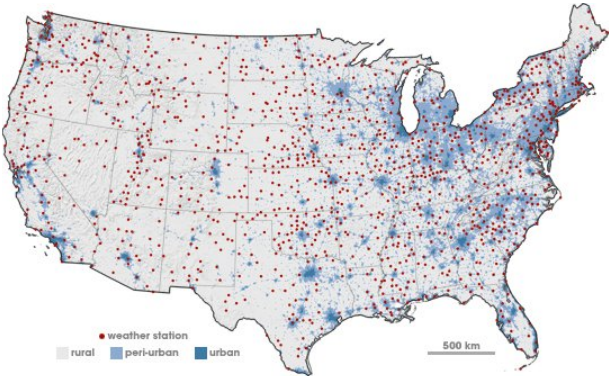
Protocols for Learning Classifiers on Distributed Data

Suresh Venkatasubramanian
University of Utah

Joint work with Hal Daumé III, Jeff Phillips, and Avishek Saha

Distributed Data



courtesy the Earth Observatory

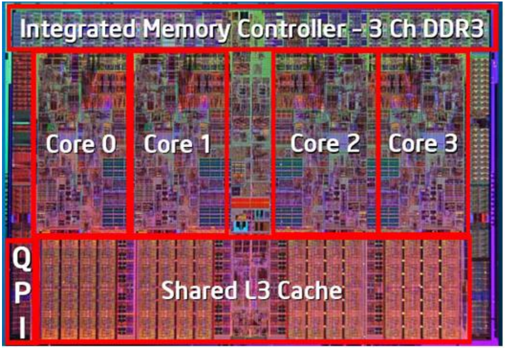
Data can be distributed across geographically distinct locations

Distributed Data

courtesy Royal Pingdom

Data can be distributed across geographically distinct locations

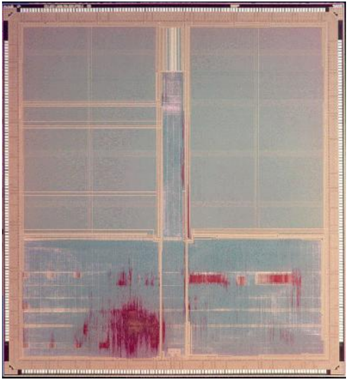
Distributed Data



Intel's Nehalem (Core I7) chip

Data can be distributed even inside a single machine

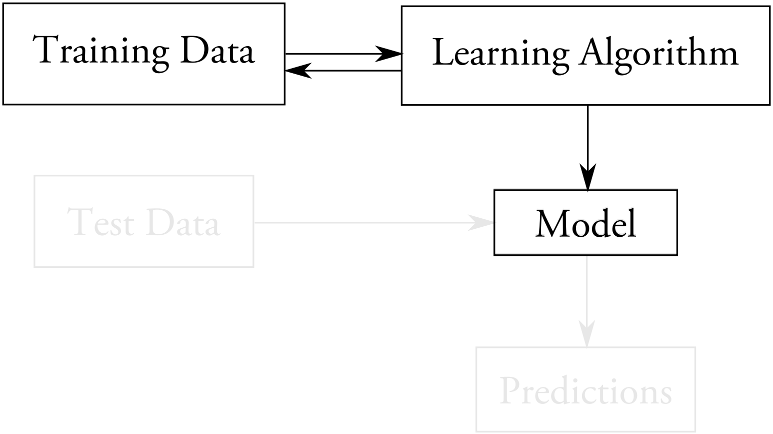
Distributed Data



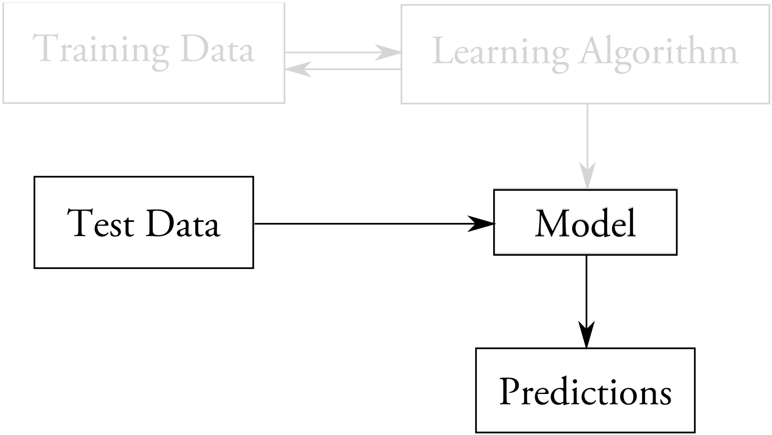
Processer-in-memory (PIM)

Data can be distributed even inside a single machine

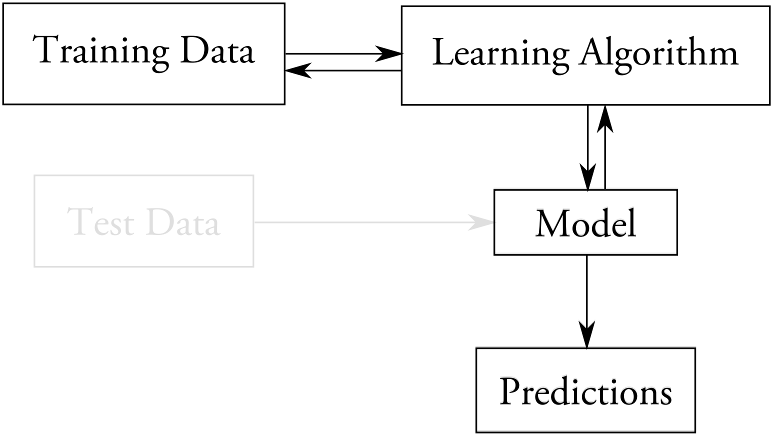
The Mechanics of Learning



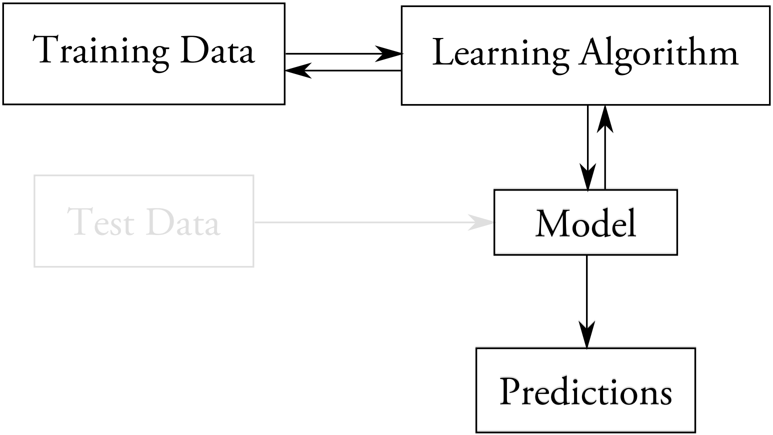
The Mechanics of Learning



The Mechanics of Learning



The Mechanics of Learning



In all cases, data is easily accessible by learning algorithm!

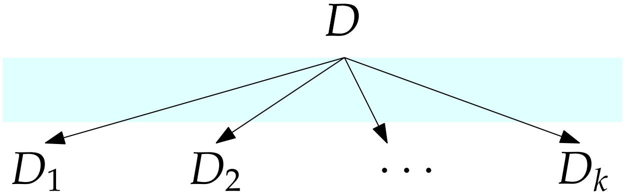
Distributed versus Parallel

Parallel Learning: you have control over all of data

$$D$$

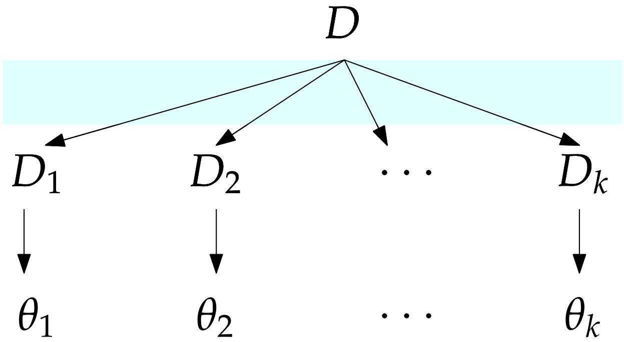
Distributed versus Parallel

Parallel Learning: you have control over all of data



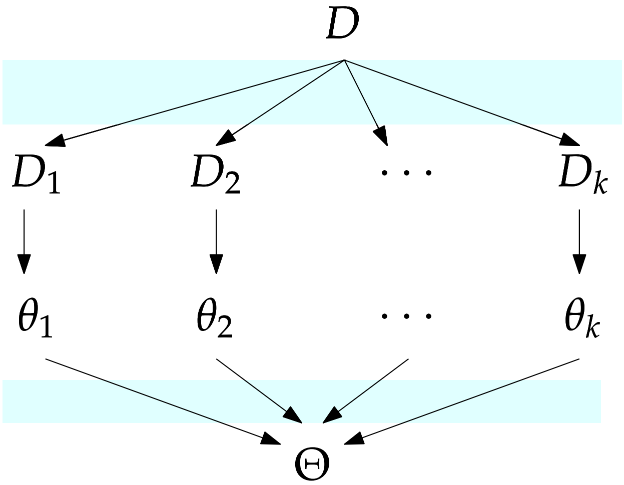
Distributed versus Parallel

Parallel Learning: you have control over all of data



Distributed versus Parallel

Parallel Learning: you have control over all of data



How can we learn in a communication-restricted environment ?

A simple model for distributed learning

- k "players"
- Each player owns data D_i. Let $D = \cup_i D_i$
- Learning task T, solution h, error $\text{err}(h, D, T)$.

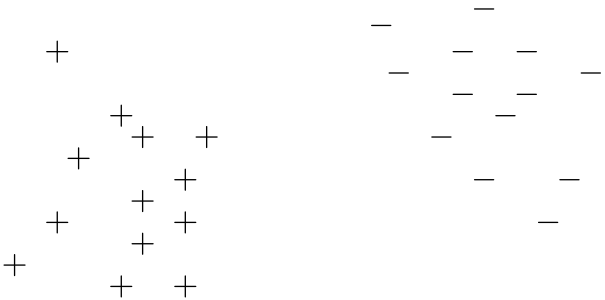
Problem

Given $\epsilon > 0$, design protocol to let players agree on solution $\tilde{h}$ such that

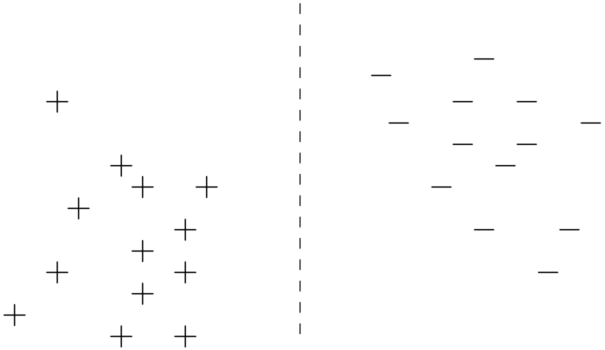
$$err(\tilde{h}, D, T) \leq err(h^*, D, T) + \epsilon$$

with minimum inter-player communication.

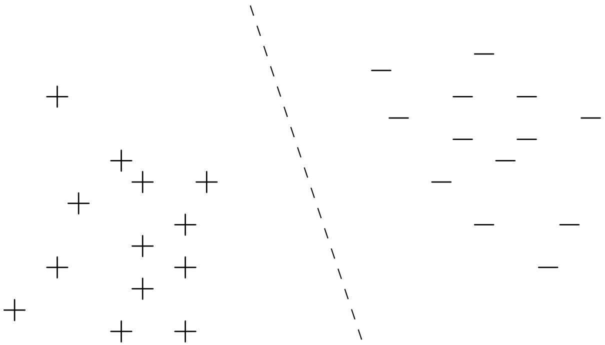
Learning a classifier



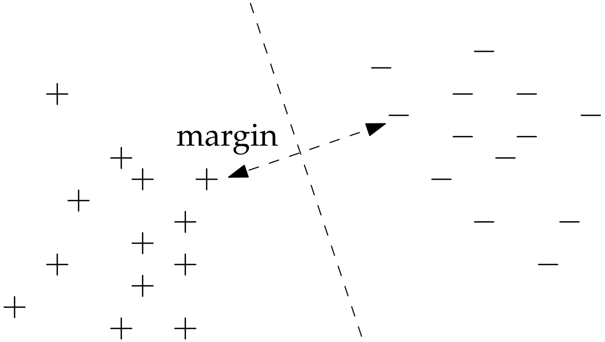
Learning a classifier



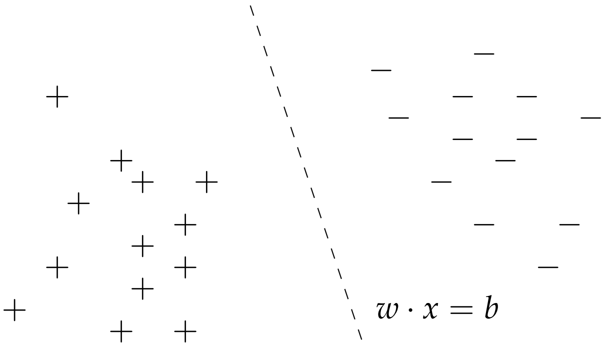
Learning a classifier



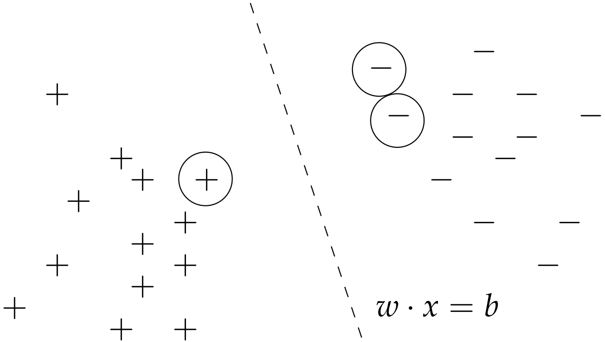
Learning a classifier



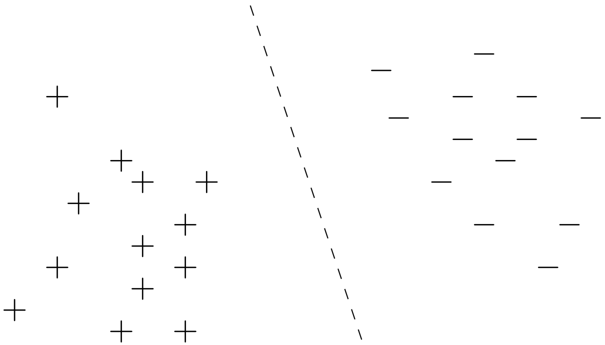
Learning a classifier



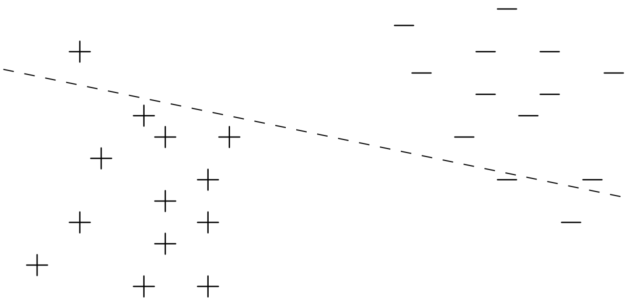
Learning a classifier



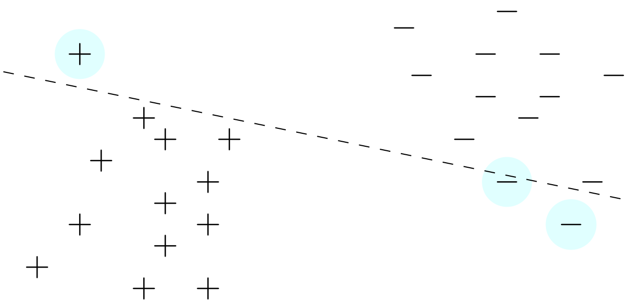
Learning a classifier



Learning a classifier

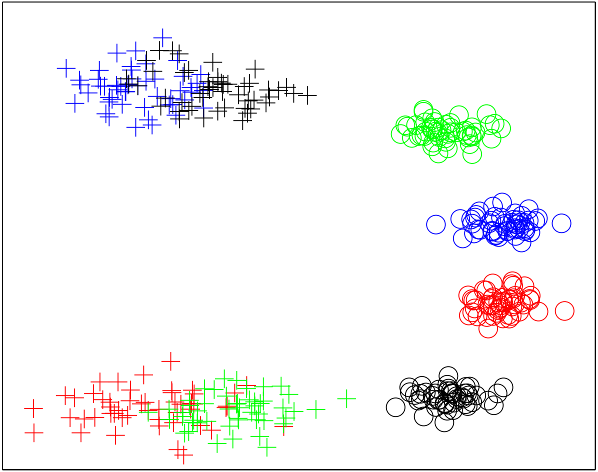


Learning a classifier



misclassification error = fraction of mistakes

Can we merely exchange classifiers ?



A simple result: randomly partitioned data

Definition

Given a range space $(X, \mathcal{R})$, a set $S \subset X$ is an ϵ-net if for all $R \in \mathcal{R}$,

$$|R \cap X| \geq \epsilon \implies R \cap S \neq \emptyset$$

Theorem

Any range space $(X, \mathcal{R})$ of VC-dimension d has an ϵ-net of size $O(\frac{d}{\epsilon} \log \frac{1}{\epsilon})$

A simple result: randomly partitioned data

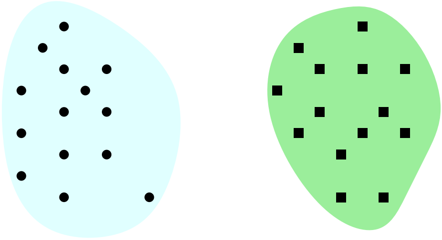
Definition

Given a range space $(X, \mathcal{R})$, a set $S \subset X$ is an ϵ-net if for all $R \in \mathcal{R}$,

$$|R \cap X| \geq \epsilon \implies R \cap S \neq \emptyset$$

Theorem

Any range space $(X, \mathcal{R})$ of VC-dimension d has an ϵ-net of size $O(\frac{d}{\epsilon} \log \frac{1}{\epsilon})$



A simple result: randomly partitioned data

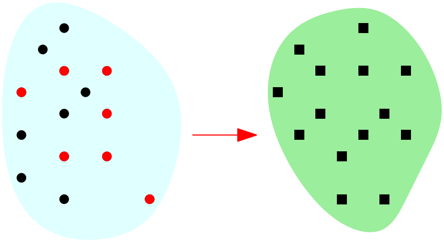
Definition

Given a range space $(X, \mathcal{R})$, a set $S \subset X$ is an ϵ-net if for all $R \in \mathcal{R}$,

$$|R \cap X| \geq \epsilon \implies R \cap S \neq \emptyset$$

Theorem

Any range space $(X, \mathcal{R})$ of VC-dimension d has an ϵ-net of size $O(\frac{d}{\epsilon} \log \frac{1}{\epsilon})$



A simple result: randomly partitioned data

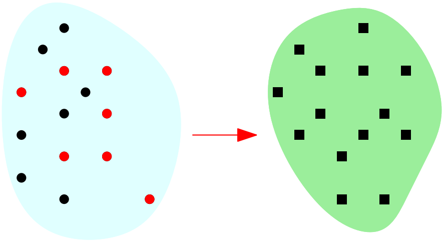
Definition

Given a range space $(X, \mathcal{R})$, a set $S \subset X$ is an ϵ-net if for all $R \in \mathcal{R}$,

$$|R \cap X| \geq \epsilon \implies R \cap S \neq \emptyset$$

Theorem

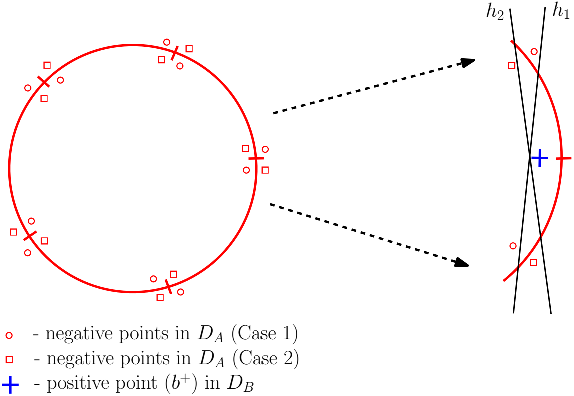
Any range space $(X, \mathcal{R})$ of VC-dimension d has an ϵ-net of size $O(\frac{d}{\epsilon} \log \frac{1}{\epsilon})$



A Lower bound

Lemma

Any one-way protocol for learning an ϵ-error classifier requires $\Omega(1/\epsilon)$ communication.



Proof by reduction from INDEXING

Two-way communication

Theorem

There exists a two-way protocol for two players that can compute an ϵ-error linear classifier for labelled points in $\mathbb{R}^2$ using $O(\log(1/\epsilon))$ bits of communication.

Two-way communication

Theorem

There exists a two-way protocol for two players that can compute an ϵ-error linear classifier for labelled points in $\mathbb{R}^2$ using $O(\log(1/\epsilon))$ bits of communication.

High level idea:

- In each round, each player exchanges a constant amount of information with the other.

Two-way communication

Theorem

There exists a two-way protocol for two players that can compute an ϵ-error linear classifier for labelled points in $\mathbb{R}^2$ using $O(\log(1/\epsilon))$ bits of communication.

High level idea:

- In each round, each player exchanges a constant amount of information with the other.
- After each round, the number of points misclassified by a player reduces by a factor of 2.

Two-way communication

Theorem

There exists a two-way protocol for two players that can compute an ϵ-error linear classifier for labelled points in $\mathbb{R}^2$ using $O(\log(1/\epsilon))$ bits of communication.

High level idea:

- In each round, each player exchanges a constant amount of information with the other.
- After each round, the number of points misclassified by a player reduces by a factor of 2.
- The number of misclassified points is at most n, and only needs to get below ϵn.

Two-way communication

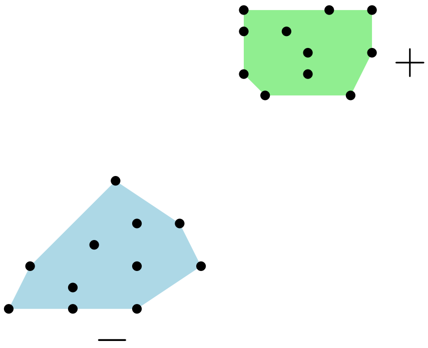
Theorem

There exists a two-way protocol for two players that can compute an ϵ-error linear classifier for labelled points in $\mathbb{R}^2$ using $O(\log(1/\epsilon))$ bits of communication.

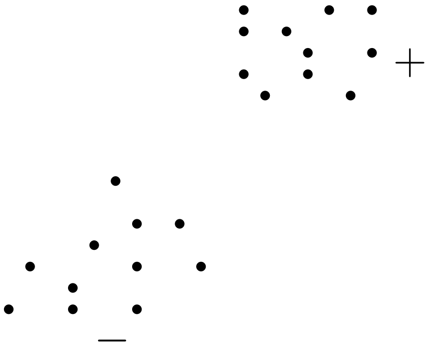
High level idea:

- In each round, each player exchanges a constant amount of information with the other.
- After each round, the number of points misclassified by a player reduces by a factor of 2.
- The number of misclassified points is at most n, and only needs to get below ϵn.
- Therefore, the number of rounds is $\log \frac{1}{\epsilon}$

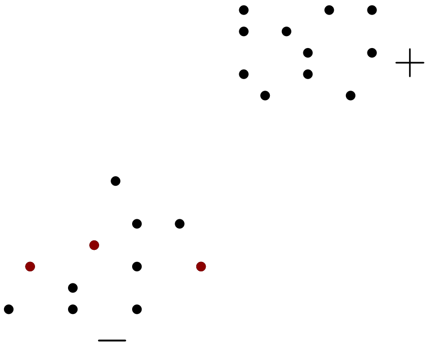
Regions of Uncertainty



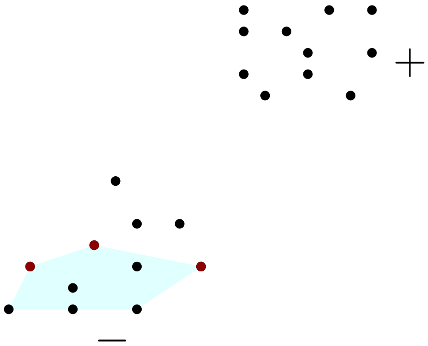
Regions of Uncertainty



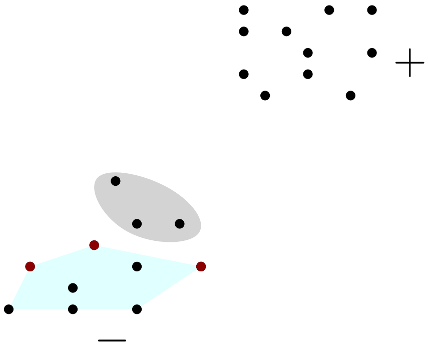
Regions of Uncertainty



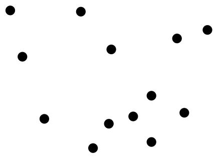
Regions of Uncertainty

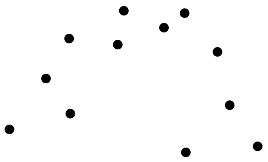


Regions of Uncertainty

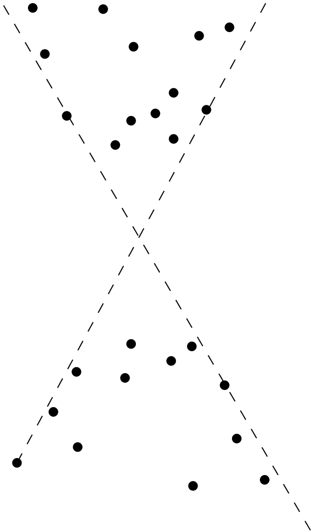


A's first move

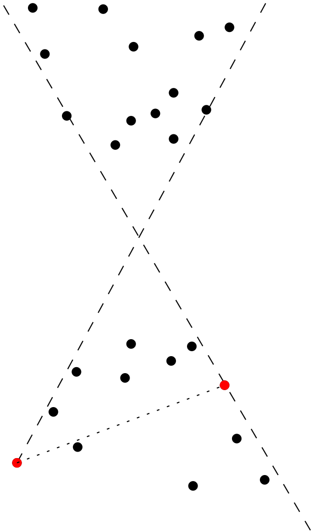




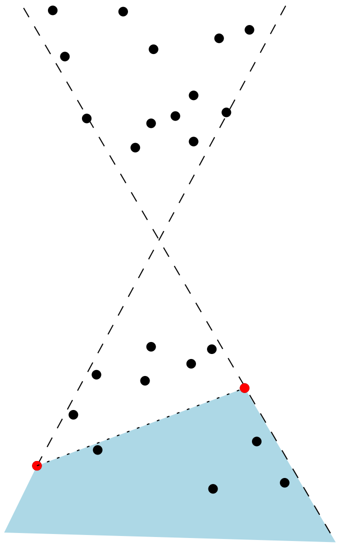
A's first move



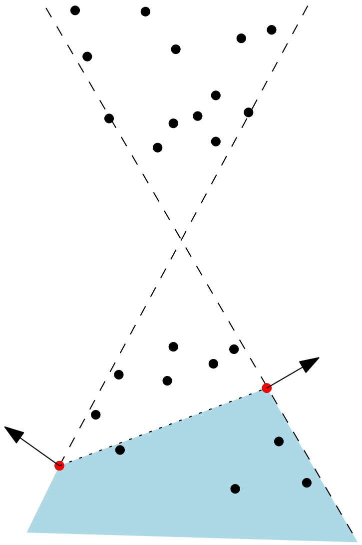
A's first move



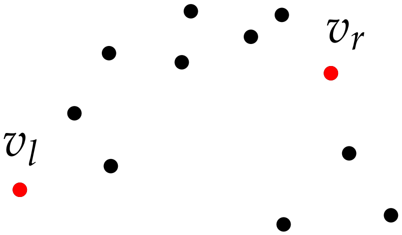
A's first move



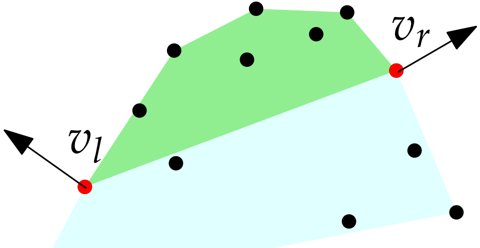
A's first move



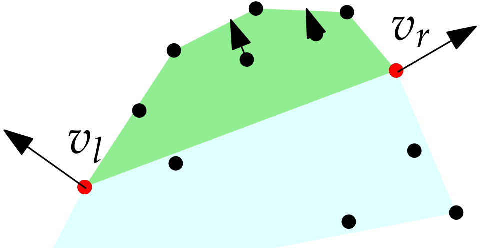
A's first move



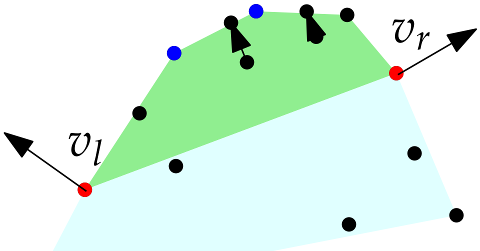
A's first move



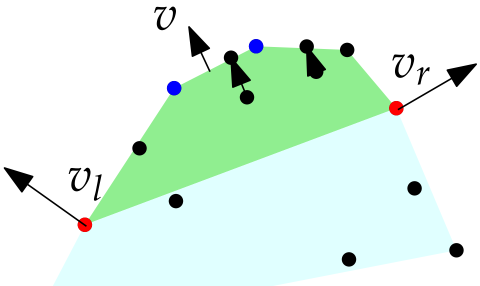
A's first move



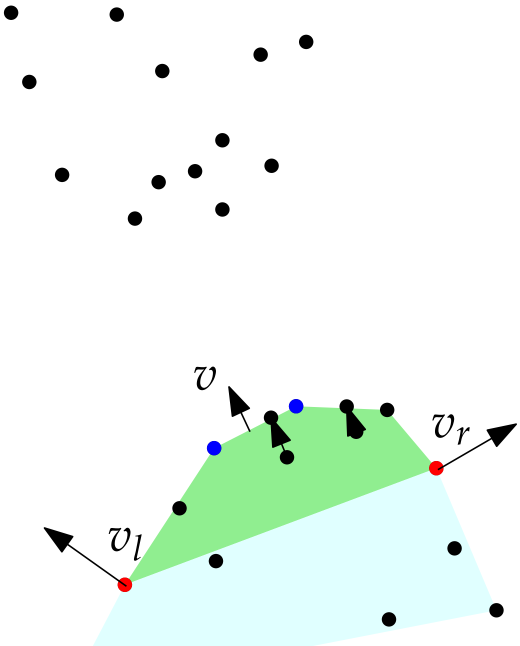
A's first move



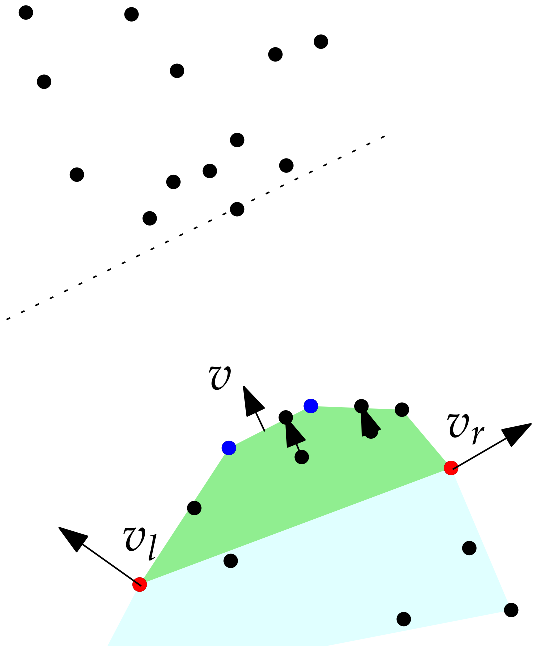
A's first move



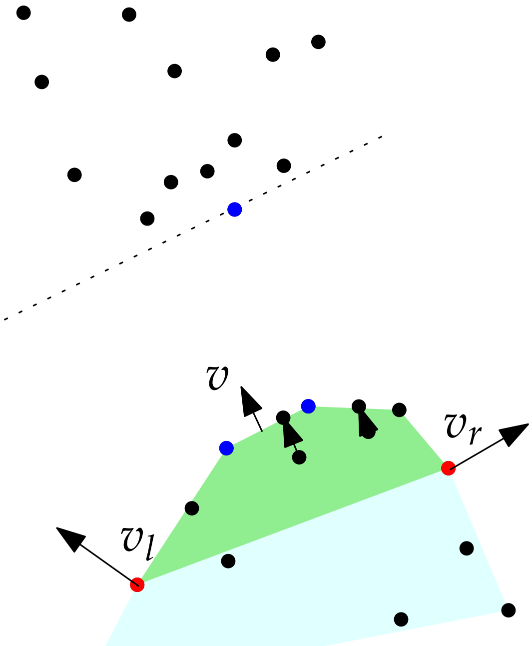
A's first move



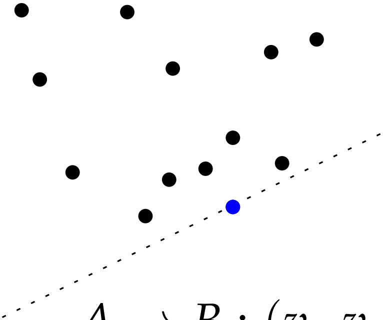
A's first move



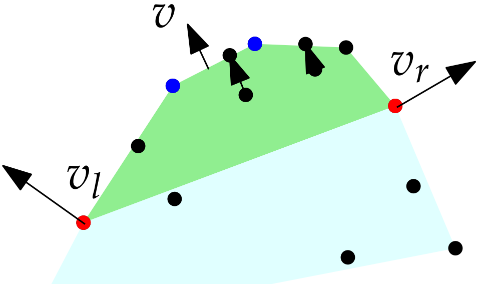
A's first move



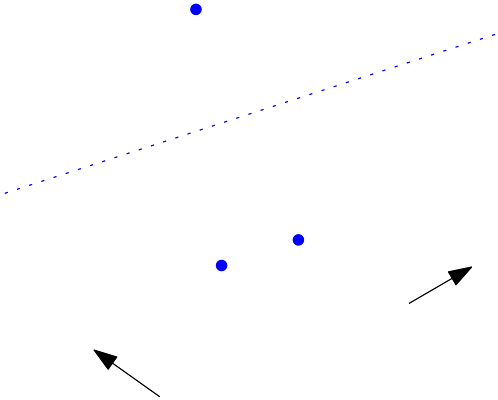
A's first move



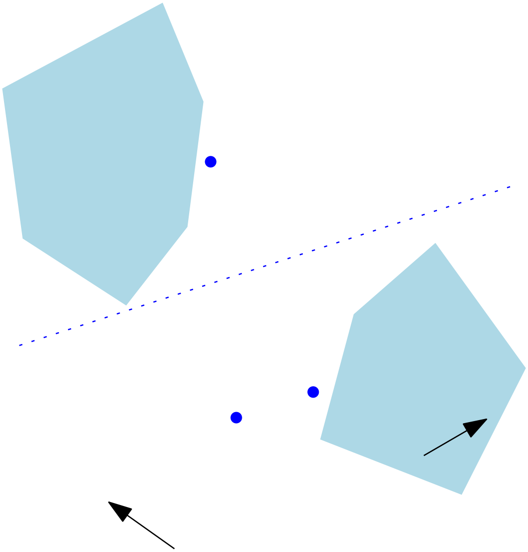
$$A \to B : (v_l, v, v_r, \bullet)$$



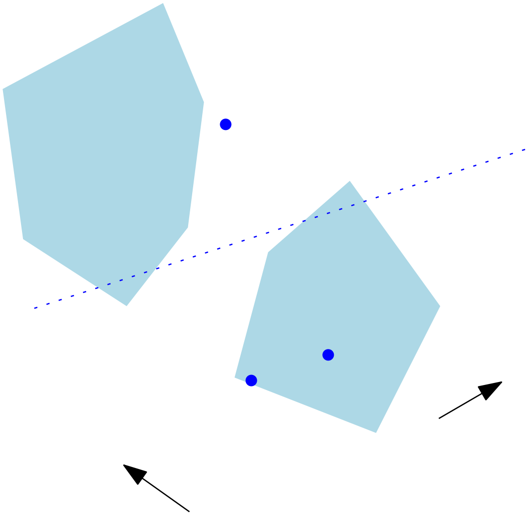
B's response



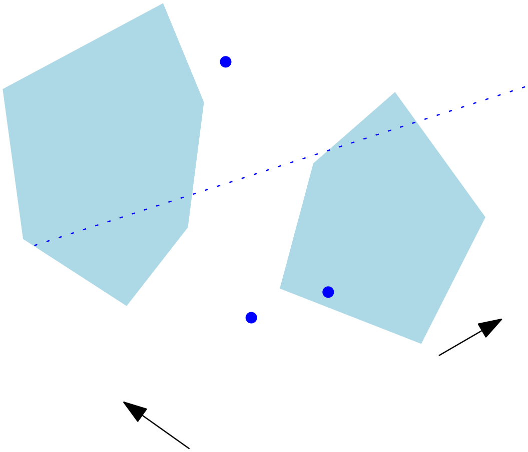
B's response



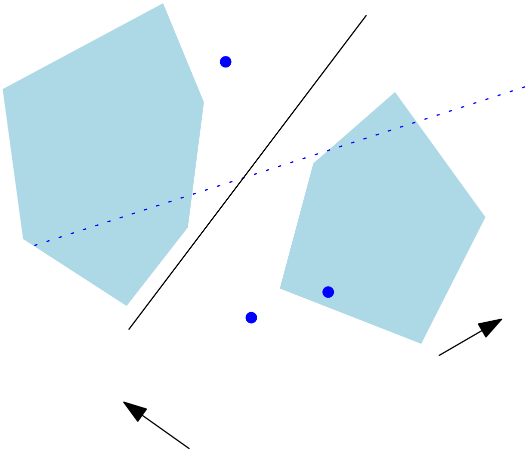
B's response



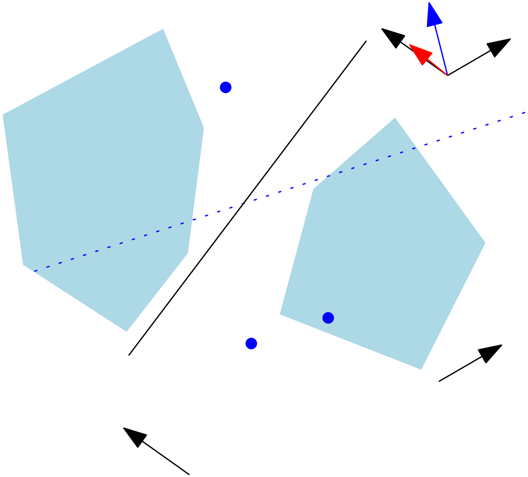
B's response



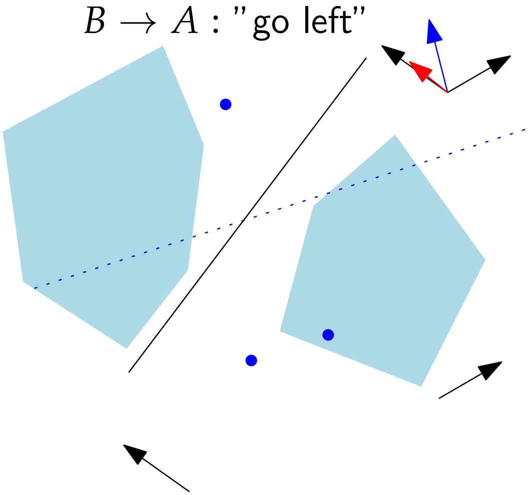
B's response



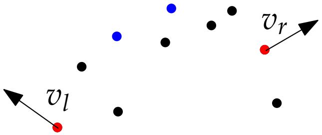
B's response



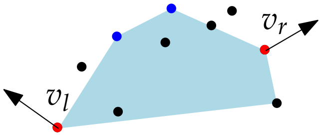
B's response



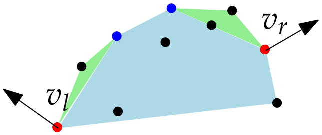
Back to A



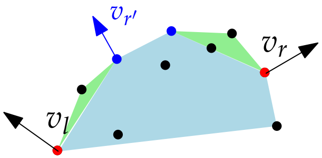
Back to A



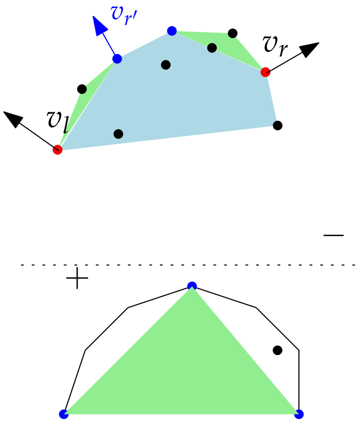
Back to A



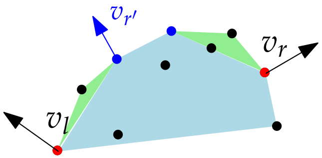
Back to A

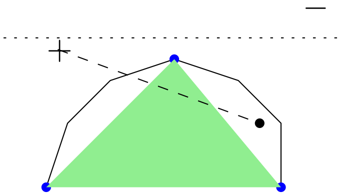


Back to A

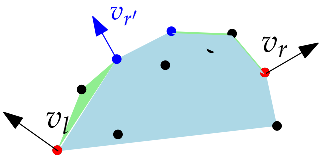


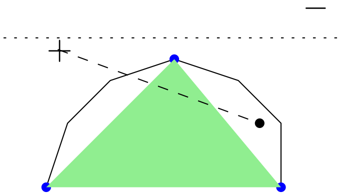
Back to A





Back to A





Summarizing

- In each round, each player exchanges a constant amount of information with the other.

Theorem

There exists a two-way protocol for two players that can compute an ϵ-error linear classifier for labelled points in $\mathbb{R}^2$ using $O(\log(1/\epsilon))$ bits of communication.

Summarizing

- In each round, each player exchanges a constant amount of information with the other.
- After each round, the number of points misclassified by a player reduces by a factor of 2.

Theorem

There exists a two-way protocol for two players that can compute an ϵ-error linear classifier for labelled points in $\mathbb{R}^2$ using $O(\log(1/\epsilon))$ *bits of communication.*

Summarizing

- In each round, each player exchanges a constant amount of information with the other.
- After each round, the number of points misclassified by a player reduces by a factor of 2.
- The number of misclassified points is at most n, and only needs to get below ϵn.

Theorem

There exists a two-way protocol for two players that can compute an ϵ-error linear classifier for labelled points in $\mathbb{R}^2$ using $O(\log(1/\epsilon))$ bits of communication.

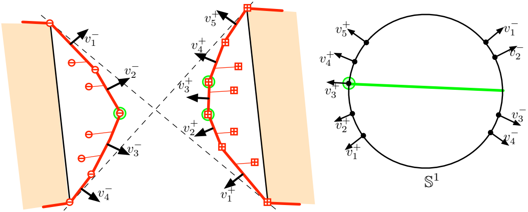
Summarizing

- In each round, each player exchanges a constant amount of information with the other.
- After each round, the number of points misclassified by a player reduces by a factor of 2.
- The number of misclassified points is at most n, and only needs to get below ϵn.
- Therefore, the number of rounds is $\log \frac{1}{\epsilon}$

Theorem

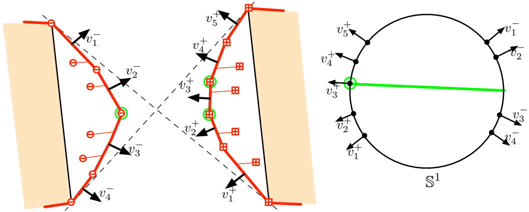
There exists a two-way protocol for two players that can compute an ϵ-error linear classifier for labelled points in $\mathbb{R}^2$ using $O(\log(1/\epsilon))$ bits of communication.

Other details



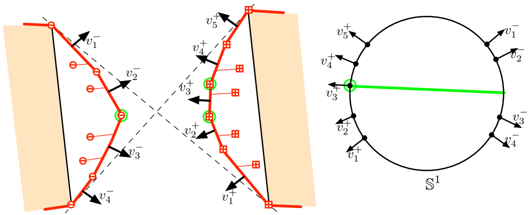
- A variant on the main argument gives us an algorithm to deal with both negative and positive examples in A simultaneously.

Other details



- A variant on the main argument gives us an algorithm to deal with both negative and positive examples in A simultaneously.
- By interleaving moves of A and parallel moves of B, we can get a single classifier that has at most ϵ error for both A and B.

Other details



- A variant on the main argument gives us an algorithm to deal with both negative and positive examples in A simultaneously.
- By interleaving moves of A and parallel moves of B, we can get a single classifier that has at most ϵ error for both A and B.
- If there are more than one player, simulate all pairwise interactions in $k^2 \log \frac{1}{\epsilon}$ communication.

Further directions I

- How do we extend this to higher dimensions ?
- What happens if the optimal classifier itself has nonzero error (the agnostic case)
- What about kernels ?

A general perspective on distributed learning

- Most machine learning problems reduce to some form of convex optimization
- Points become "constraints", and concepts ("hyperplanes") become points.

A general perspective on distributed learning

- Most machine learning problems reduce to some form of convex optimization
- Points become "constraints", and concepts ("hyperplanes") become points.

Problem

Suppose you have k players that each own a set of constraints $A_i x \leq b_i$? What is the communication needed to find a feasible point (or an optimal solution) for the LP

$$\max cx \ s.t \ A_i x \leq b_i$$