Machine Learning

CSCI 5622 Fall 2020

Prof. Claire Monteleoni

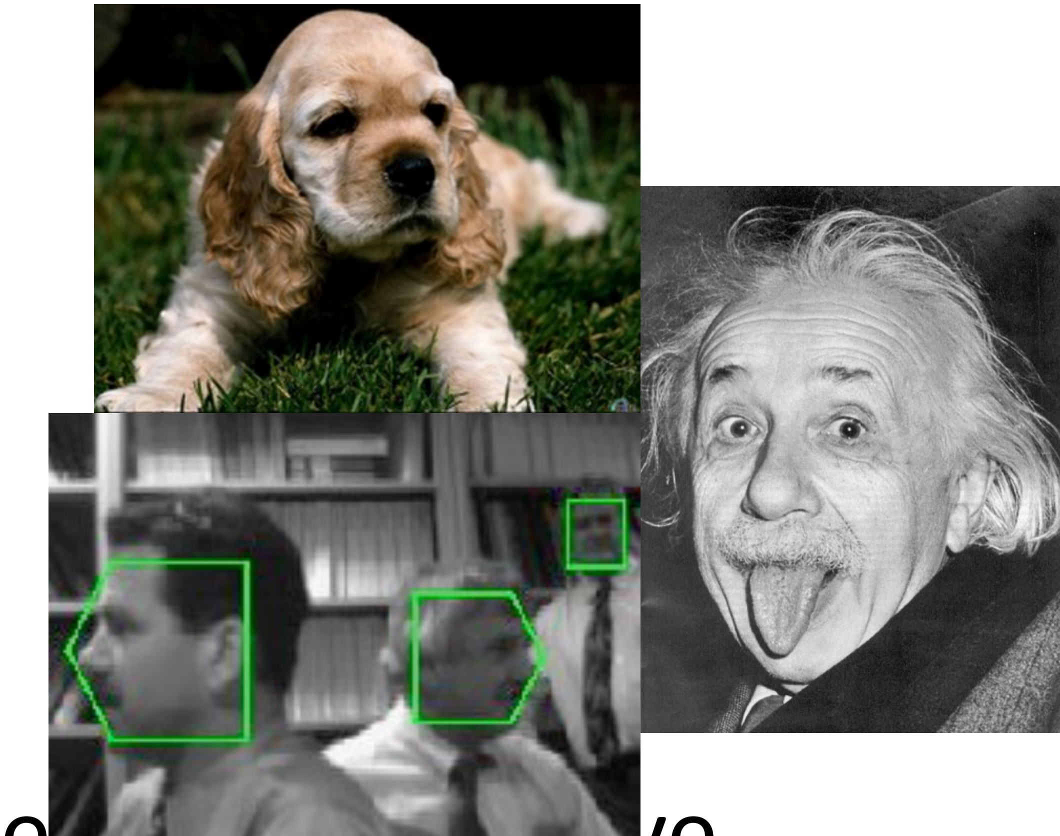
Today

- Intro. to Interactive Learning
  – Intro. to Active Learning (continued)

- Intro to Online Learning

With credit to S. Dasgupta

Active Learning



Many data-rich applications:

- Image/document classification
- Object detection/classification in video
- Speech recognition
- Analysis of sensor data

Unlabeled data is abundant, but labels are expensive.

Active Learning model: learner can pay for labels.

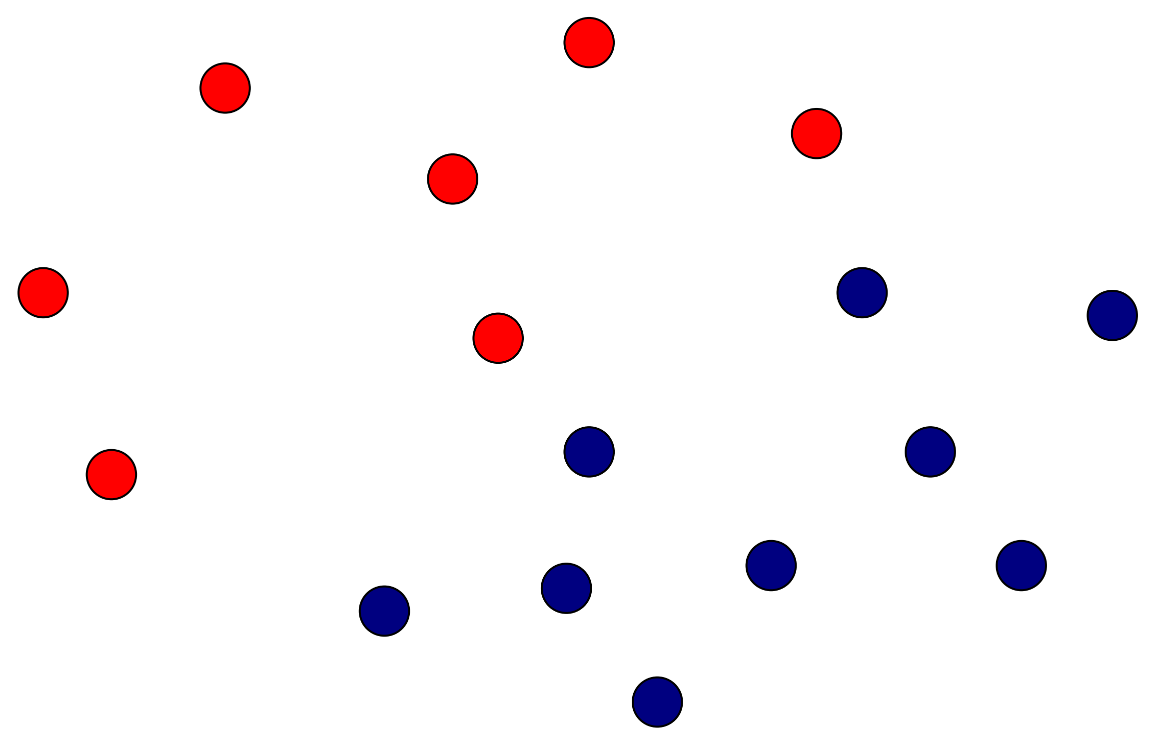
- Allows for intelligent choices of which examples to label.

Goal: given stream (or pool) of unlabeled data, use fewer labels to learn (to a fixed accuracy) than via supervised learning.

General field: Interactive Learning: learner interacts with teacher

Supervised learning

Given access to labeled data (drawn iid from an unknown underlying distribution P), want to learn a classifier chosen from hypothesis class H, with misclassification rate < ε.



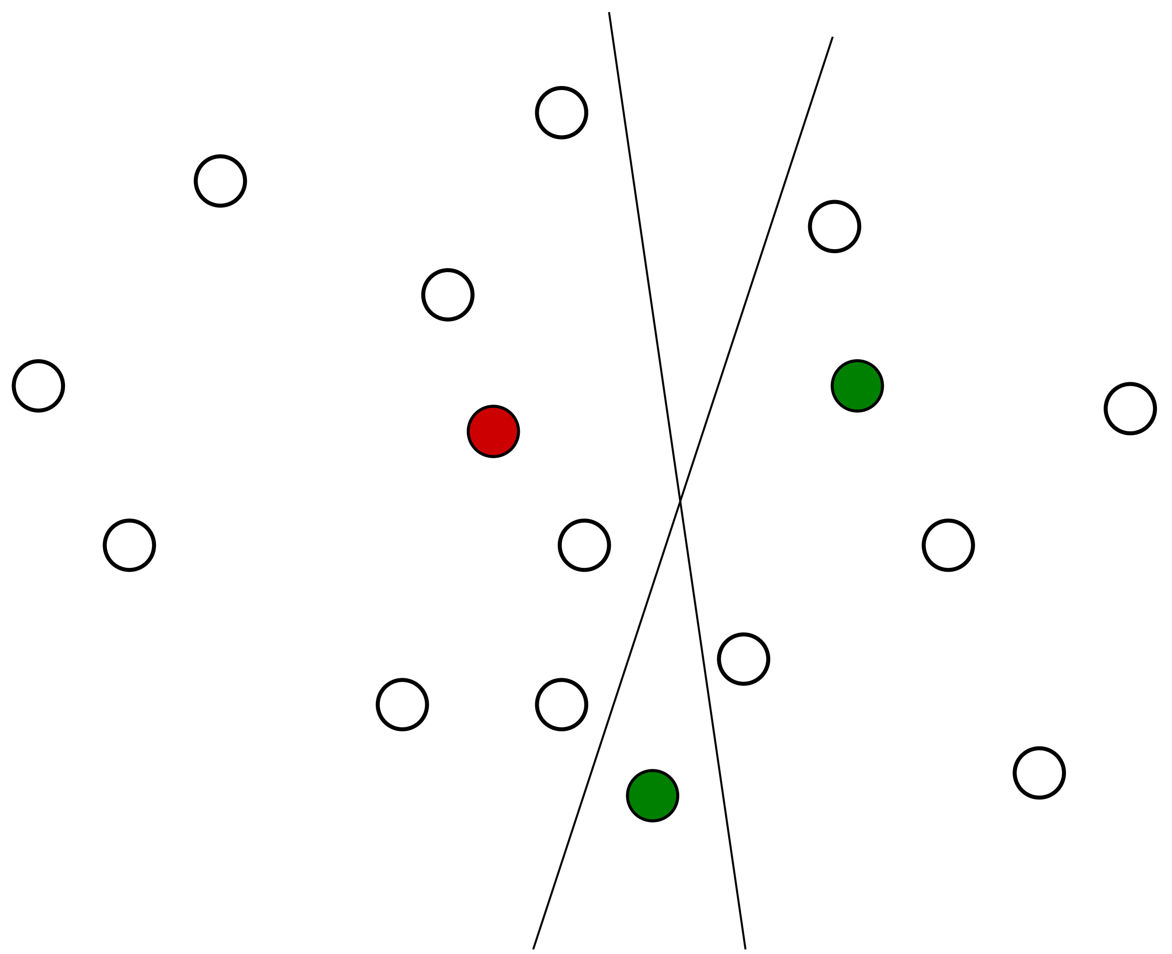
Sample complexity characterized by d = VC dimension of H.

If data is *separable*, need roughly d/ε labeled samples (PAC sample complexity).

Slide credit: Sanjoy Dasgupta

Active Learning

Given unlabeled data, choose which labels to buy, to attain a good classifier, at a low cost (in labels).



Label-complexity: What is the minimum number of labels needed to achieve the target error rate?

Active learning variants

There are several models of active learning:

- Query learning (a.k.a. Membership queries)
- Pool-based AL
- Active model selection
- Experiment design

Various evaluation frameworks:

- Regret minimization
- Minimize label-complexity to reach fixed error rate
- Label-efficiency (fixed label budget)

Membership queries

Early model of active learning in theory work [Angluin 1992]

X = space of possible inputs, e.g. R^n

H = class of hypotheses

Target concept h^* in H to be identified *exactly*.

You can ask for the label of any point in X: *no unlabeled data*.

$H_0 = H$

For $t = 1,2,\ldots$

    pick a point x in X and query its label $h^*(x)$

    let H_t = all hypotheses in H_{t-1} consistent with $(x, h^*(x))$

What is the minimum number of "membership queries" needed to reduce H to just $\{h^*\}$?

Slide credit: S. Dasgupta

Membership queries: problem

Many results in this framework, even for complicated hypothesis classes.

Problem: informative synthetic queries can be hard to label!

[Baum and Lang, 1991] tried fitting a neural net to handwritten characters.

Synthetic instances created were incomprehensible to humans!

[Lewis and Gale, 1992] tried training text classifiers.

"an artificial text created by a learning algorithm is unlikely to be a legitimate natural language expression, and probably would be uninterpretable by a human teacher."

Slide credit: S. Dasgupta

Pool-based active learning

Framework due to [Cohn, Atlas, Ladner, et al. NIPS '89]

Assume a fixed probability distribution, D over X x Y, X some input space, $Y = \{+1, -1\}$.

Given: stream (or pool) of unlabeled examples, x, drawn i.i.d. from marginal distribution, D_X over X.

Learner may request labels on examples in the stream.

- Oracle access to labels, y in $\{+1, -1\}$ from conditional at x, $D_{Y \mid x}$
- Constant cost per label.

The error rate of any classifier v is w.r.t. distribution D:

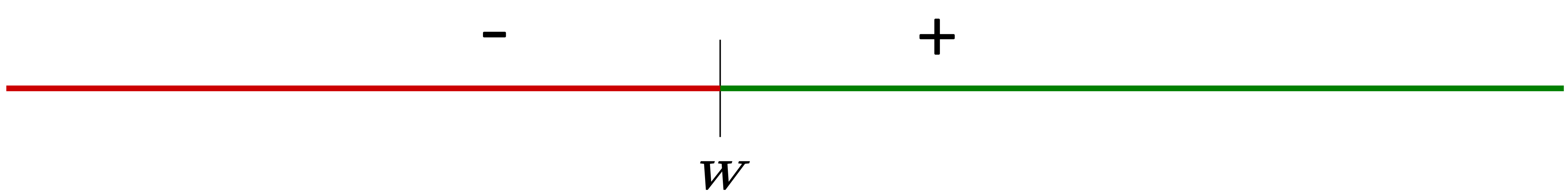
$$\mathrm{err}(h) = P_{(x,\, y)\sim D}[v(x) \neq y]$$

Goal: minimize number of *labels* to learn the concept (w.h.p.) to a fixed final error rate, ε, on input distribution.

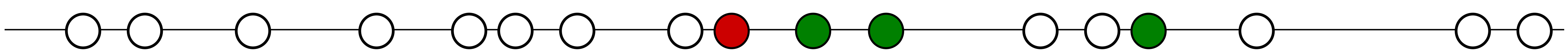
Can active learning really help?

[Cohn, Atlas & Ladner '94; Dasgupta '04]:
Threshold functions on the real line: $h_w(x) = \text{sign}(x - w)$, $H = \{h_w: w \text{ in } R\}$

Supervised learning: need $\Omega(1/\varepsilon)$ examples to reach error rate $< \varepsilon$.



Active learning: given $1/\varepsilon$ unlabeled points,



Binary search – need just $\log(1/\varepsilon)$ labels, from which the rest can be inferred! Exponential improvement in sample complexity.

→ However, many negative results, e.g. [Dasgupta '04], [Kääriäinen '06].

More general hypothesis classes

For a general hypothesis class with VC dimension d, is a "generalized binary search" possible?

Random choice of queries $\quad d/\varepsilon$ labels

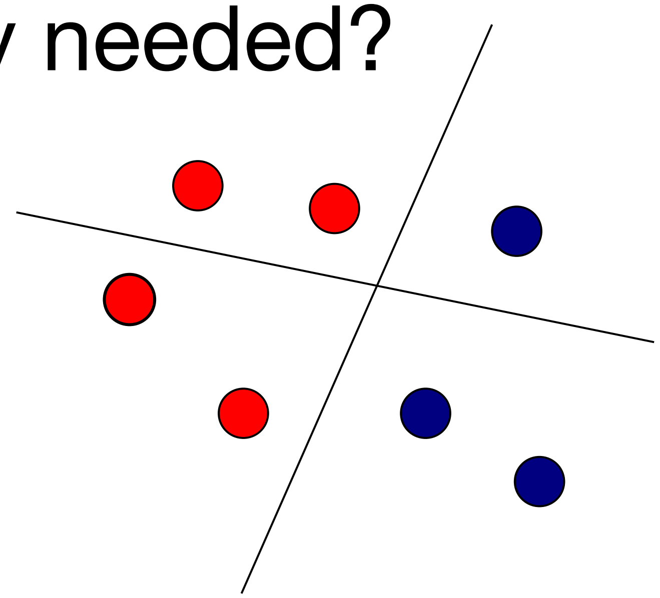
Perfect binary search $\quad d \log 1/\varepsilon$ labels

Where in this large range does the label complexity of active learning lie?

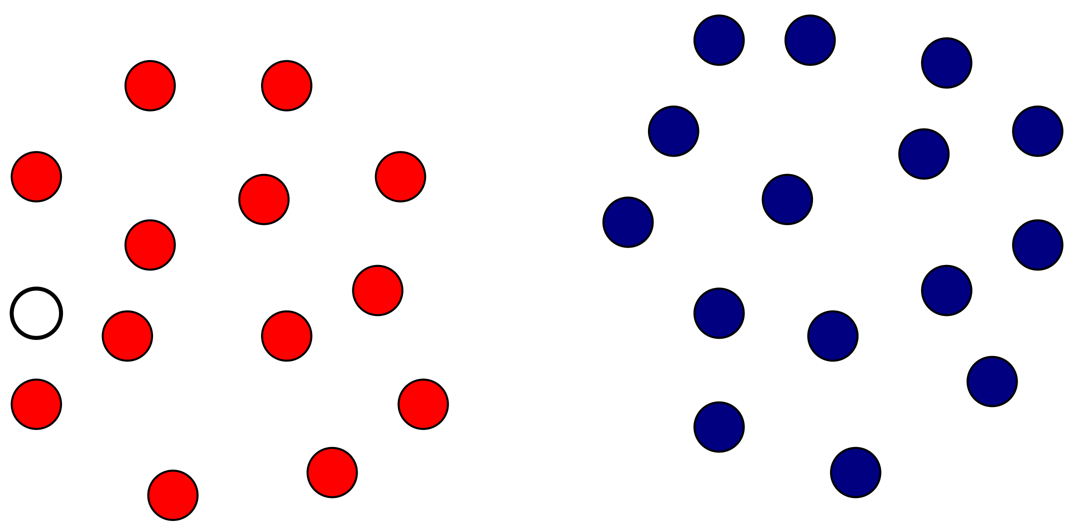
We've already handled linear separators in 1-d…

When is a label needed?

Is a label query needed?



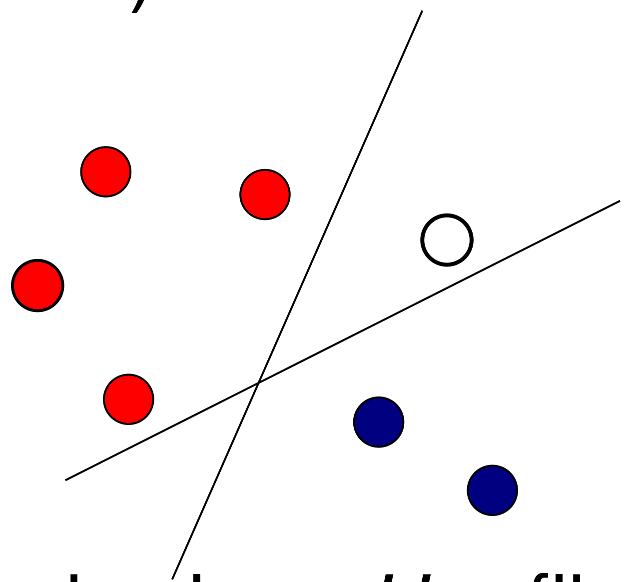
- Linearly separable case: NO
- There may not be a perfect linear separator: YES



- Either case: NO

Selective sampling algorithm

Region of uncertainty [CAL '94]: subset of data space for which there exist hypotheses (in *H*) consistent with all previous data, that disagree.



Example: hypothesis class, *H* = {linear separators}. Separable assumption.

Algorithm: Selective sampling [Cohn, Atlas & Ladner '94]:

For each point in the stream, if point falls in region of uncertainty, request label.

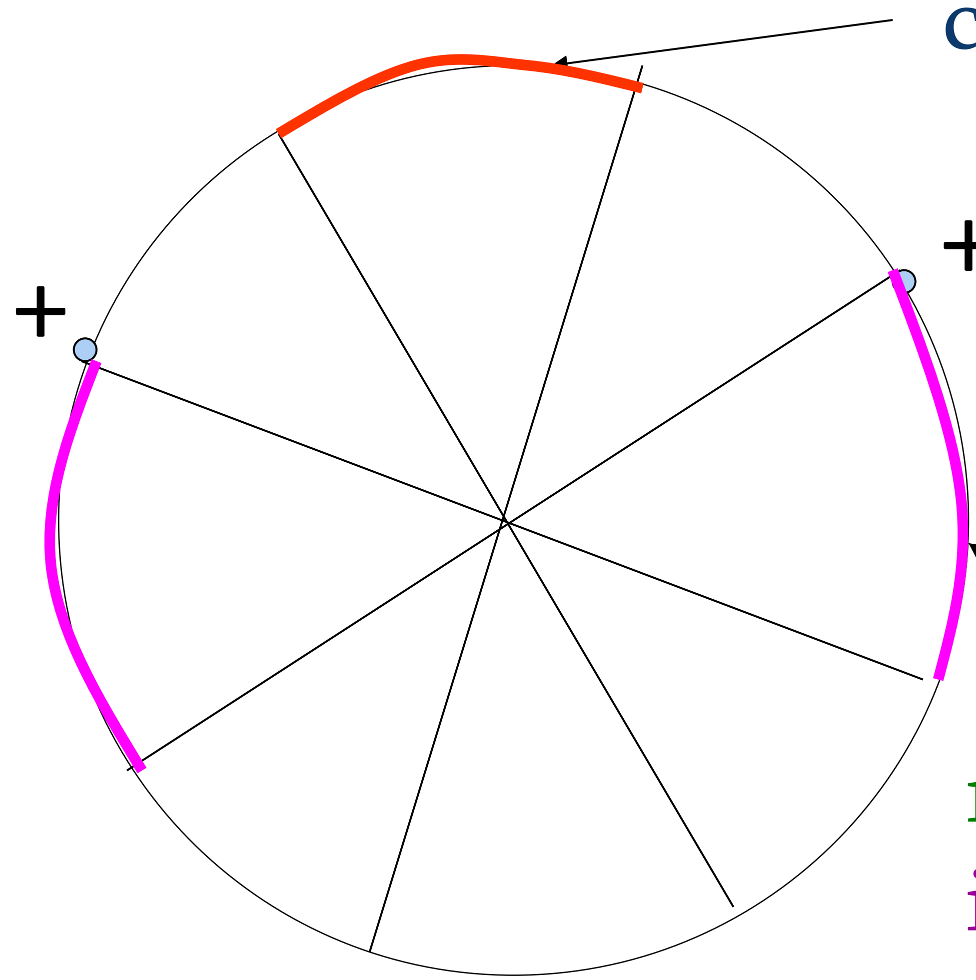
Region of uncertainty

Current version space: portion of H consistent with labels so far.

"Region of uncertainty" = part of data space about which there is still some uncertainty (ie. disagreement within version space)

Suppose data lies on circle in R^2; hypotheses are linear separators.

(spaces X, H superimposed)



current version space:

normal vecs to separators consistent with labeled points

region of uncertainty in data space

Slide credit: S. Dasgupta

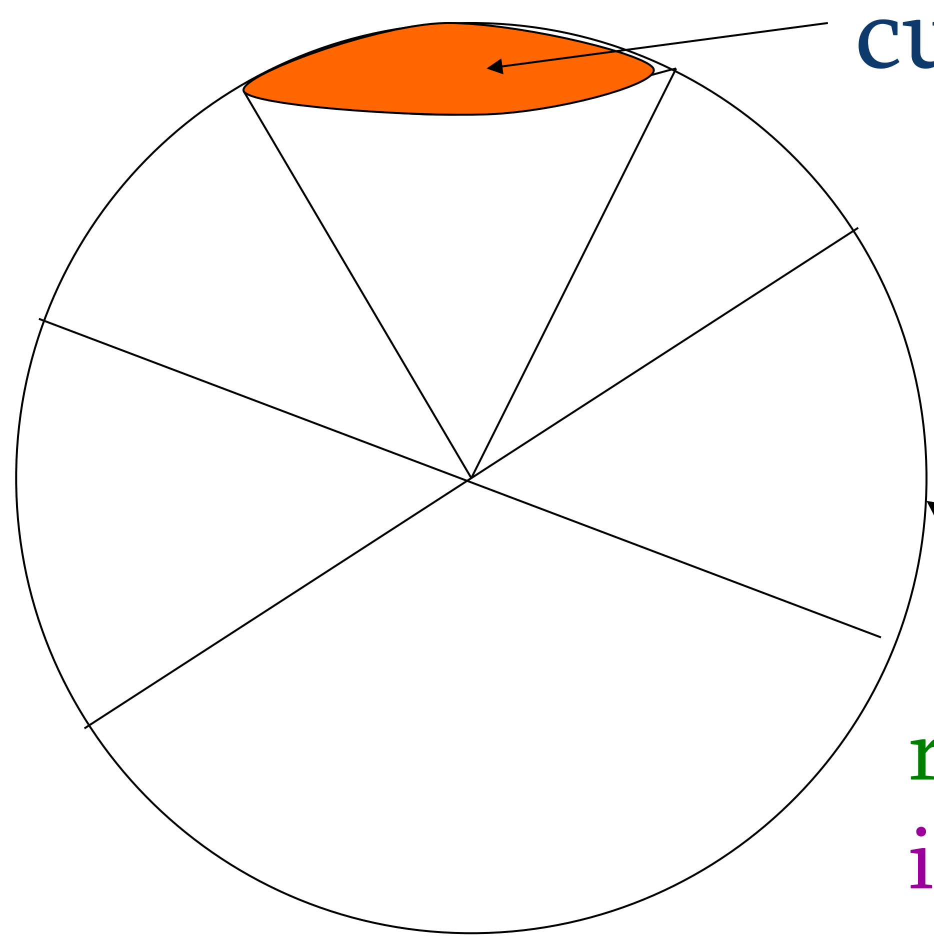
Region of uncertainty

Selective Sampling Algorithm [CAL+ 89]:

Of the unlabeled points which lie in the region of uncertainty, pick one at random and query its label.

Data and hypothesis spaces, superimposed:

(both are the surface of the unit sphere in R^d)



Slide credit: S. Dasgupta

Region of uncertainty

Number of labels needed depends on H and also on P.

Special case: H = {linear separators in R^d}, P = uniform distribution over unit sphere.

Theorem [Dasgupta, Hsu, & Monteleoni, NIPS '07]: $\tilde{O}(d \log 1/\varepsilon)$ labels suffice to reach a hypothesis with error rate $< \varepsilon$.

In contrast: supervised learning: $\Theta(d/\varepsilon)$ labels (PAC complexity).

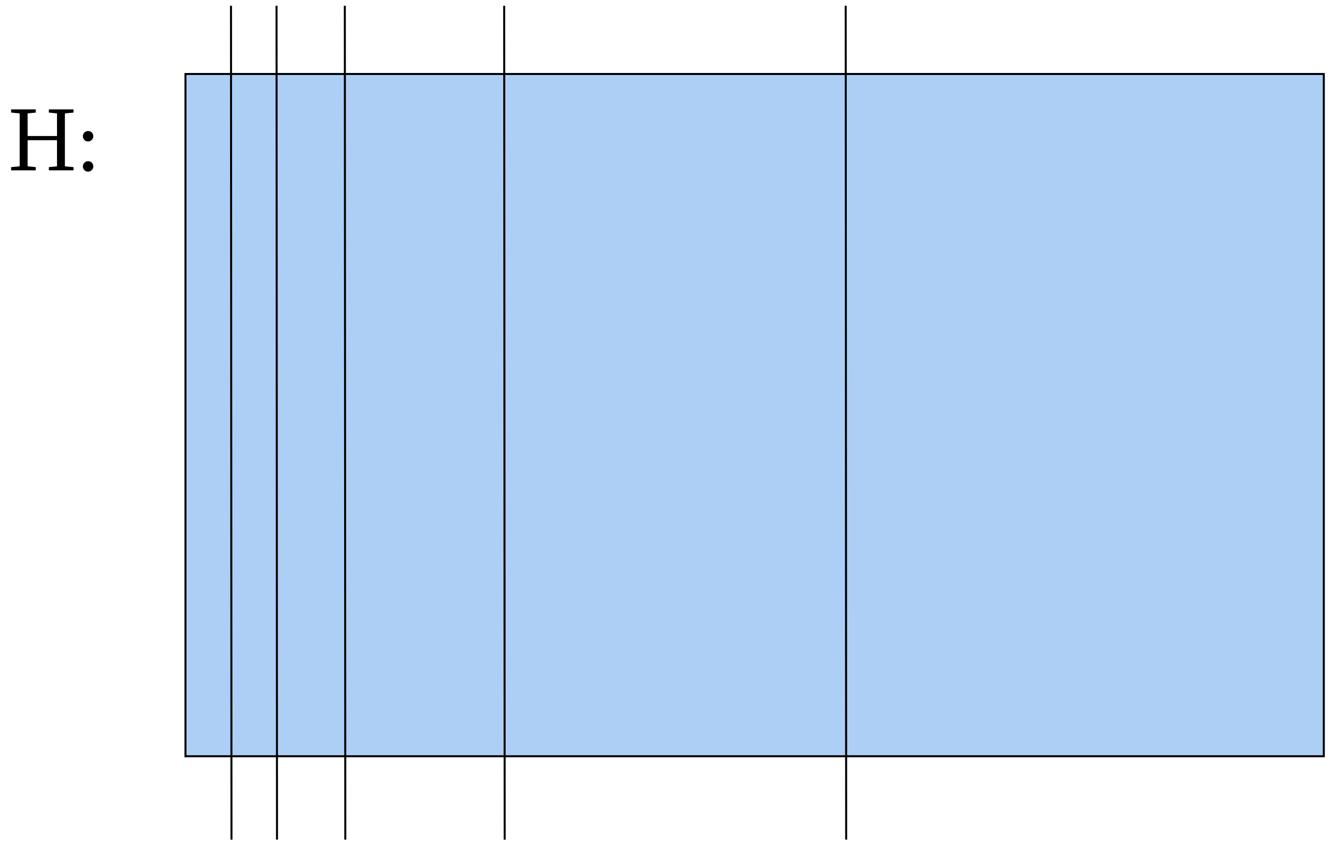
Slide credit: S. Dasgupta

Query-by-committee

First idea: Try to rapidly reduce volume of version space?

Problem: doesn't take data distribution into account.

To keep things simple, say d(h,h') ≈ Euclidean distance in this picture.



Error is likely to remain large!

Slide credit: S. Dasgupta

Query-by-committee

[Seung, Opper, Sompolinsky, 1992; Freund, Seung, Shamir, Tishby 1997]

First idea: Try to rapidly reduce volume of version space?

Problem: doesn't take data distribution into account.

H:

Which pair of hypotheses is closest? Depends on data distribution P.

Distance measure on H: $d(h,h') = P[h(x) \neq h'(x)]$

Slide credit: S. Dasgupta

Query-by-committee

Elegant scheme which decreases volume in a manner which is sensitive to the data distribution.

Bayesian setting: given a prior π on H

$H_1 = H$

For t = 1, 2,

Receive an unlabeled point x_t drawn from P

Choose two hypotheses h,h' randomly (i.i.d.) from (π, H_t)

If $h(x_t) \neq h'(x_t)$: ask for x_t's label

H_{t+1}= all hypotheses in H_t consistent with x_t and label

Else $H_{t+1} = H_t$

Slide credit: S. Dasgupta

Query-by-committee

For t = 1, 2, …

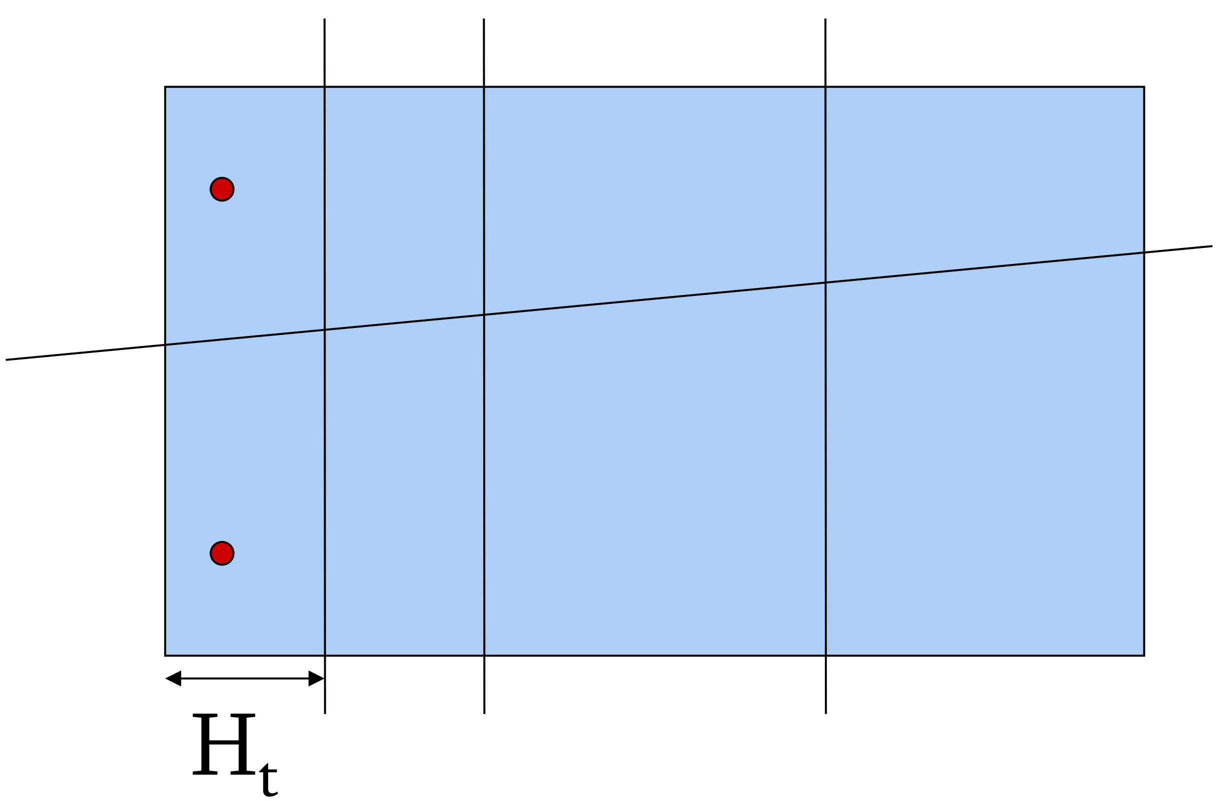
receive an unlabeled point x_t drawn from P

choose two hypotheses h,h' randomly from (π, H_t)

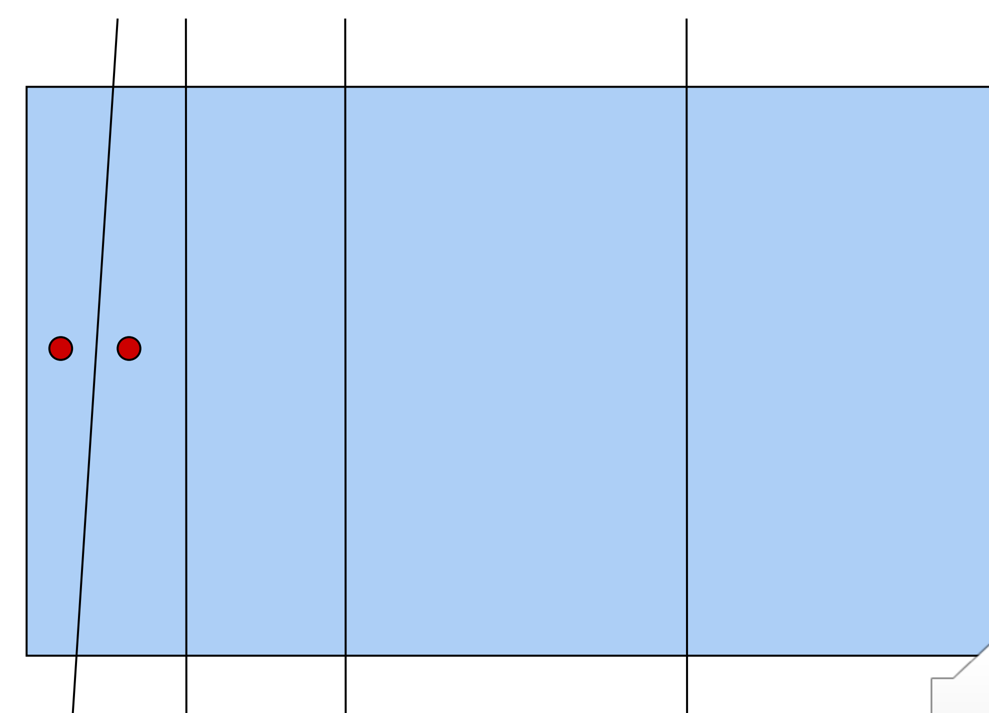
if $h(x_t) \neq h'(x_t)$: ask for x_t's label

set H_{t+1}

Observation: the probability of getting pair (h,h') that leads to a label query is proportional to $\pi(h)\ \pi(h')\ d(h,h')$.



vs.



Slide credit: S. Dasgupta

Query-by-committee

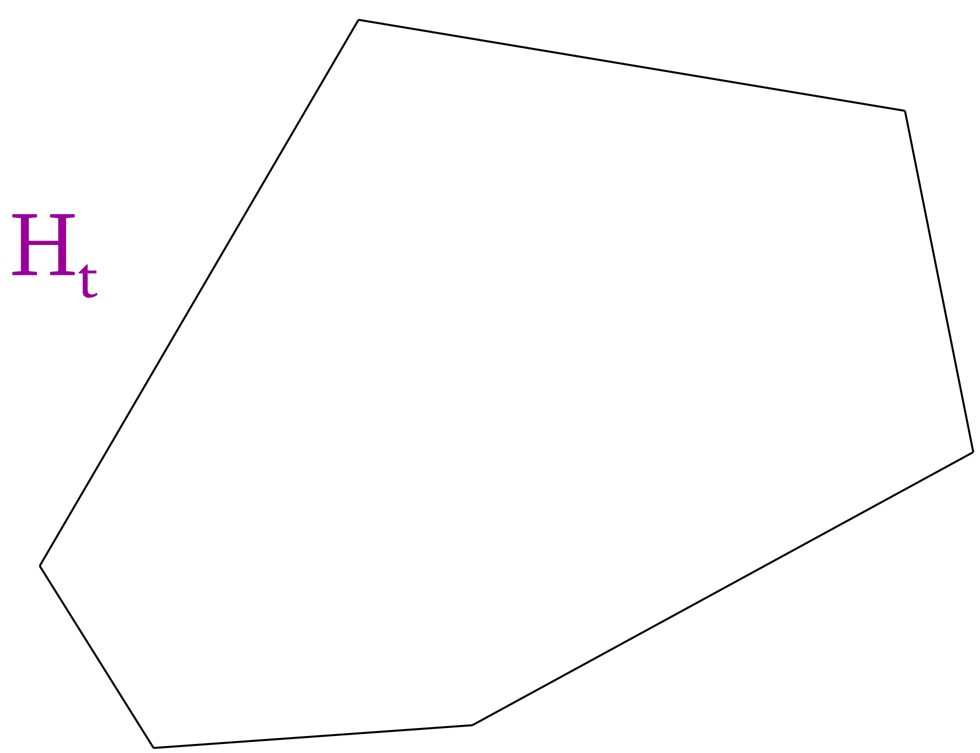
Label bound, Theorem [FSST97] :

For $H = \{\text{linear separators in } R^d\}$, P = uniform distribution, then

$\tilde{O}(d \log 1/\varepsilon)$ labels suffice to reach a hypothesis with error $< \varepsilon$.

Implementation: need to randomly pick h according to (π, H_t).

e.g. $H = \{\text{linear separators in } R^d\}$, π = uniform distribution:



How do you pick a random point from a convex body? (Difficult)

Random walk techniques, see a practical variant: [Gilad-Bachrach, Navot & Tishby NIPS '05]

Slide credit: S. Dasgupta

Active Learning

For linear separators in high dimension, is a generalized binary search possible, allowing exponential label savings?

YES!

[Dasgupta, Kalai & Monteleoni, JMLR 2009 (COLT 2005)]: Online active learning with exponential error convergence.

Theorem. There exists an online active learning algorithm that converges to generalization error □□after Õ(d log 1/□) labels.

Corollary. The total errors (labeled and unlabeled) will be at most Õ(d log 1/□).

Active Learning

In general, is it possible to reduce active learning to supervised learning?

YES!

[Monteleoni, Open Problem, COLT 2006]: Goal: general, efficient active learning.

[Dasgupta, Hsu & Monteleoni, NIPS 2007]: General active learning via reduction to supervised learning.

Problem: efficient, general AL

[Monteleoni, Open Problem, COLT 2006]

Efficient algorithms for active learning under general input distributions, *D*.

→ Previous label complexity upper bounds for general distributions are based on *intractable* schemes!

Provide an algorithm such that w.h.p.:

1. After *L* label queries, algorithm's hypothesis *v* obeys:

   $$P_{x \sim D_X}[v(x) \neq u(x)] < \varepsilon.$$

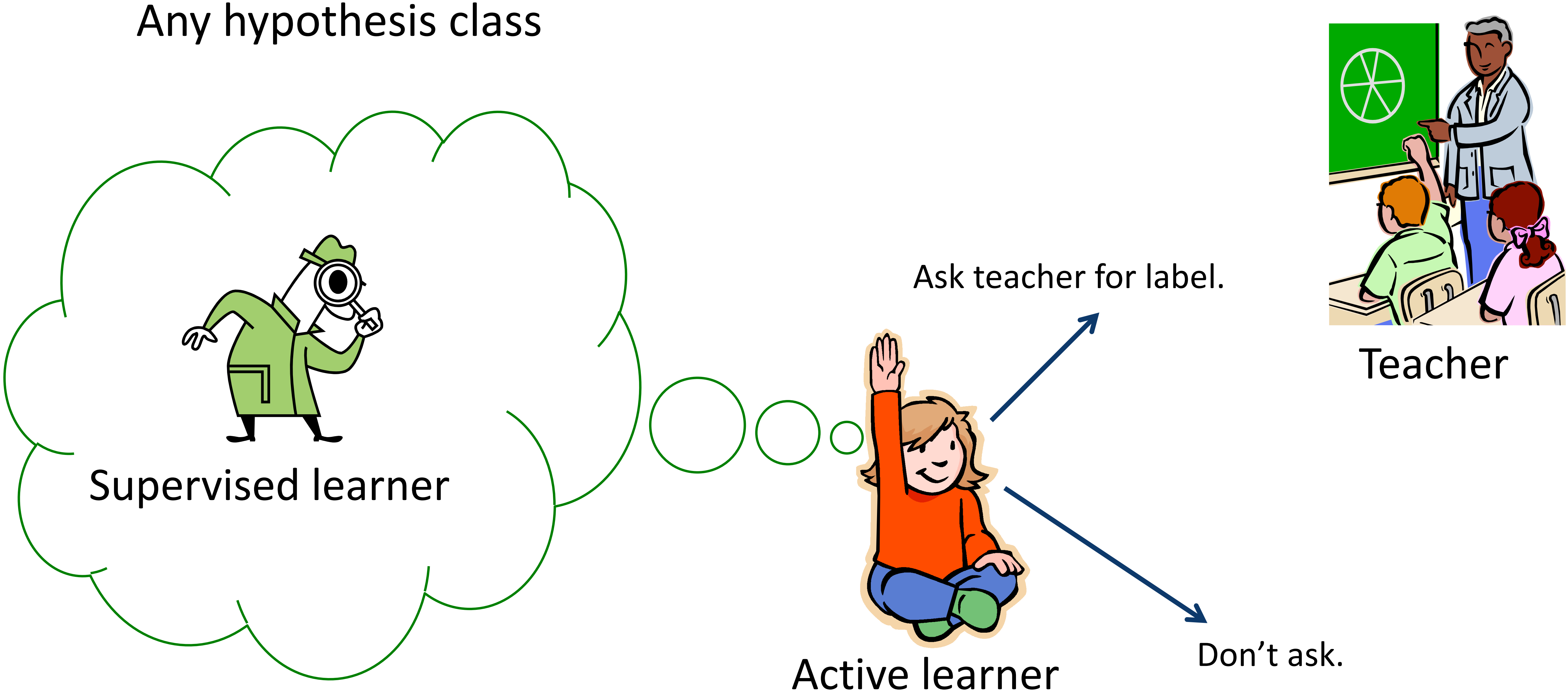
2. *L* is at most the supervised sample complexity, and for a general class of input distributions, *L* is significantly lower.
3. Running time is at most *poly*(d, 1/ε).

→ Was open even for specific hypothesis classes, batch case!

General active learning via reduction

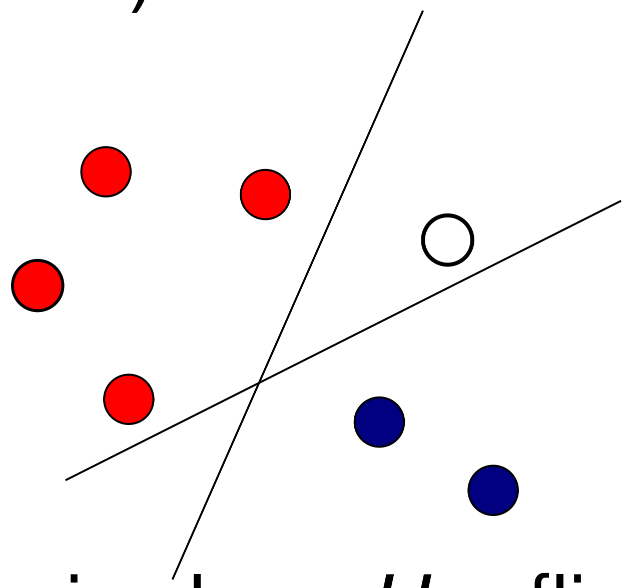
First reduction from active learning to supervised learning.

- Any data distribution (including arbitrary noise)
- Any hypothesis class



Selective sampling algorithm

Region of uncertainty [CAL '94]: subset of data space for which there exist hypotheses (in *H*) consistent with all previous data, that disagree.



Example: hypothesis class, *H* = {linear separators}. Separable assumption.

Algorithm: Selective sampling [Cohn, Atlas & Ladner '94] (orig. NIPS 1989):

For each point in the stream, if point falls in region of uncertainty, request label.

Easy to represent the region of uncertainty for certain, separable problems. BUT, in this work we address:

- What if data is not separable?
- General hypothesis classes?

} → Reduction!

General active learning via reduction

[Dasgupta, Hsu & Monteleoni, NIPS 2007]

First positive step towards answering [M, Open Problem, COLT 2006]:
general active learning: arbitrary input distribution and hypothesis class.

Technique: reduce to supervised learning. Call a supervised learner twice to determine whether a current unlabeled point is "uncertain." Only request labels on uncertain points.

Performance guarantees:

Upper bounds on label complexity:

- Never worse than supervised (PAC) sample complexity.
- Exponential savings for families of distributions/problems.

Consistency: algorithm's error converges to optimal.

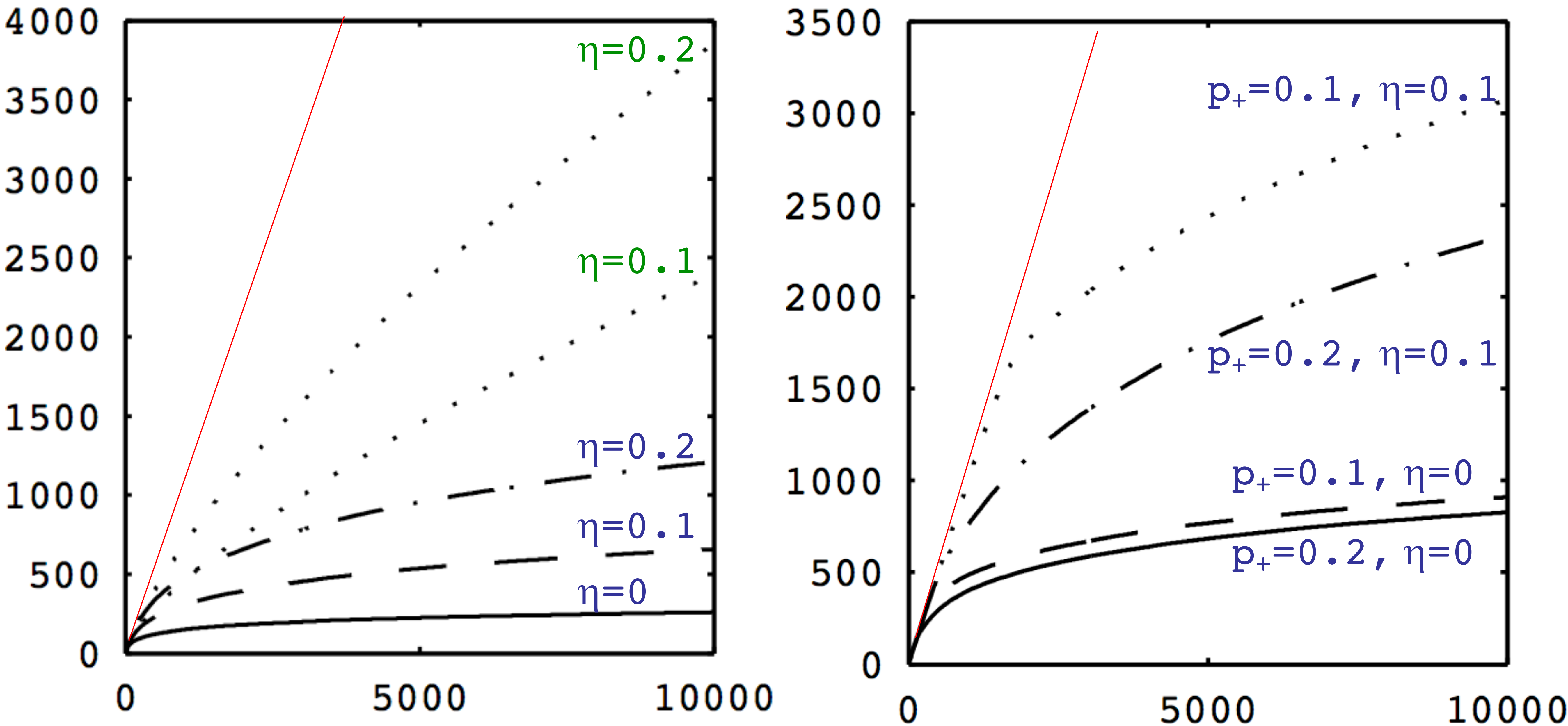
Efficiency: running time is at most (up to polynomial factors) that of supervised learning algorithm for the problem.

Active learning via reduction: experiments

Hypothesis classes in R^1:

Thresholds: $h^*(x) = \text{sign}(x - 0.5)$ Intervals: $h^*(x) = I(x \text{ in } [\text{low}, \text{high}])$

$p_+ = P_{x \sim D_X}[h^*(x) = +1]$



Number of label queries versus points received in stream.

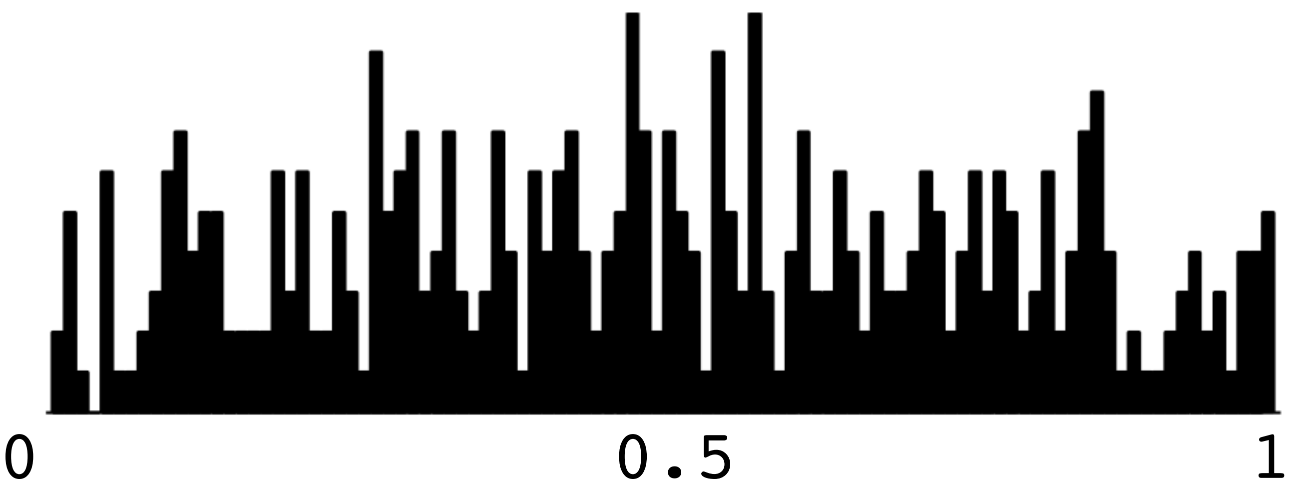
Red: supervised learning. Blue: random misclassification, Green: Tsybakov boundary noise

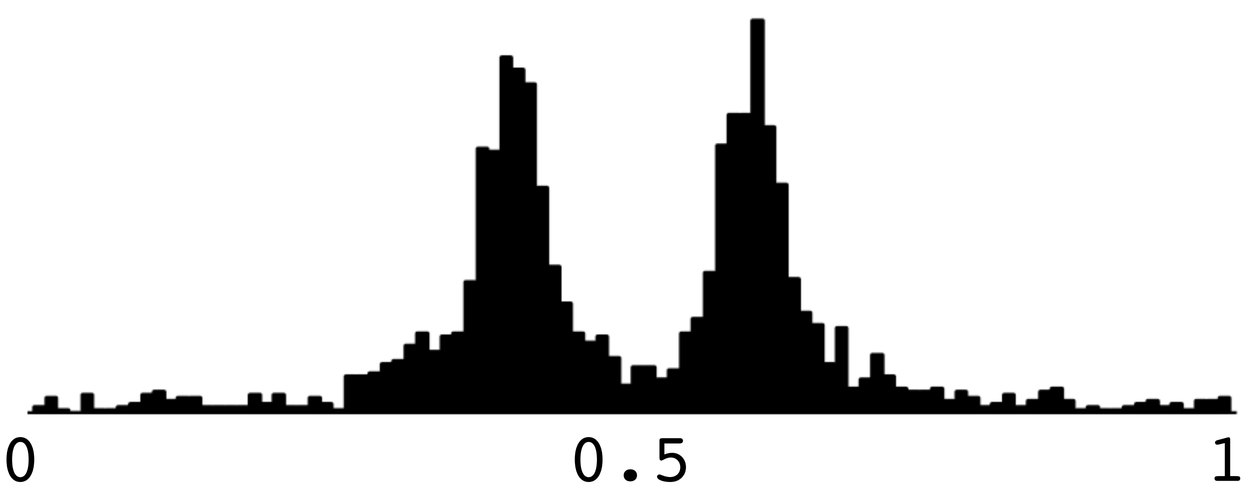
Experiments

Interval in $\mathbb{R}^1$:

$$h^*(x) = \mathbb{1}(x \text{ in } [0.4, 0.6])$$

Label queries: 1-400:

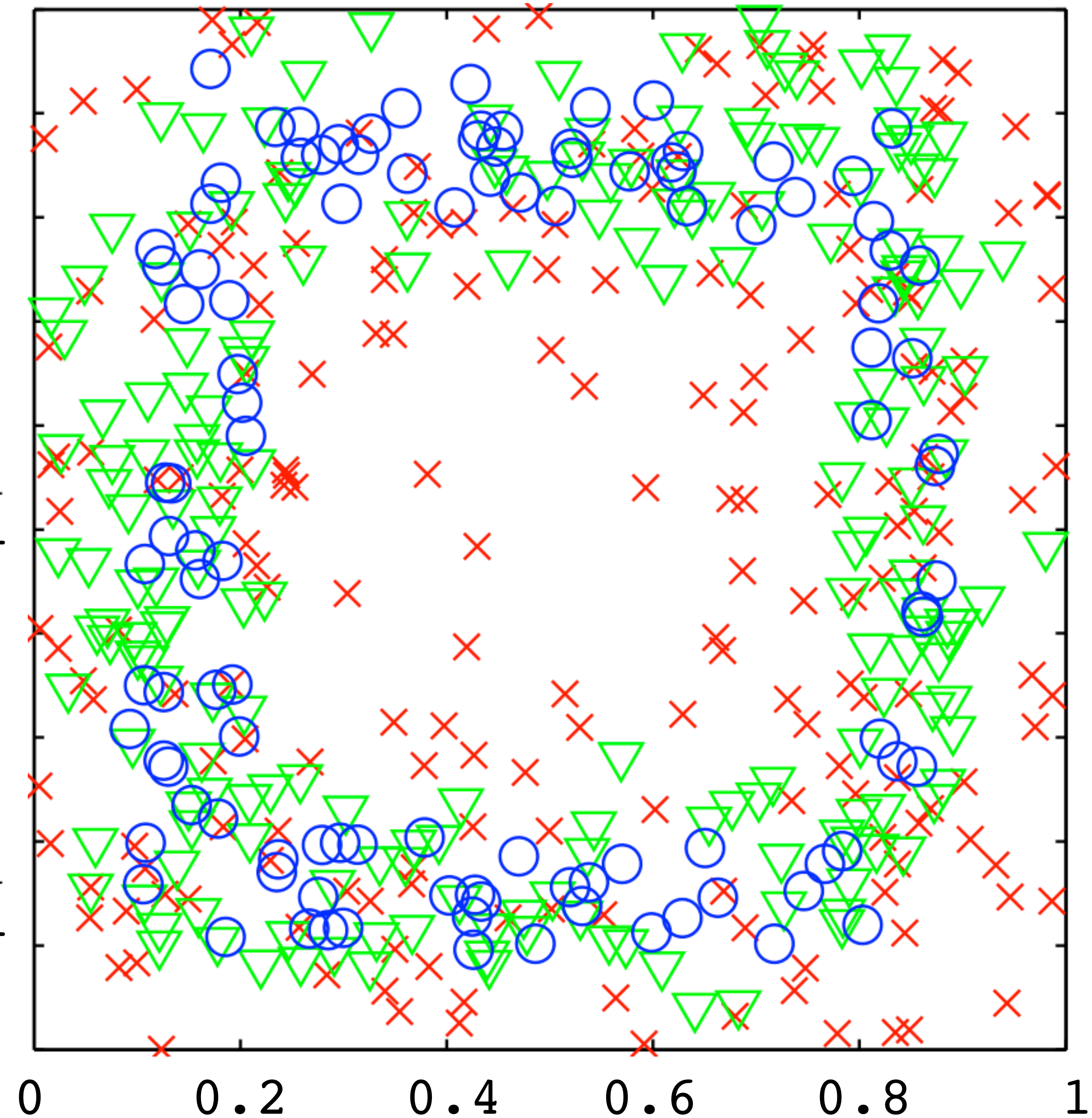




All label queries (1-2141).

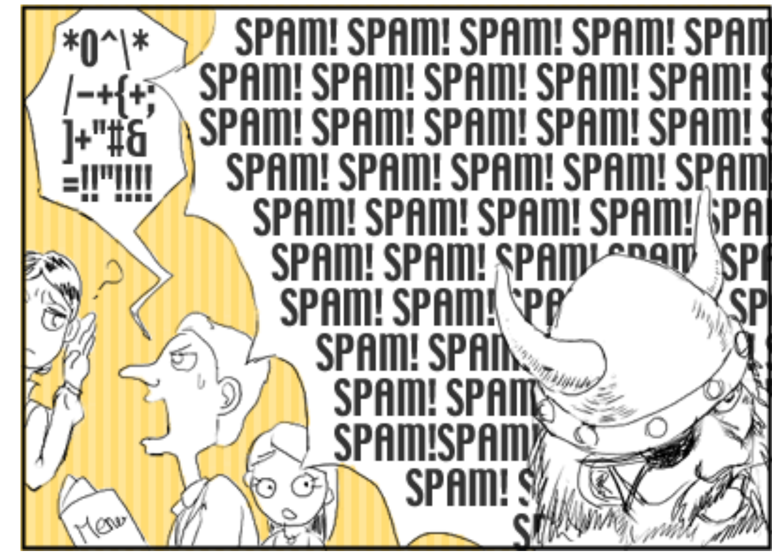
Interval in $\mathbb{R}^2$ (Axis-parallel boxes):

$$h^*(x) = \mathbb{1}(x \text{ in } [0.15, 0.85]^2)$$



Temporal breakdown of label request locations. Queries: 1-200, 201-400, 401-509.

Online active learning: motivations



Data-rich applications:

Spam filtering, e.g. [Sculley CEAS 2007]

Image/webpage relevance filtering

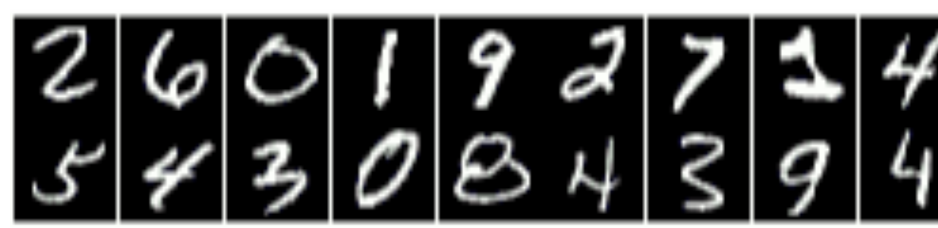
Object detection in video

Resource-constrained applications:

Interactive learning on sensors, mobile robots.

Human-interactive learning on small devices:

e.g. OCR on handhelds used by doctors.



Active learning scenario:

User writes characters.

Device occasionally queries a label (user must type into keypad). Human users likely prefer fewer interruptions (label queries).

Online active learning: OCR application

[M & Kääriäinen CVPR workshop ‘07]

We apply online active learning to OCR due to its potential efficacy for OCR on small devices:

- Scenario: user writes characters.
- Device occasionally queries a label (user must type into keypad).
- Human users likely prefer fewer interruptions (label queries).

OCR data highly non-uniform: test [DKM’05/’09] algorithm when relax distributional and separability assumptions.

Algorithms and evaluation

[Cesa-Bianchi,Gentile & Zaniboni '06] algorithm (parameter b):

- Filtering rule: flip a coin w.p. $b/(b + |x \cdot v_t|)$
- Update rule: standard Perceptron.
- Relative bounds on error w.r.t. best linear classifier (regret, non i.i.d.).
- Fraction of labels queried depends on b.

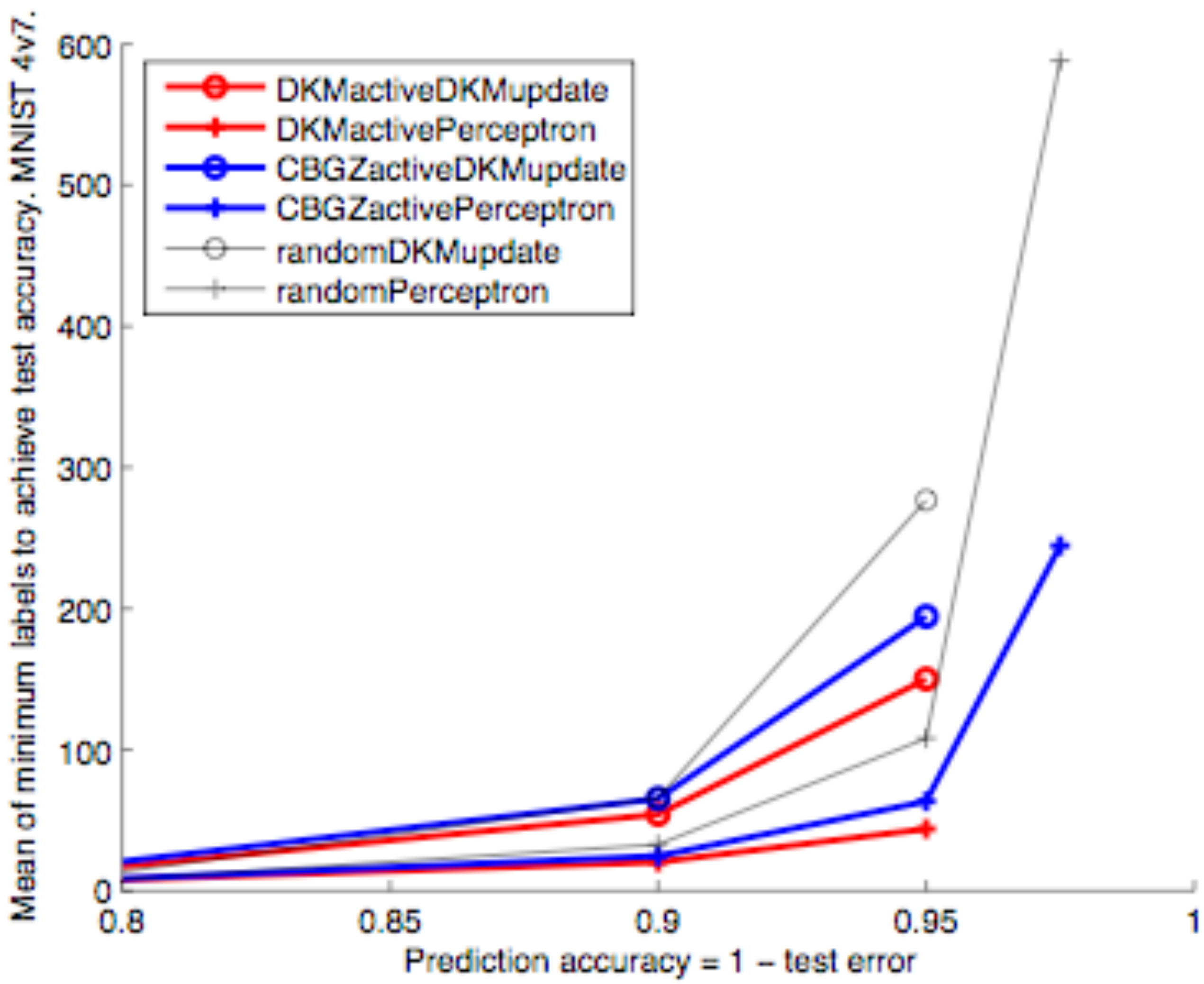
Experiments with all 6 combinations of:

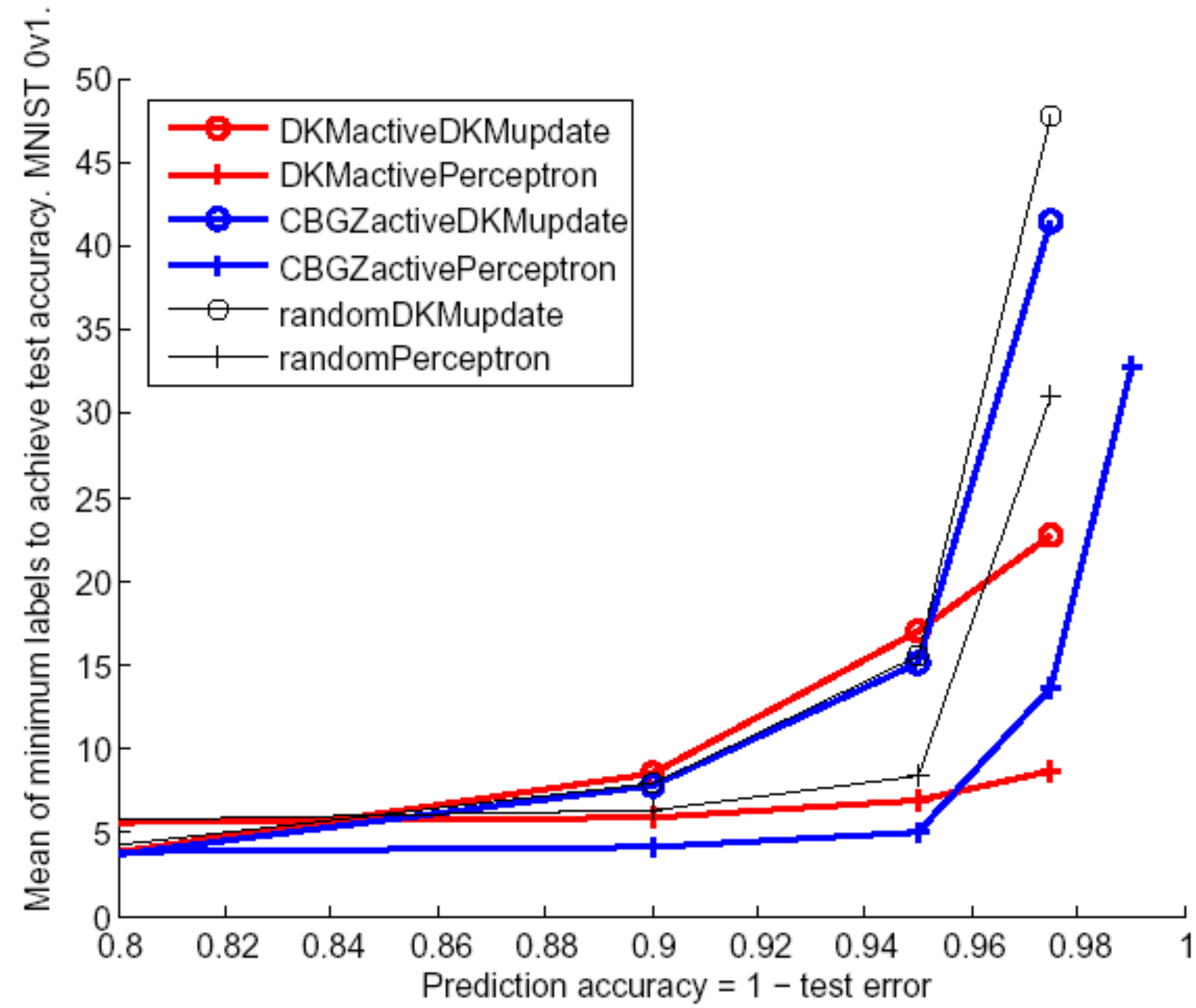
- Update rule in {Perceptron, DKM modified Perceptron}
- Active learning logic in {DKM, C-BGZ, random}

MNIST (d=784) and USPS (d=256) OCR data.

7 problems, with approx 10,000 examples each.

5 random restarts of 10-fold cross-validation.





Active learning always quite outperformed random sampling:

> Random sampling perc. used 1.26–6.08x as many labels as active.
>
> Factor was at least 2 for more than half of the problems.

DKMperceptron performed best overall, followed by DKM.

Superior supervised learning sub-algorithm: Perceptron.

Superior active learning rule: DKM.

Online active learning framework

Pool-based framework of [Cohn,Atlas&Ladner'89]

Assume a fixed probability distribution, D over X x Y, X some input space, $Y = \{\pm 1\}$.

Given: stream (or pool) of unlabeled examples, x, drawn i.i.d. from marginal distribution, D_X over X.

Learner may request labels on examples in the stream.

Oracle access to labels, y in $\{\pm 1\}$ from conditional at x, $D_{Y \mid x}$

Constant cost per label.

The error rate of any classifier v is measured on distribution D:

$$\mathrm{err}(v) = P_{(x,\, y)\sim D}[v(x) \neq y]$$

Goal: minimize number of *labels* to learn the concept (whp) to a fixed final error rate, ε, on input distribution.

Must respect online constraints on time and memory.

Measures of complexity

PAC sample complexity:

Supervised setting: number of (labeled) examples, sampled iid from *D*, to reach error rate ε.

Mistake-complexity:

Supervised setting: number of mistakes to reach error rate ε.

Label-complexity:

Active setting: number of label queries to reach error rate ε.

Error complexity:

Total prediction errors made on (labeled and/or unlabeled) examples, before reaching error rate ε.

Supervised setting: equal to mistake-complexity.

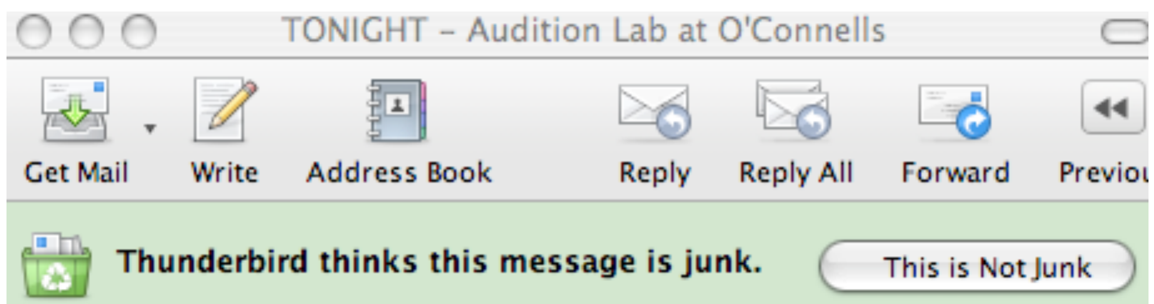
Active setting: mistakes are a subset of total errors on which learner queries a label.

Today

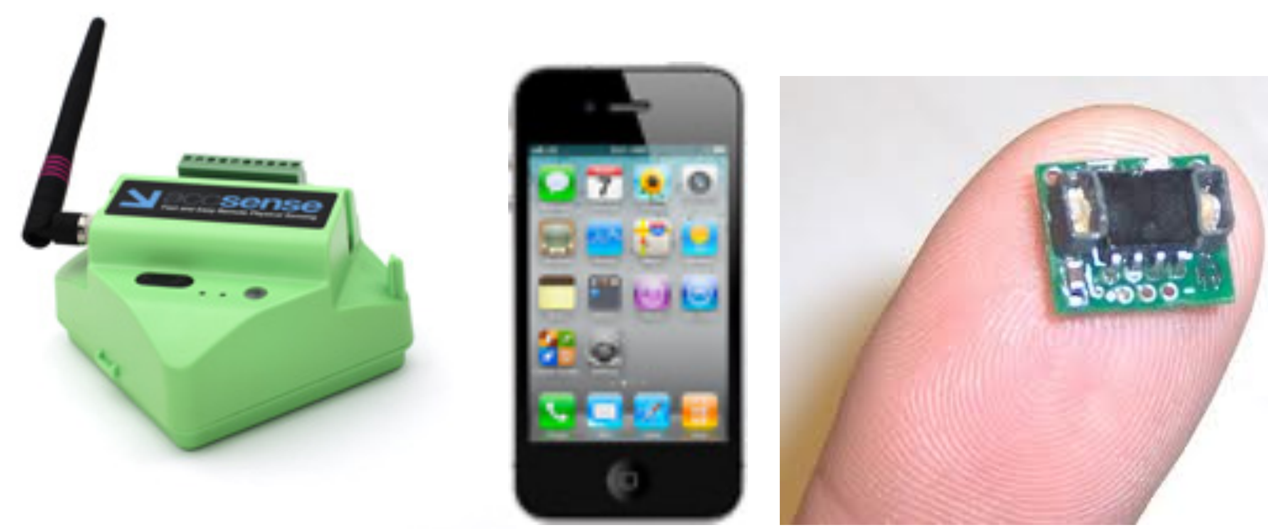
- Learning Theory
  - Online learning

Learning from data streams

Forecasting, real-time decision making, streaming data applications,



online classification,

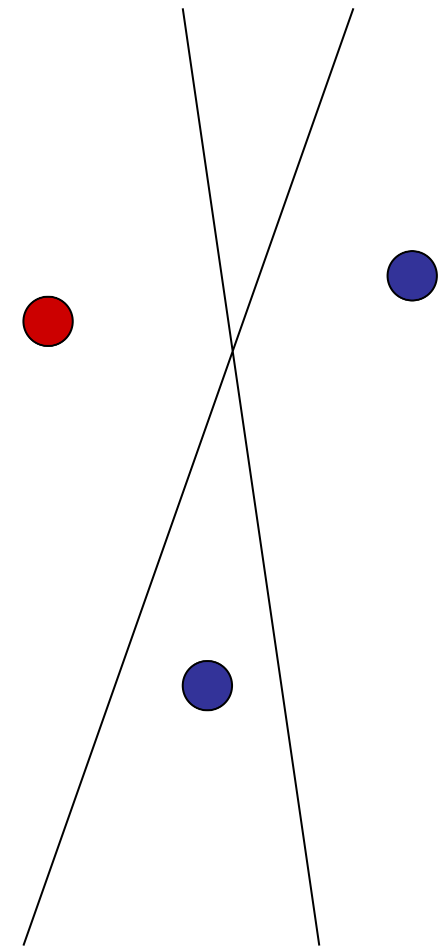


resource-constrained learning.

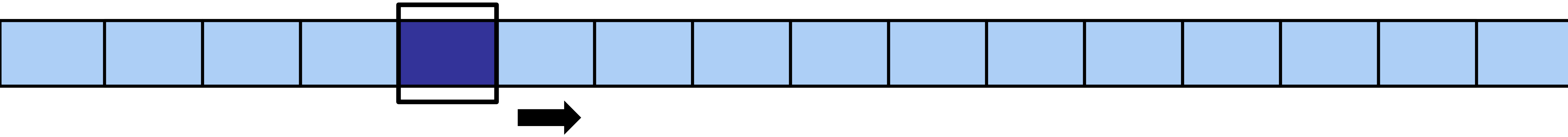
Learning from data streams

Data arrives in a stream over time.

E.g. linear classifiers:



Learning from data streams



1. Access to the data observations is one-at-a-time.

- Once a data point has been observed, it might never be seen again.
- Optional: Learner makes a prediction on each observation.

→ Models forecasting, real-time decision making, high-dimensional, streaming data applications.

2. Time and memory usage must not grow with data.

- Algorithms may not store all previously seen data and perform batch learning.

→ Models resource-constrained learning.

General framework of online learning

Learning proceeds in stages, as data points arrive in a stream.

At each stage learner receives labeled point (x, y), x from input space X, y from label space Y.

Learner first observes only x, and makes prediction v(x), where v is current hypothesis.

Then learner observes y, and can update its hypothesis v, usually based on prediction loss L(v(x), y), appropriate to problem.

Topics

Canonical online learning

Additive updates (e.g. standard Perceptron)

Multiplicative updates (e.g. Winnow)

Some online learning algorithms:

- A modification of Perceptron in which the angle decreases monotonically → monotonic error decrease when data is Uniform

- Algorithms descended from Winnow, for learning from time-varying data streams:
  - Tracking the best expert
  - Tracking shifting experts
    - Fixed-share
    - Learn-α

Multiplicative updates

Canonical algorithms:

- Halving Algorithm of [Angluin, ‘88].
- Winnow Algorithm due to [Littlestone, ‘88].
- Weighted Majority Algorithm [Littlestone & Warmuth, ’89]

- Algorithms descended from these for learning from time-varying data streams:
    - Tracking the best expert
    - Tracking shifting experts
        - Fixed-share
        - Learn-α

Halving algorithm

- H is the set of all possible hypotheses (classifiers)
- We are in the binary classification setting, with 0-1 loss.

Algorithm: Halving(H)

[on white board]

Halving algorithm: mistake bound

Theorem: In the realizable case (i.e. there exists h* in H, such that h* has zero true error), $M_{alg} \leq \log_2|H|$.

- The algorithm makes predictions using the majority vote, i.e. the vote corresponding to at least half of the experts.
- So if the algorithm's prediction is a mistake, at least half of the experts are wrong.
- The algorithm then removes these experts from the active set.
- Therefore, each mistake reduces the size of the active set by at least a factor of ½.
- Therefore, after $\log_2|H|$ mistakes there can remain only one active hypothesis.
- The problem is realizable, so h* will never make a mistake, and so it must be the remaining one in the active set. And recall, it has 0 error.

Simple example: learning an OR f'n

- Suppose features are boolean: $X = \{0,1\}^d$.
- Target is an OR function, like $x_3 \vee x_9 \vee x_{12}$, with no noise.
- Can we find an on-line strategy and bound its mistakes?

- Labeled examples are of the form
  - 0 1 0 1 0 0 0, +1
  - 0 0 0 0 0 0 1, -1
  - 0 1 0 0 0 0 1, +1
  - 1 0 0 0 0 1 0, +1

Winnow

Winnow: If $y_t\,(w \,.\, x_t - d) < 0$ Filtering rule

If (y == 1) Update step

For each i s.t. (x_i == 1)

$w_i = 2\, w_i$ Update type 1

Else

For each i s.t. (x_i == 1)

$w_i = w_i / 2$ Update type 2

Due to [Littlestone, ‘88]. Similar to Halving Algorithm of [Angluin, ‘88].

- If many dimensions are irrelevant (k << d), Winnow typically converges faster than Perceptron.
- If number of examples is small w.r.t. dimensions (n << d), Perceptron typically better than Winnow.
- Extensions, *e.g.*
  - Allow constants other than 2
  - Consider each dimension i as an “expert”

Online learning with expert advice

Consider any prediction or forecasting problem with an ensemble of "experts." An expert is a time-series (*i.e.* a sequence of "predictions"), but need not be a good predictor.

- Predicting climate change
  - Intergovernmental Panel on Climate Change (IPCC) multi-model ensemble of climate models
- Weather prediction
  - Combine the predictions of an ensemble of weather models
- Portfolio management / volatility prediction
  - Experts can be analysts regularly making predictions about stock performance
  - Experts can be the stock prices themselves
- GDP Nowcasting
  - Experts can be monthly reports, *cf.* GDPNow (FT900, Monthly Retail Trade Report)
  - Experts can be GDPNow and other such real-time prediction methods

Online learning with expert advice

Problem set-up:

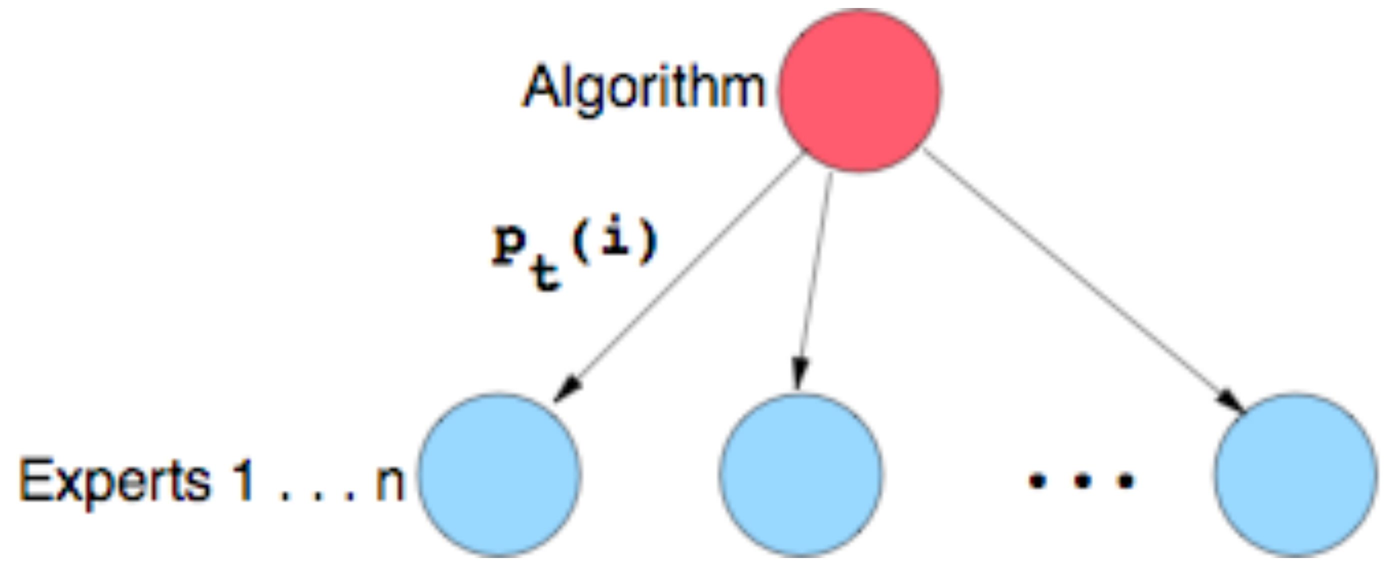
- Observations arrive one-at-a-time, in a stream
- A set of "experts" make predictions, at each time

Typical online learning algorithm template:

- First, the algorithm observes expert predictions (only) and must make a combined prediction
- Then, the true observation is revealed
- Finally, the Algorithm can update the "weight" of each expert
- Repeat

Online learning with expert advice

Learner maintains distribution over *n* “experts.” [An ensemble method]



Experts are black boxes: need not be good predictors, can vary with time, and depend on one another.

Learner predicts based on a probability distribution $p_t(i)$ over experts, i, representing how well each expert has predicted recently.

L(i, t) is prediction loss of expert i at time t. Defined per problem.

Update $p_t(i)$ using Bayesian updates: $p_{t+1}(i) \propto p_t(i)e^{-L(i,t)}$

Regret model

No statistical assumptions (non-stochastic setting)

- No assumptions on observation sequence.
- E.g., observations can even be generated online by an adaptive adversary.

Framework models supervised learning:

- Regression, estimation or classification.
- Many prediction loss functions:
  - many hypothesis classes
  - problem need not be separable

Analyze regret: difference in cumulative prediction loss from that of the optimal (in hind-sight) comparator algorithm for the particular sequence observed.