf{s}) = \min_{\mathbf{w}\in\mathbb{R}^d} \sup_{\mathbf{z}\in B} (\phi_t(\mathbf{s}+\mathbf{z}) + \mathbf{w}\cdot\mathbf{z} - \delta \|\mathbf{w}\|)$$

$\phi_0(0) =$ the value of the game

- Shepherd's strategy

$$\mathbf{w}_i^t = \arg\min_{\mathbf{w}} \sup_{\mathbf{z}\in B} (\phi_t(\mathbf{s}+\mathbf{z}) + \mathbf{w}\cdot\mathbf{z} - \delta \|\mathbf{w}\|)$$

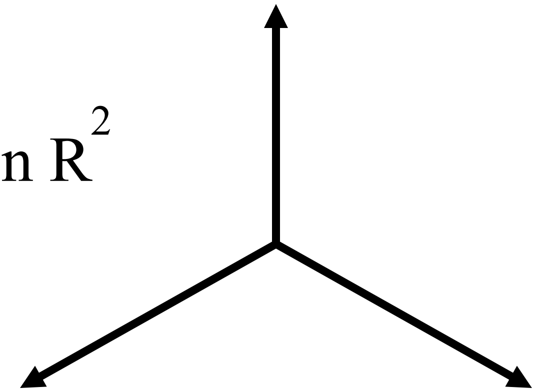
Restricting allowed steps

B = the set of all allowable step

Normal B = minimal set that spans the space. (~basis)

Regular B = a symmetric regular set. (~orthonormal basis)

Regular step set in R^2



The solution simplifies when $\delta \to 0$

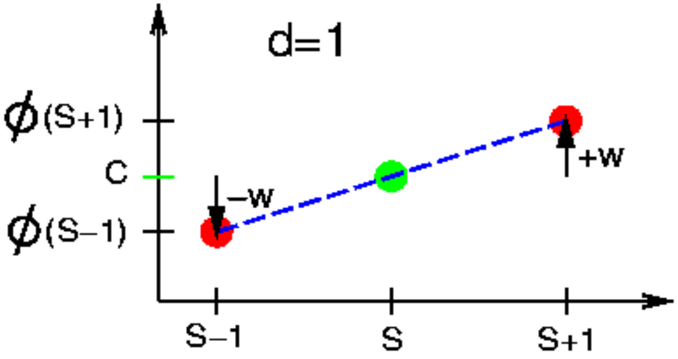
If B is normal, and δ is sufficiently small then $\exists \mathbf{w}^*$ such that

$$\phi_{t-1}(\mathbf{s}) = \phi_t(\mathbf{s}+\mathbf{z}) + \mathbf{w}^* \cdot \mathbf{z} - \delta \|\mathbf{w}^*\|$$

for all $\mathbf{z} \in B$ (and all $t = 1, 2 \ldots$, $\mathbf{s} \in \mathbb{R}^d$)

Implies that: $\mathbf{w}^*$ is the "local slope" at $\phi_t(\mathbf{s})$, i.e.

$$\phi_t(\mathbf{s}+\mathbf{z}_i) = C + \mathbf{w}^*\mathbf{z}_i \; ; \quad C \doteq \frac{\Sigma_{j=0}^{d} \phi_t(\mathbf{s}+\mathbf{z}_j)}{d+1}$$



and that

$$\phi_{t-1}(\mathbf{s}) = C - \delta \|\mathbf{w}^*\|$$

Increasing the number of steps

- Consider T steps in a unit time
- Drift d should scale like 1/T
- Step size O(1/T) gives game to shepherd
- Step size $O\left(1/\sqrt{T}\right)$ keeps game balanced

The solution for

The local slope becomes the gradient

The potential is a solution of a PDE

$$\frac{\partial \phi(\mathbf{s}, \tau)}{\partial \tau} = -\frac{1}{2} \sum_{k,l} D_{kl} \frac{\partial^2 \phi(\mathbf{s}, t)}{\partial s^k \partial s^l} + \delta \left\| \nabla \phi(\mathbf{s}, \tau) \right\|_p$$

Same PDE describes time evolution of
Brownian motion with drift proportional to gradient

Applications of continuous drifting games

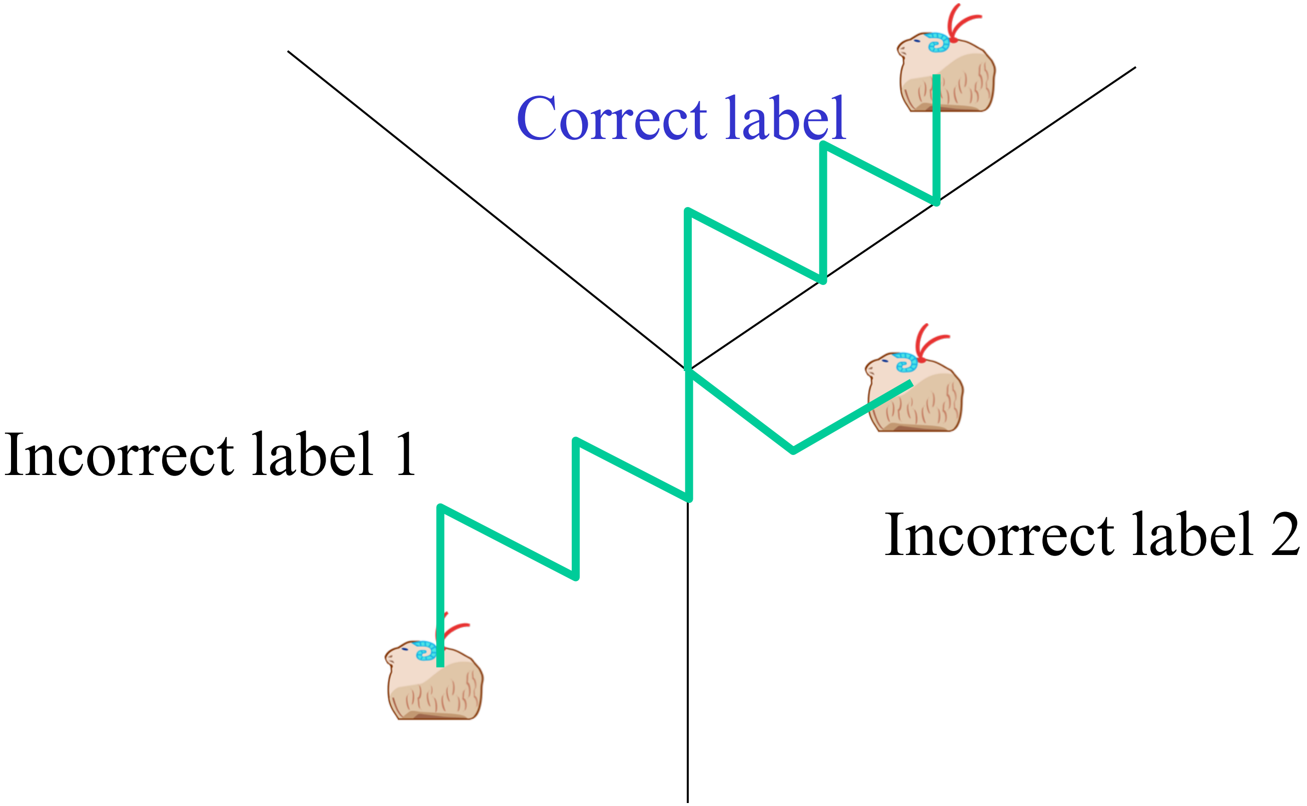
Some known solutions for 1d

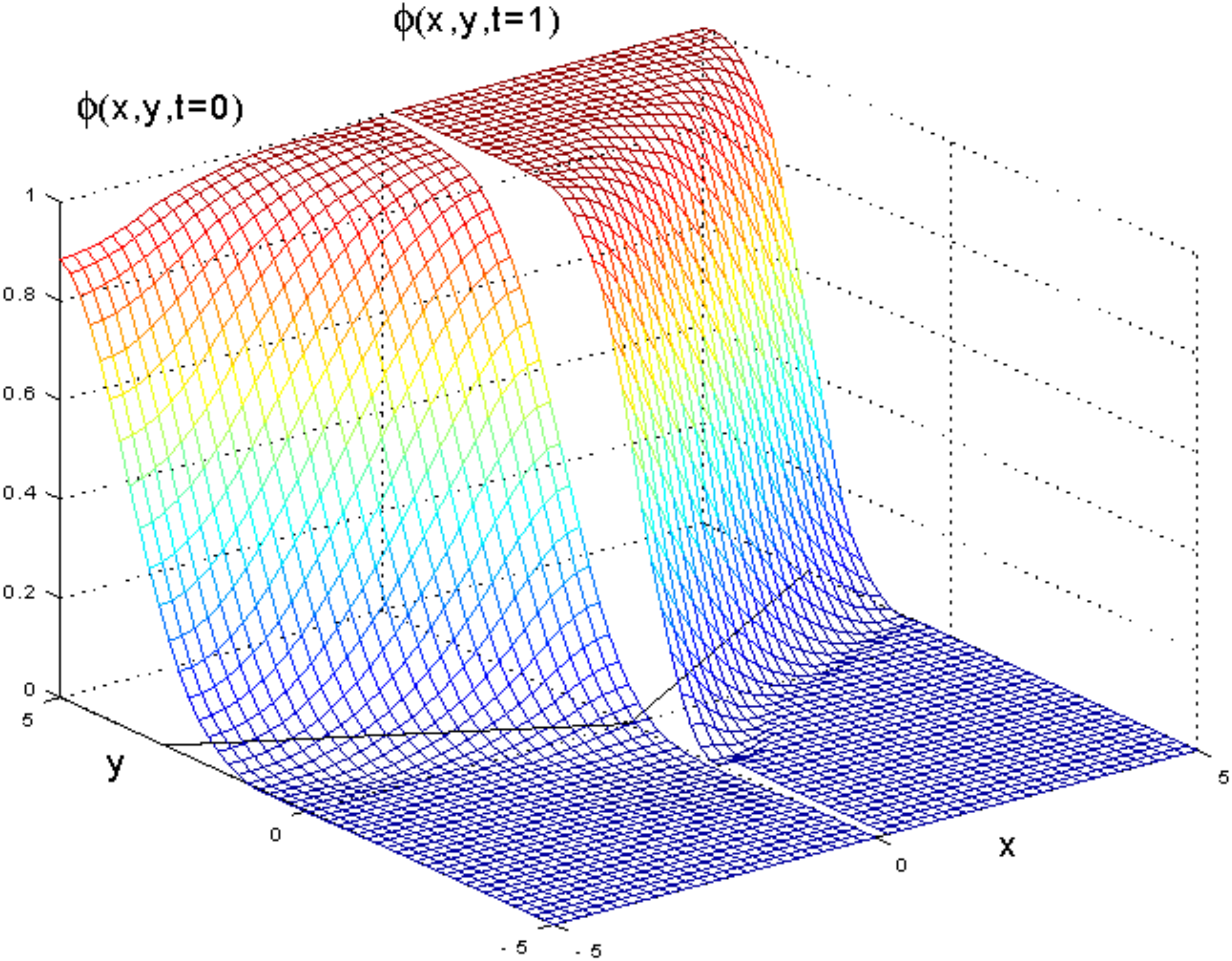
Adaboost [Freund,Schapire 97]

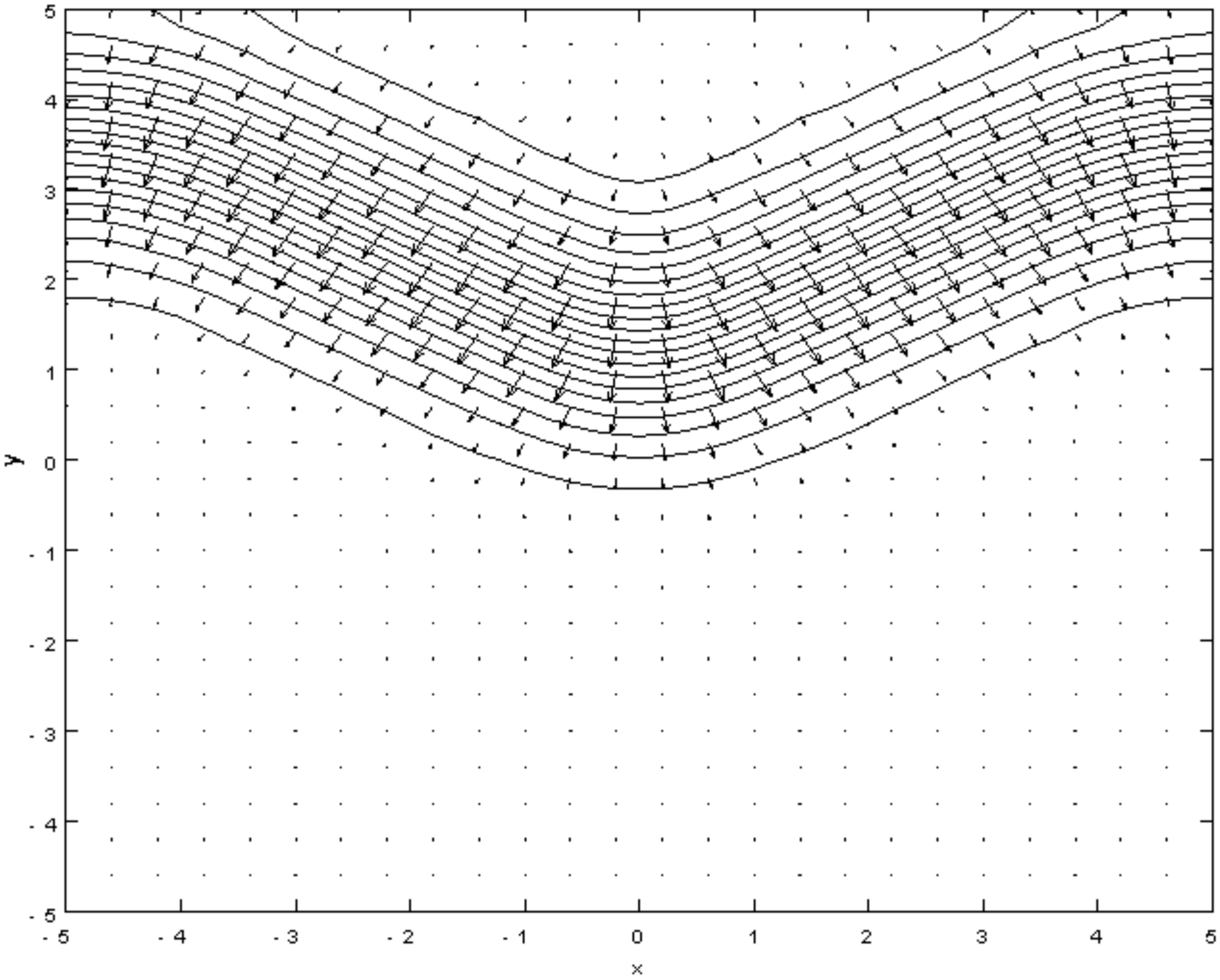
Brownboost [Freund 01]

Boosting for more than 2 labels

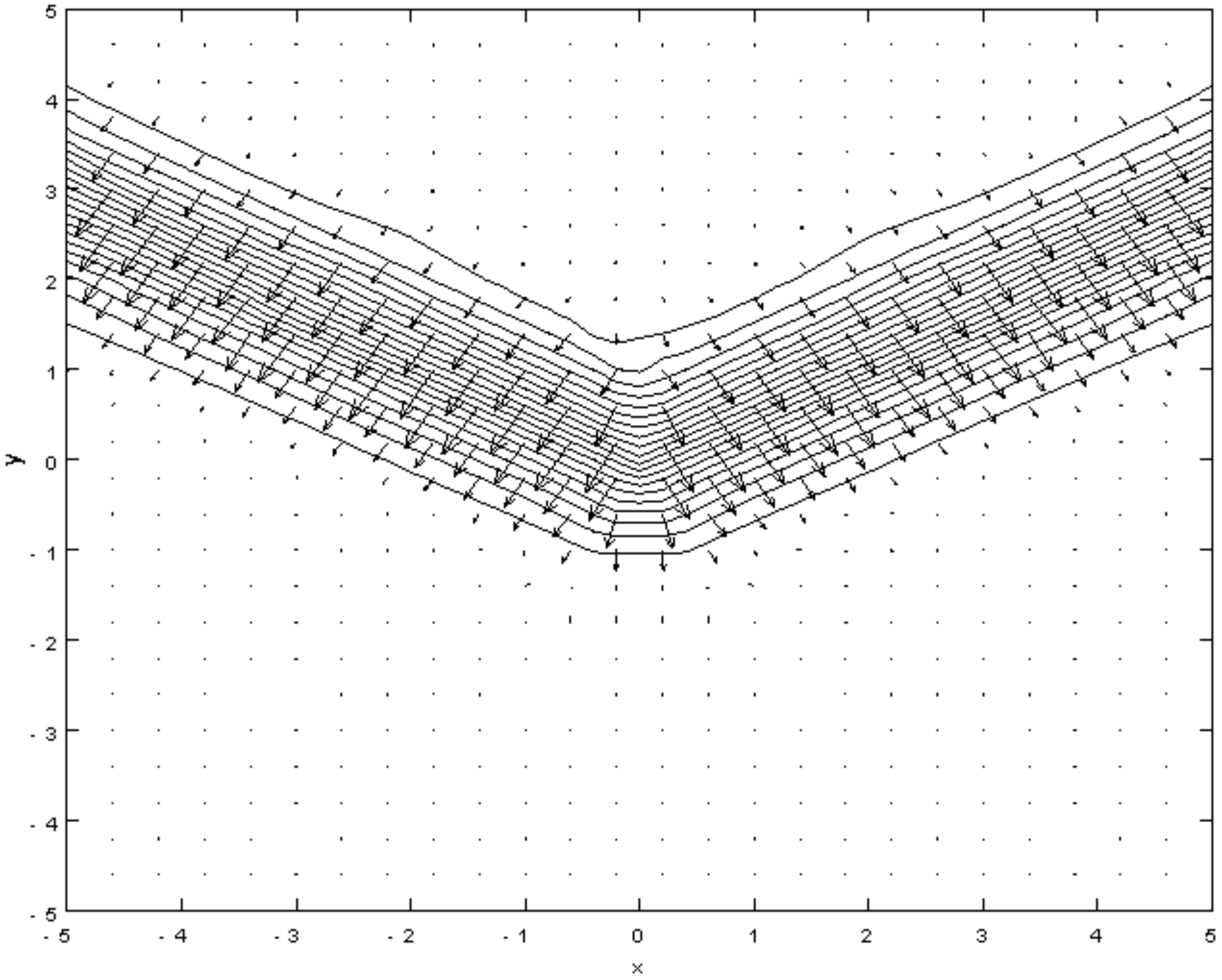
Predict according to the most popular prediction



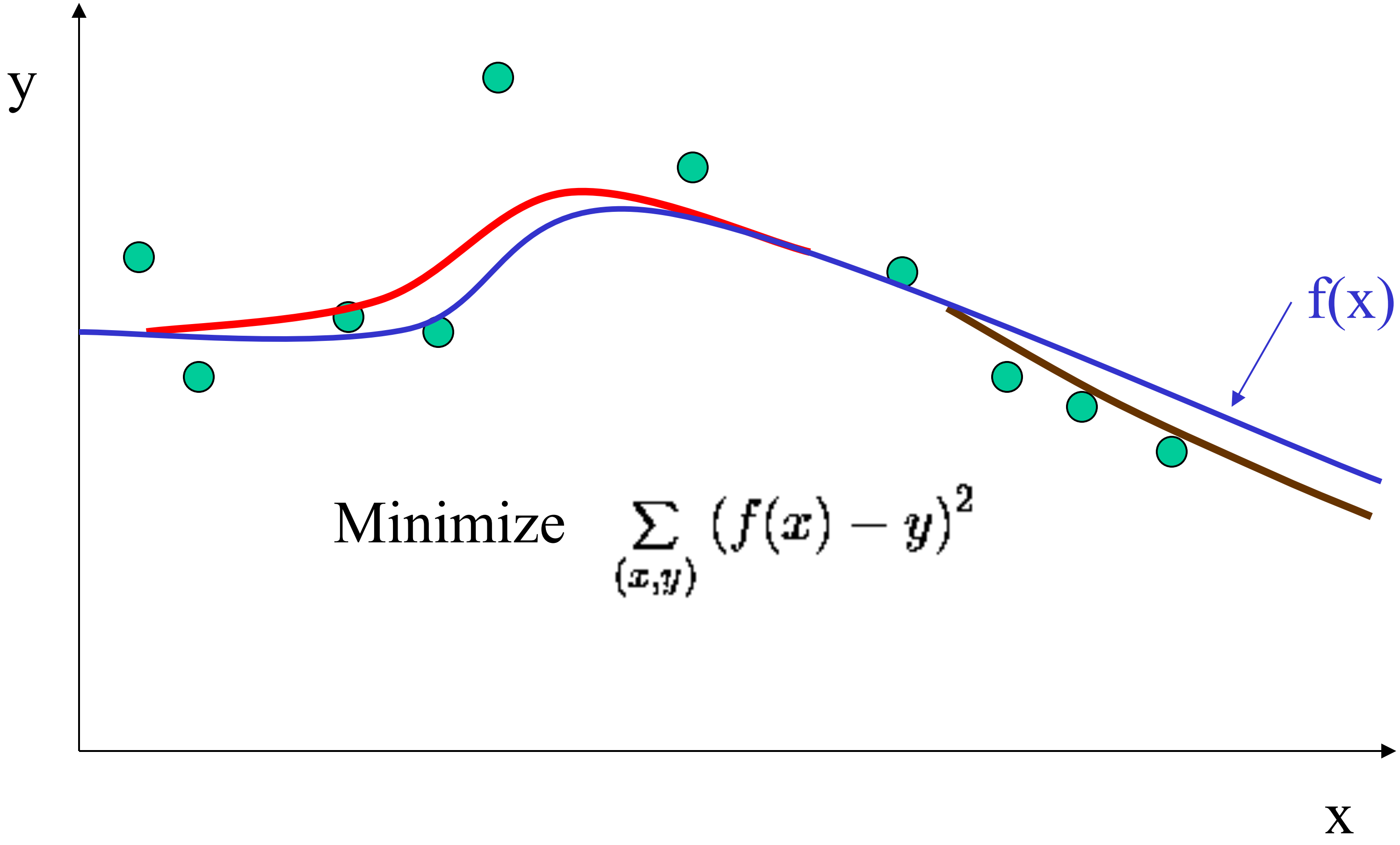

φ(x,y,t=1)
φ(x,y,t=0)
1
0.8
0.6
0.4
0.2
0
5
y
0
-5
-5
0
x
5



5
4
3
2
1
0
- 1
- 2
- 3
- 4
- 5
y
- 5
- 4
- 3
- 2
- 1
0
1
2
3
4
5
x


5
4
3
2
1
0
-1
-2
-3
-4
-5
y
-5
-4
-3
-2
-1
0
1
2
3
4
5
x

Boosting for variational optimization



Minimize $\sum_{(x,y)} (f(x) - y)^2$

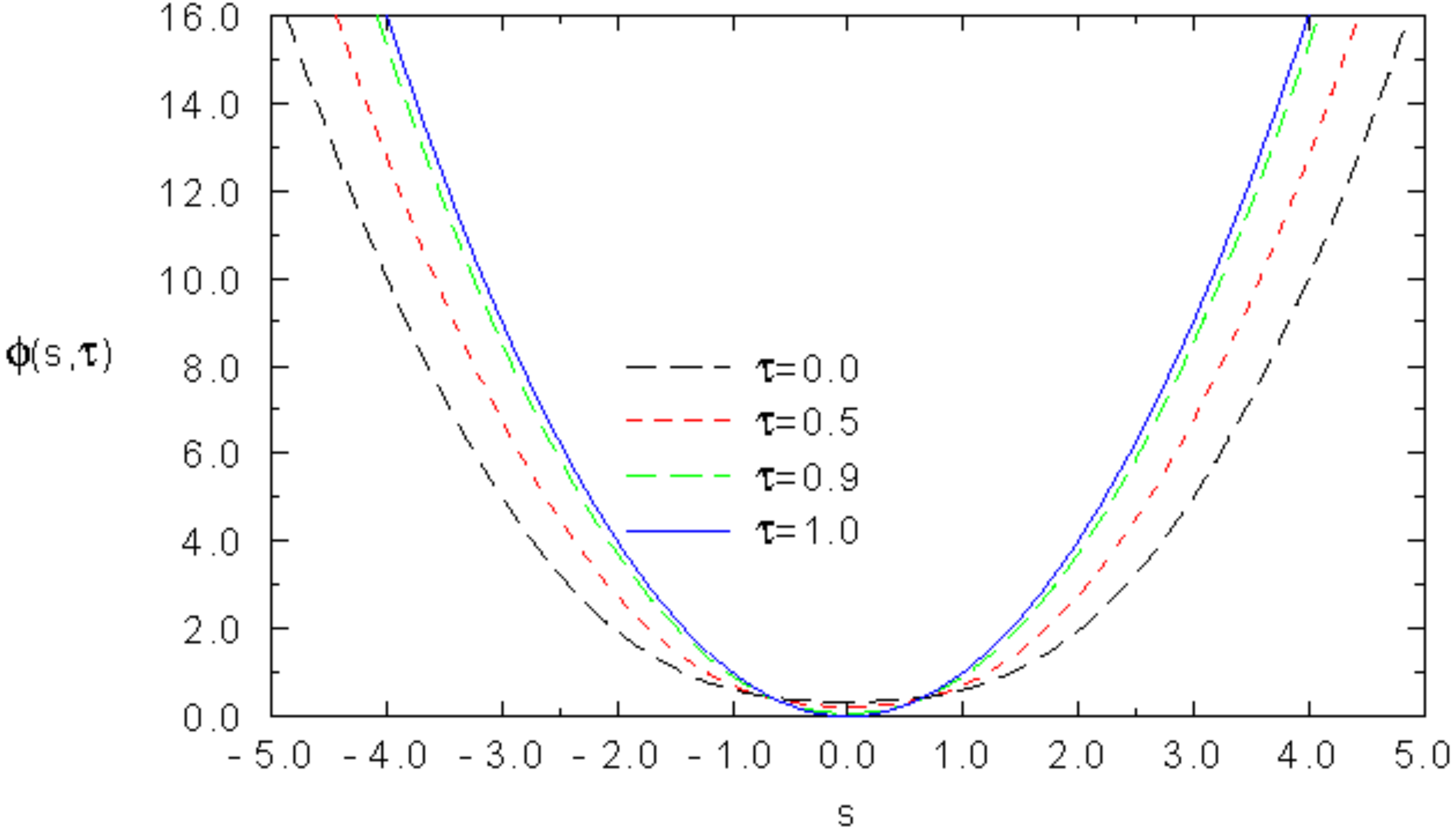
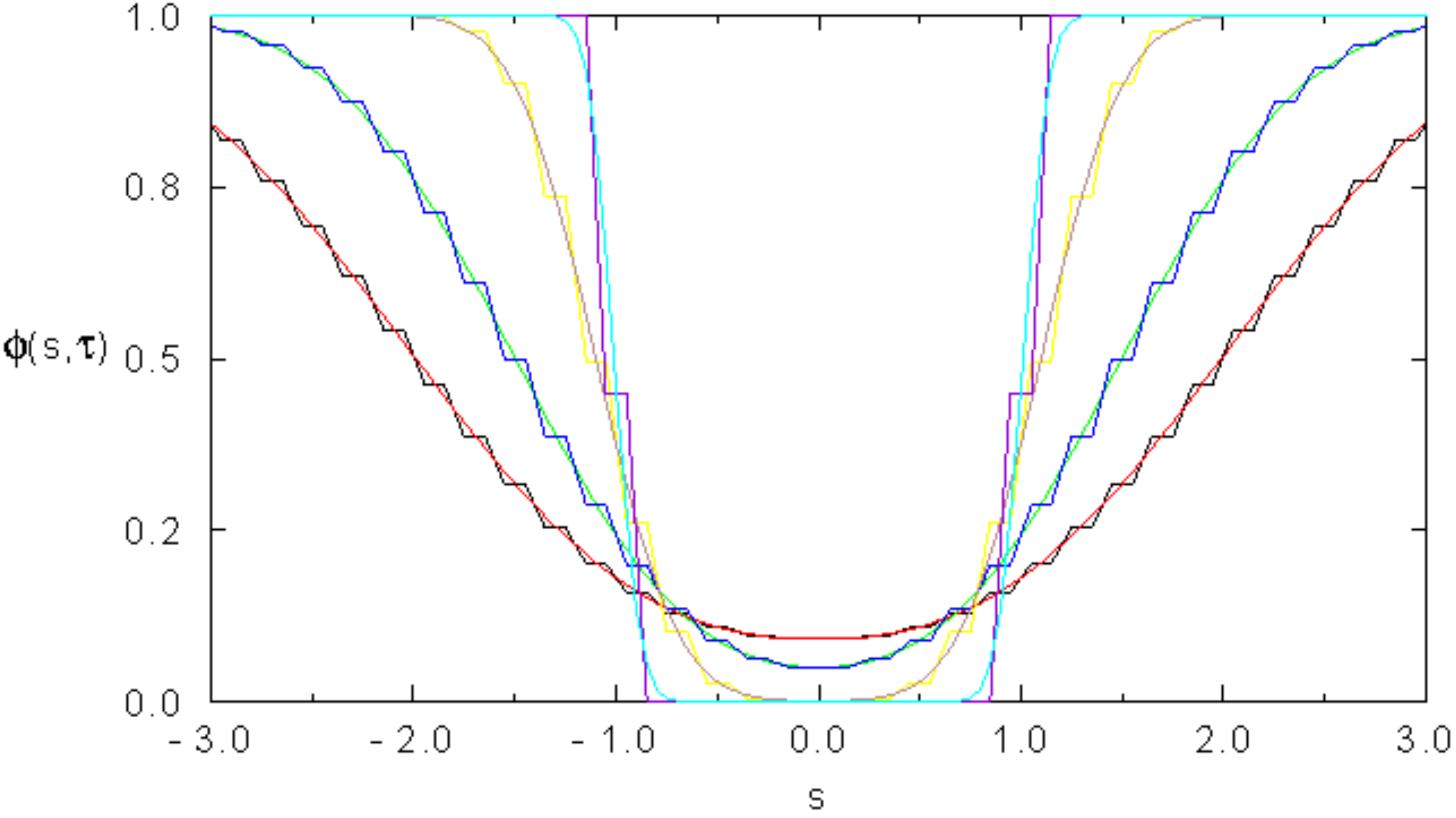


Figure 2: The potential $\phi(\mathbf{s}, t)$ for the square loss $L(y) = y^2$.

solution for $L(s) = I_{|s|>1}$



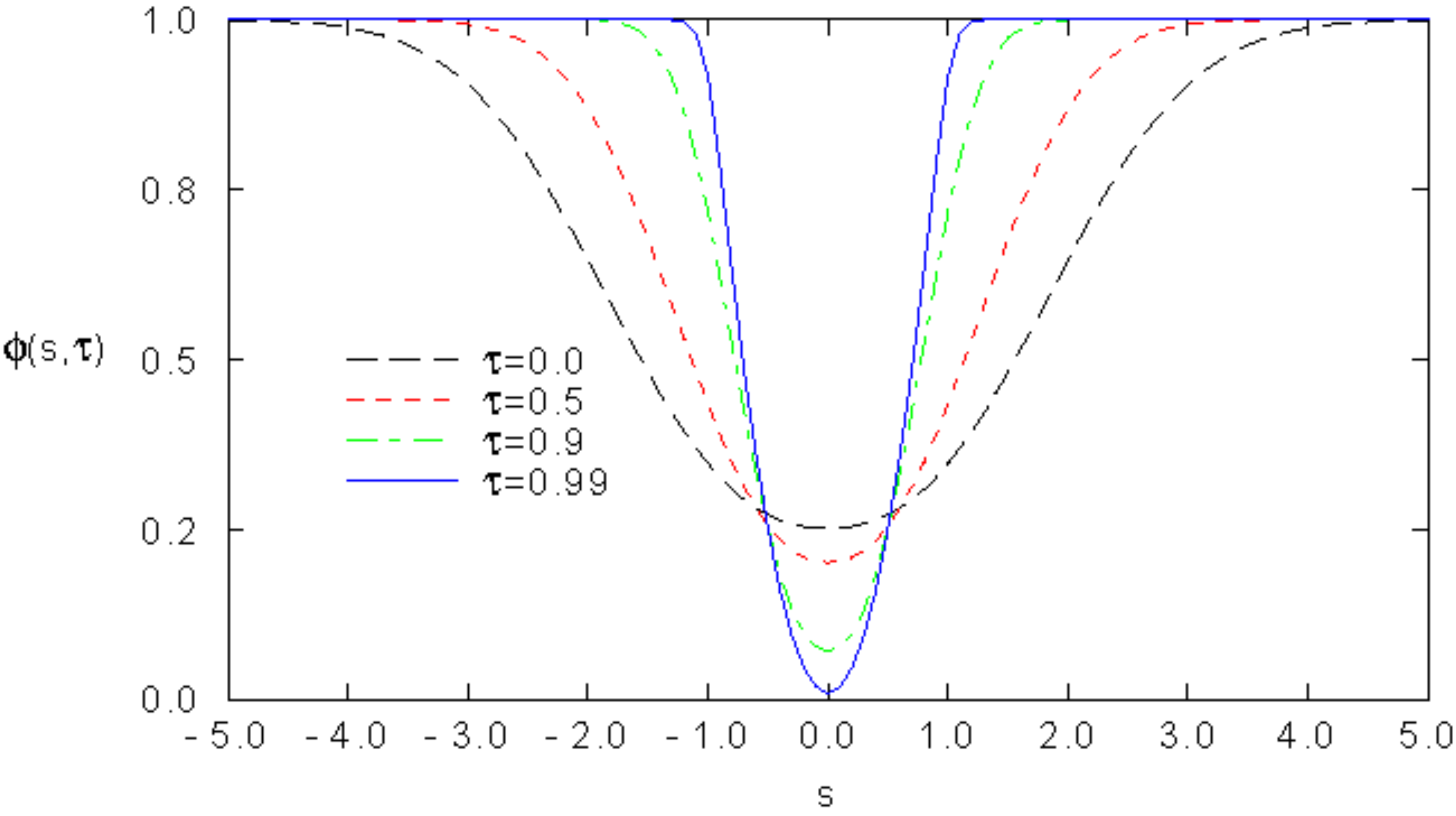


Figure 3: The potential $\phi(\mathbf{s}, \boldsymbol{t})$ for the loss $\boldsymbol{L(y)} = \mathbf{min}(\boldsymbol{y}^{\mathbf{2}}, 1)$.

Normalized drifting

- Boosting tends not to over-fit
- Explanation: “boosting the margins” [schapire et at. 98]
- Goal is to minimize

Normalized drifting game in 1d

Regular steps

Leads to Brownian motion

Normalized steps

What is the appropriate limit for ?

1d problem with 3 allowed steps.

- Allowed steps: {-1,0,+1}
- Importance: allows weak rules that abstain.
- Worst case choice of sheep depends on whether potential is convex or concave.

Open question

- In the continuous time limit, assuming that there are two regimes what characterizes the boundary condition between them?
- “The one-phase Stefan problem with temperature-boundary specification”?
- “The least sub-parabolic majorant of a function *u*”?
- The boundary between solid and fluid?