Active Learning

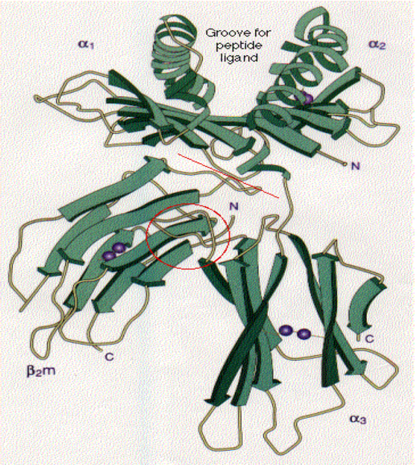
Maria-Florina Balcan

04/01/2015

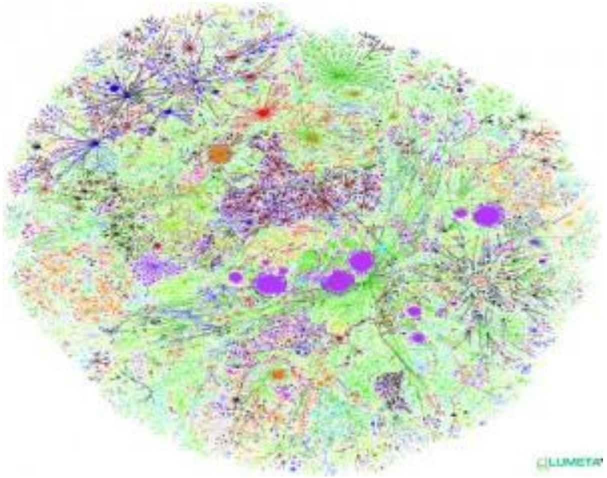
Classic Fully Supervised Learning Paradigm Insufficient Nowadays

Modern applications: massive amounts of raw data.

Only a tiny fraction can be annotated by human experts.



Protein sequences



Billions of webpages

Images

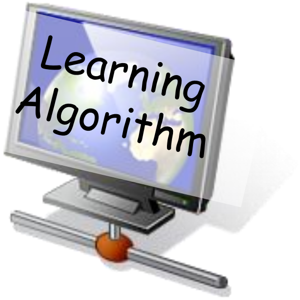
Modern ML: New Learning Approaches

Modern applications: massive amounts of raw data.

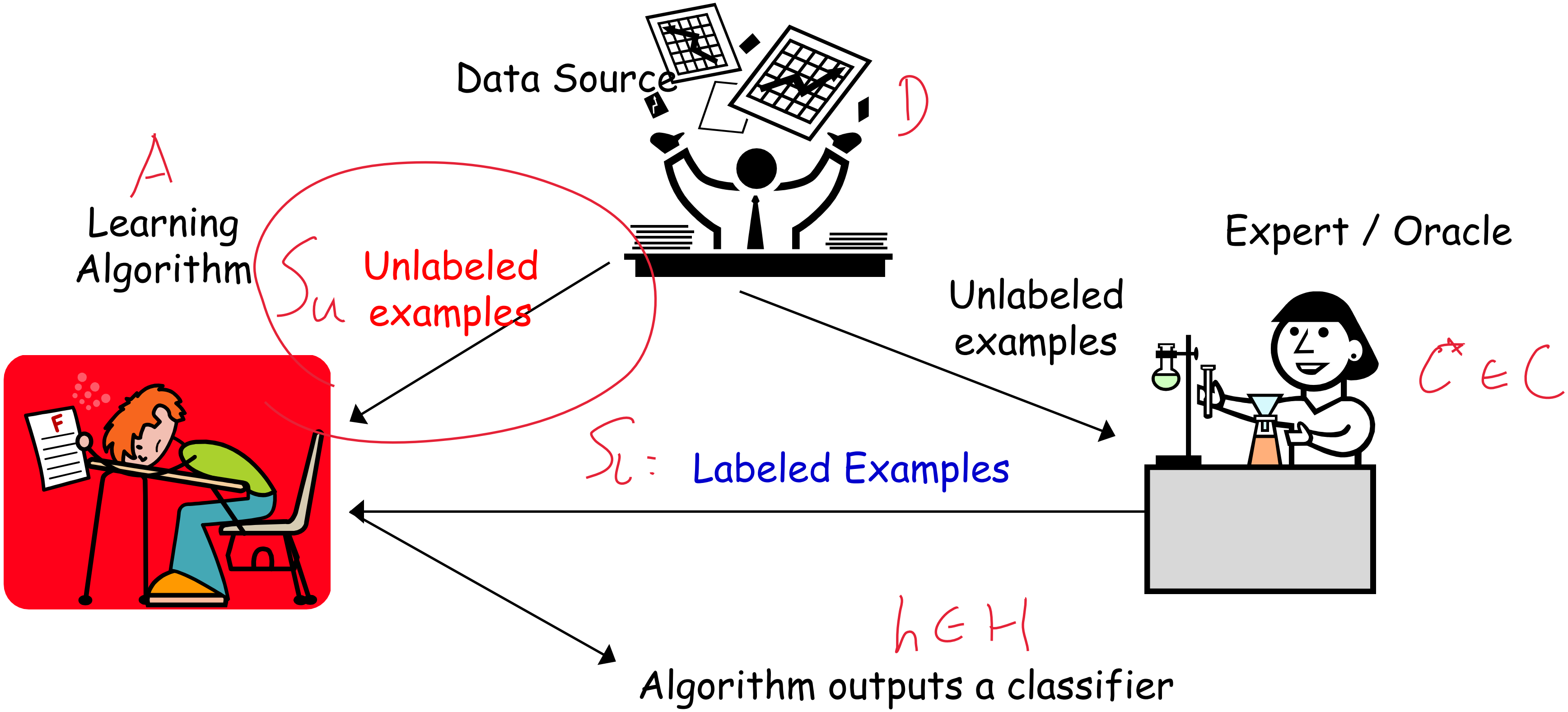
Techniques that best utilize data, minimizing need for expert/human intervention.

Paradigms where there has been great progress.

- Semi-supervised Learning, (Inter)active Learning.



Semi-Supervised Learning


Data Source

D

A

Learning Algorithm

S_u

Unlabeled examples

Expert / Oracle

$c^* \in C$

Unlabeled examples

S_l :

Labeled Examples

$h \in H$

Algorithm outputs a classifier

$S_l=\{(x_1,y_1),\dots,(x_{m_l},y_{m_l})\}$

x_i drawn i.i.d from D, $y_i = c^*(x_i)$

$S_u=\{x_1,\dots,x_{m_u}\}$ drawn i.i.d from D

Goal: h has small error over D.

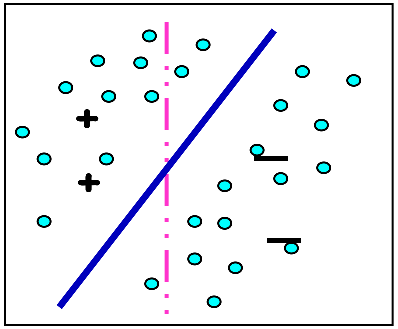
$$err_D(h) = \Pr_{x\sim D}(h(x) \neq c^*(x))$$

Semi-supervised Learning

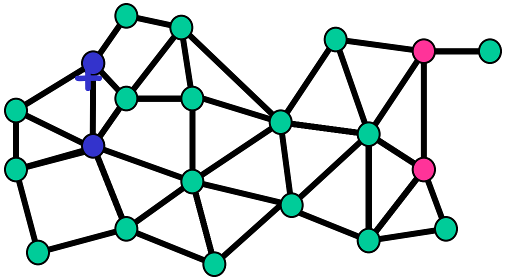
Key Insight/Underlying Fundamental Principle

Unlabeled data useful if we have a bias/belief not only about the form of the target, but also about its relationship with the underlying data distribution.

E.g., "large margin separator" [Joachims '99]

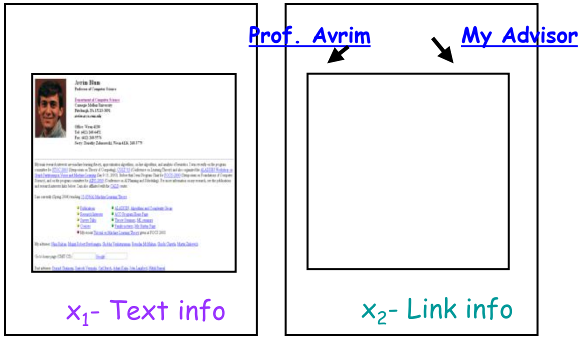


Similarity based ("small cut") [B&C01], [ZGL03]



"self-consistent rules" [Blum & Mitchell '98]

$x = \langle x_1, x_2 \rangle$ $h_1(x_1)=h_2(x_2) = h(x)$



- Unlabeled data can help reduce search space or re-order the fns in the search space according to our belief, biasing the search towards fns satisfying the belief (which becomes concrete once we see unlabeled data).

A General Discriminative Model for SSL

[BalcanBlum, COLT 2005; JACM 2010]

As in PAC/SLT, discuss algorithmic and sample complexity issues.

Analyze fundamental sample complexity aspects:

- How much unlabeled data is needed.
  - depends both on complexity of H and of compatibility notion.
- Ability of unlabeled data to reduce #of labeled examples.
  - compatibility of the target, helpfulness of the distrib.

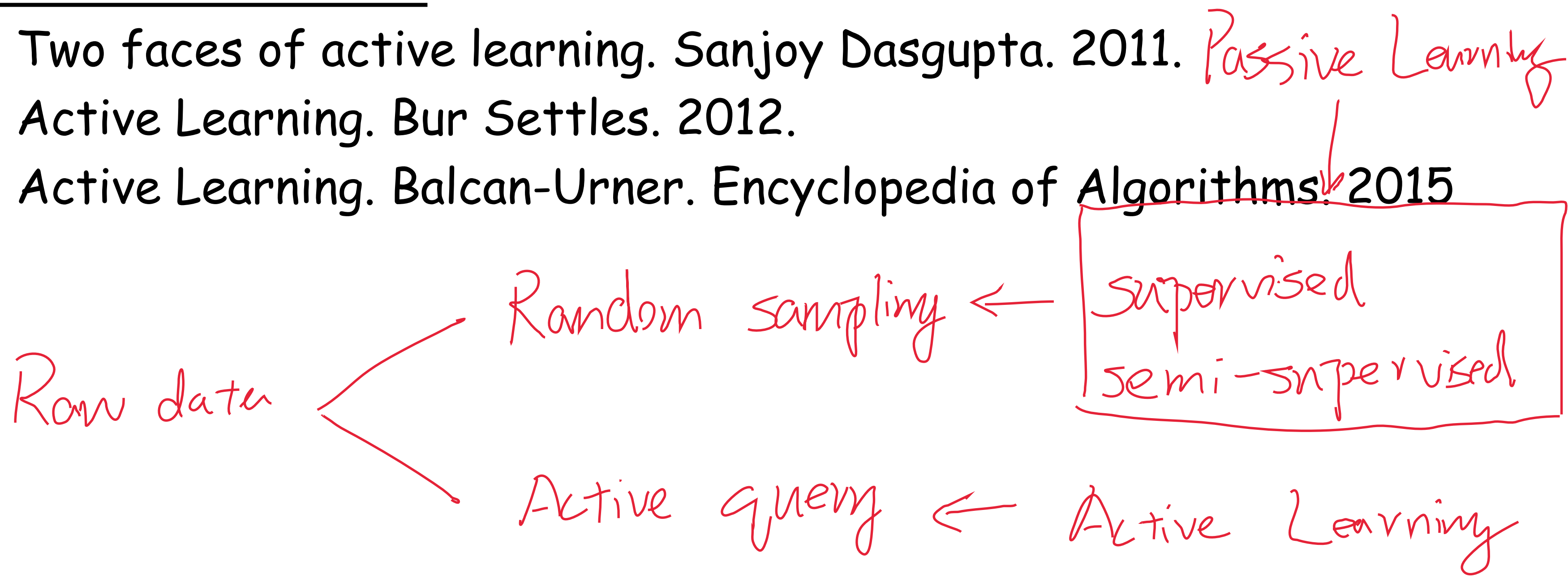
- Survey on "Semi-Supervised Learning" (Jerry Zhu, 2010) explains the SSL techniques from this point of view.

- Note: the mixture method that Tom talked about on Feb 25th can be explained from this point of view too. See the Zhu survey.

Active Learning

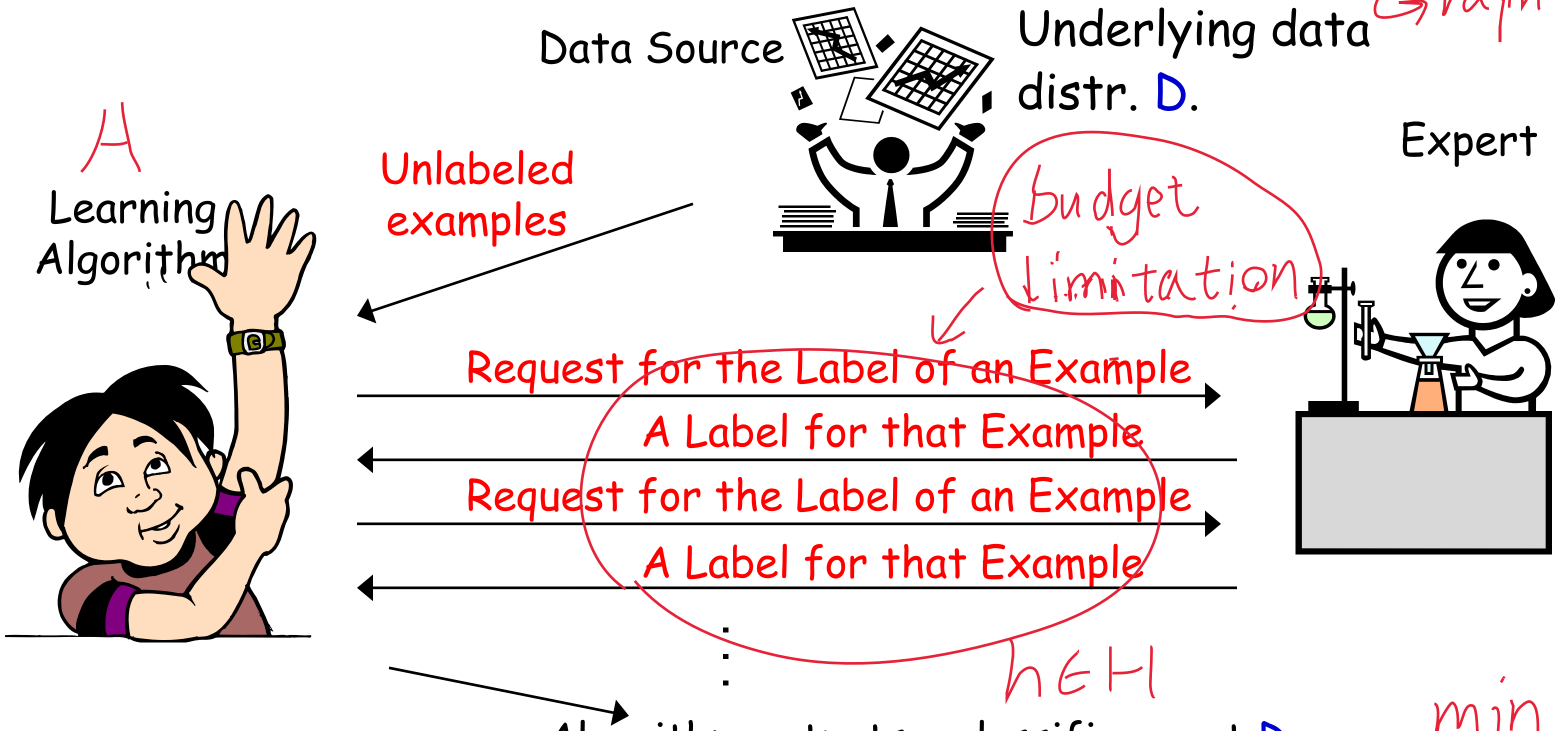
Additional resources:

- Two faces of active learning. Sanjoy Dasgupta. 2011.
- Active Learning. Bur Settles. 2012.
- Active Learning. Balcan-Urner. Encyclopedia of Algorithms. 2015



Batch Active Learning

active SVM
Graph-based



$\mathcal{A}$

Learning Algorithm

Data Source

Underlying data distr. D.

Expert

$c^* \in \mathbb{C}$

Unlabeled examples

budget limitation

Request for the Label of an Example

A Label for that Example

Request for the Label of an Example

A Label for that Example

. . .

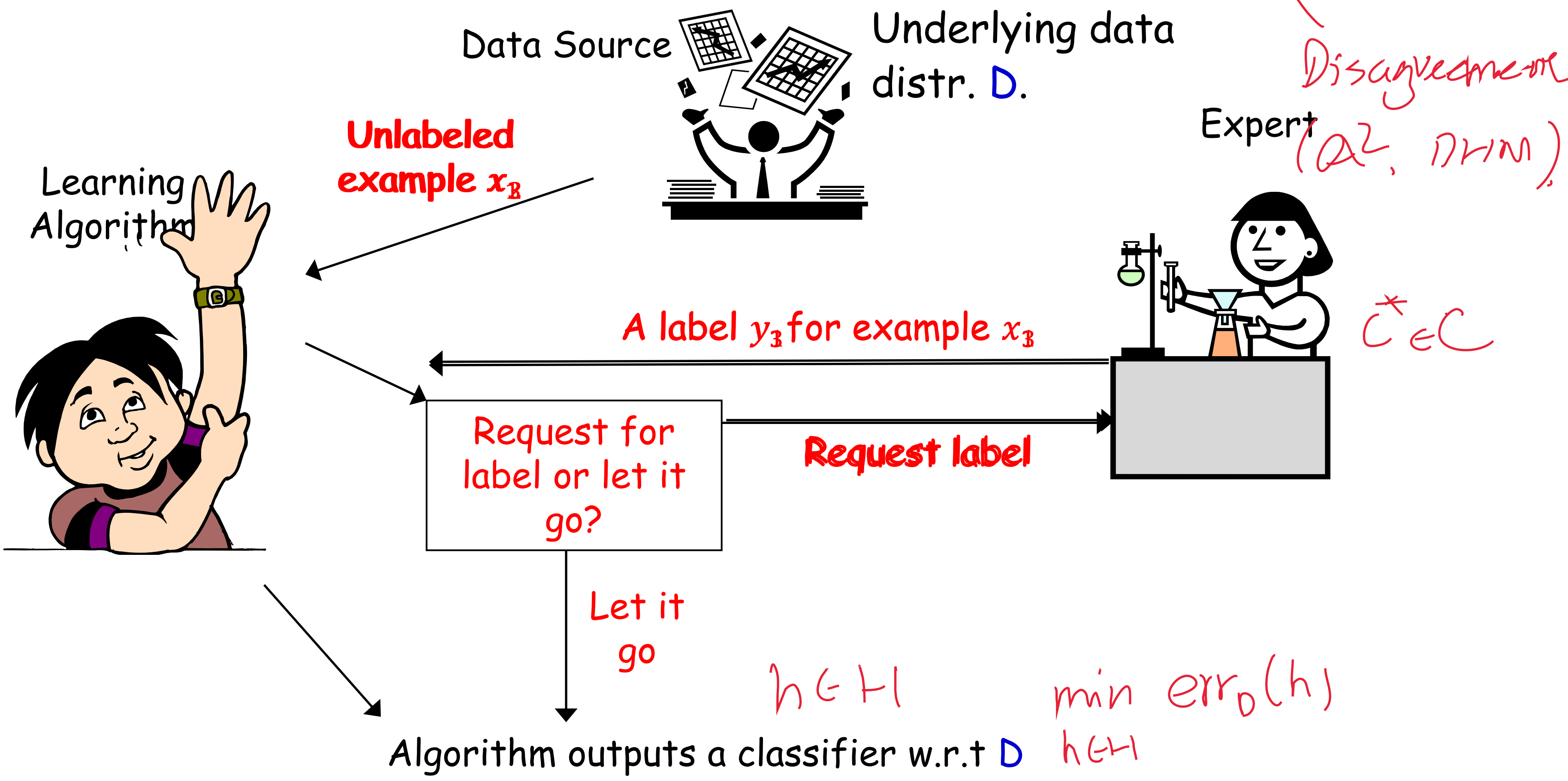
$h \in H$

Algorithm outputs a classifier w.r.t D

$$\min_{h \in H} err_D(h)$$

- Learner can choose specific examples to be labeled.
- Goal: use fewer labeled examples [pick informative examples to be labeled].

Selective Sampling Active Learning


Data Source
Underlying data distr. D.
Disagreement (A2, DHM)
Expert
Learning Algorithm
Unlabeled example x_3
A label y_3 for example x_3
Request for label or let it go?
Request label
Let it go
$c^* \in C$
$h \in H$
$\min_{h \in H} err_D(h)$
Algorithm outputs a classifier w.r.t D

- Selective sampling AL (Online AL): stream of unlabeled examples, when each arrives make a decision to ask for label or not.
- Goal: use fewer labeled examples [pick informative examples to be labeled].

What Makes a Good Active Learning Algorithm?

- Guaranteed to output a relatively good classifier for most learning problems.

- Doesn't make too many label requests.

  Hopefully a lot less than passive learning and SSL.

- Need to choose the label requests carefully, to get informative labels.

Can adaptive querying really do better than passive/random sampling?

- YES! (sometimes)

- We often need far fewer labels for active learning than for passive.

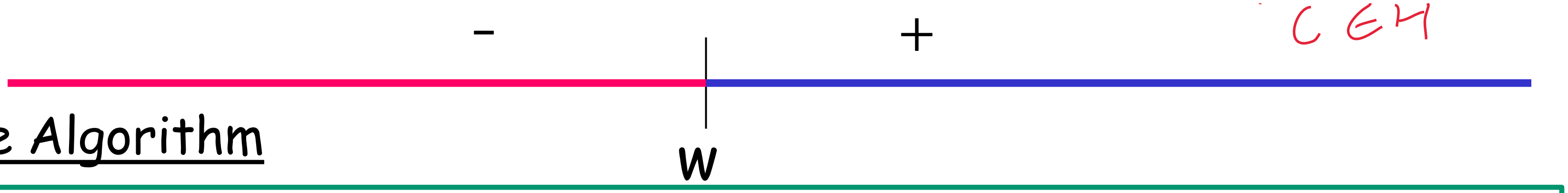
- This is predicted by theory and has been observed in practice.

Can adaptive querying help? [CAL92, Dasgupta04]

infinite

- Threshold fns on the real line: $h_w(x) = 1(x \geq w)$, C H $= \{h_w : w \in R\}$

$c^* \in H$



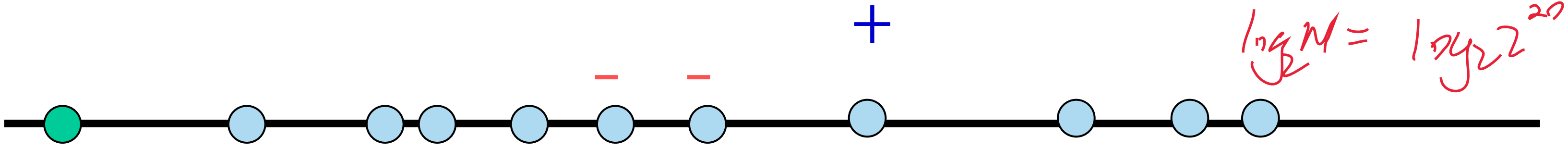
Active Algorithm

- Get N unlabeled examples
- How can we recover the correct labels with ≪ N queries?
- Do binary search!   Just need O(log N) labels!

$\approx 2^{20}$

$N = 1{,}000{,}000 = 10^6$

$\log_2 N = \log_2 2^{20} = 20$



- Output a classifier consistent with the N inferred labels.

- $N = O(1/\epsilon)$ we are guaranteed to get a classifier of error $\leq \epsilon$.

Passive supervised: $\Omega(1/\epsilon)$ labels to find an ε-accurate threshold.

Active: only $O(\log 1/\epsilon)$ labels.   Exponential improvement.

Common Technique in Practice

Uncertainty sampling in SVMs common and quite useful in practice. E.g., [Tong & Koller, ICML 2000; Jain, Vijayanarasimhan & Grauman, NIPS 2010; Schohon Cohn, ICML 2000]

Active SVM Algorithm

- At any time during the alg., we have a "current guess" w_t of the separator: the max-margin separator of all labeled points so far.
- Request the label of the example closest to the current separator.

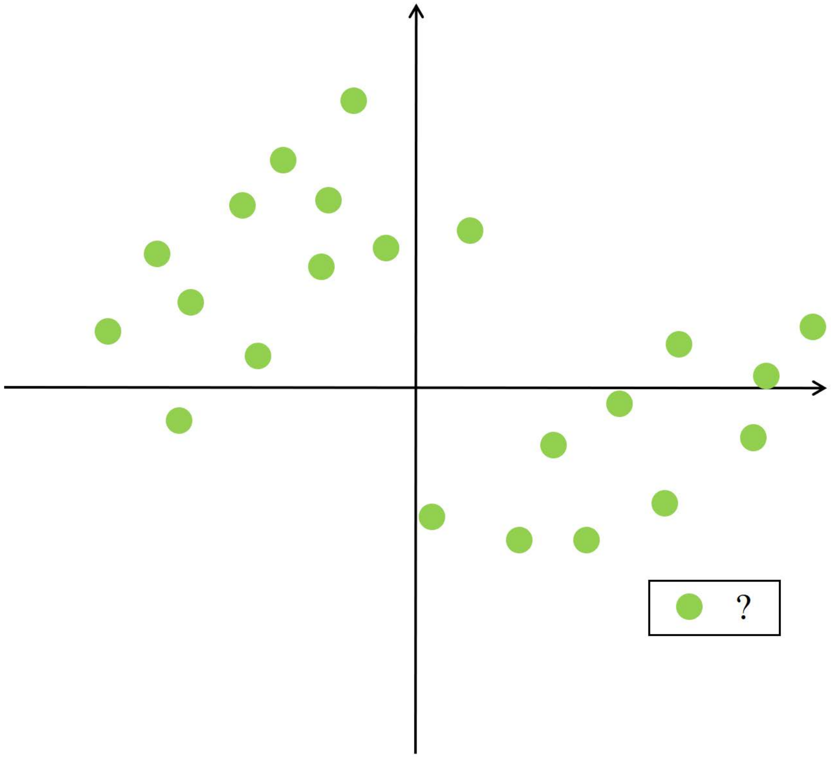
Common Technique in Practice

Active SVM seems to be quite useful in practice.

[Tong & Koller, ICML 2000; Jain, Vijayanarasimhan & Grauman, NIPS 2010]

Algorithm (batch version)

Input $S_u=\{x_1, \ldots, x_{m_u}\}$ drawn i.i.d from the underlying source D



Common Technique in Practice

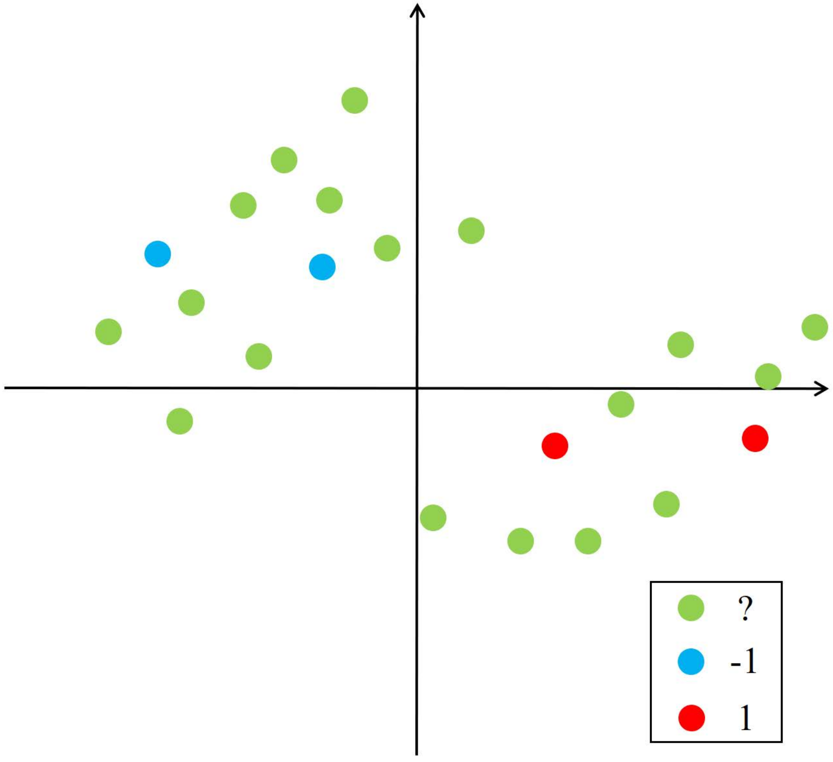
Active SVM seems to be quite useful in practice.

[Tong & Koller, ICML 2000; Jain, Vijayanarasimhan & Grauman, NIPS 2010]

Algorithm (batch version)

Input $S_u = \{x_1, \ldots, x_{m_u}\}$ drawn i.i.d from the underlying source D

Start: query for the labels of a few random x_is.



Common Technique in Practice

Active SVM seems to be quite useful in practice.

[Tong & Koller, ICML 2000; Jain, Vijayanarasimhan & Grauman, NIPS 2010]

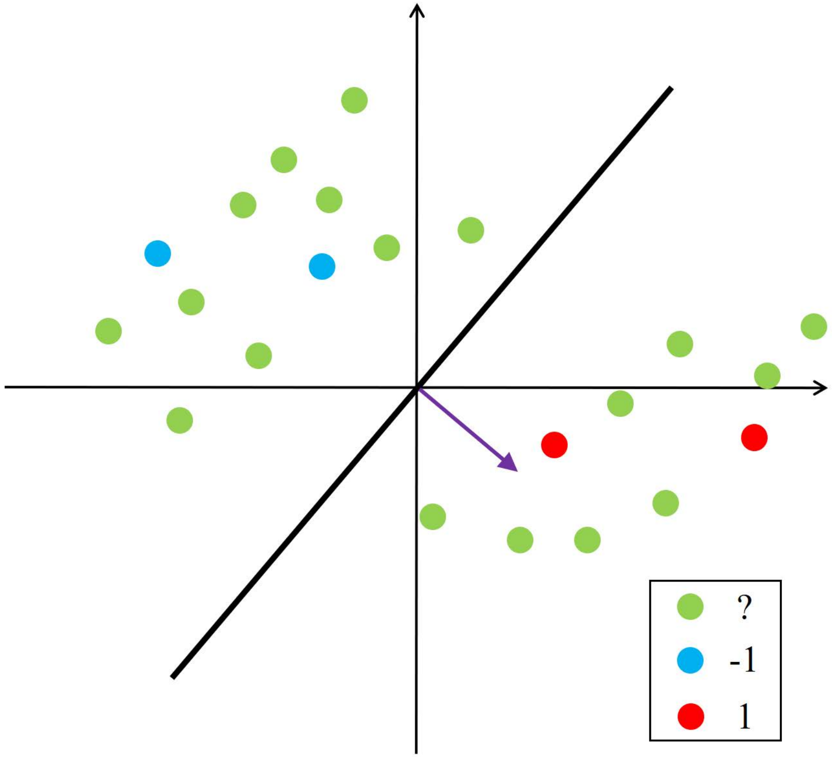
Algorithm (batch version)

Input $S_u=\{x_1, \ldots, x_{m_u}\}$ drawn i.i.d from the underlying source D

Start: query for the labels of a few random x_is.

For $t = 1, \ldots,$

- Find w_t the max-margin separator of all labeled points so far.



Common Technique in Practice

Active SVM seems to be quite useful in practice.

[Tong & Koller, ICML 2000; Jain, Vijayanarasimhan & Grauman, NIPS 2010]

Algorithm (batch version)

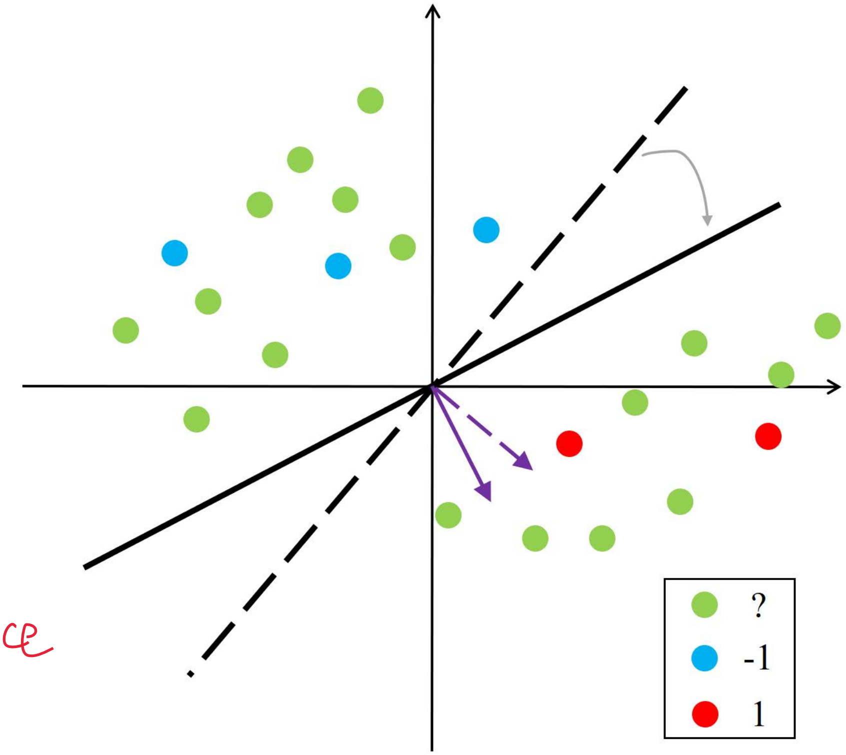
Input $S_u=\{x_1, \ldots, x_{m_u}\}$ drawn i.i.d from the underlying source D

Start: query for the labels of a few random x_is.

For $t = 1, \ldots,$

- Find w_t the max-margin separator of all labeled points so far.
- Request the label of the example closest to the current separator: minimizing $\underline{|x_i \cdot w_t|}$.

(highest uncertainty) confidence



Common Technique in Practice

Active SVM seems to be quite useful in practice.

E.g., Jain, Vijayanarasimhan & Grauman, NIPS 2010

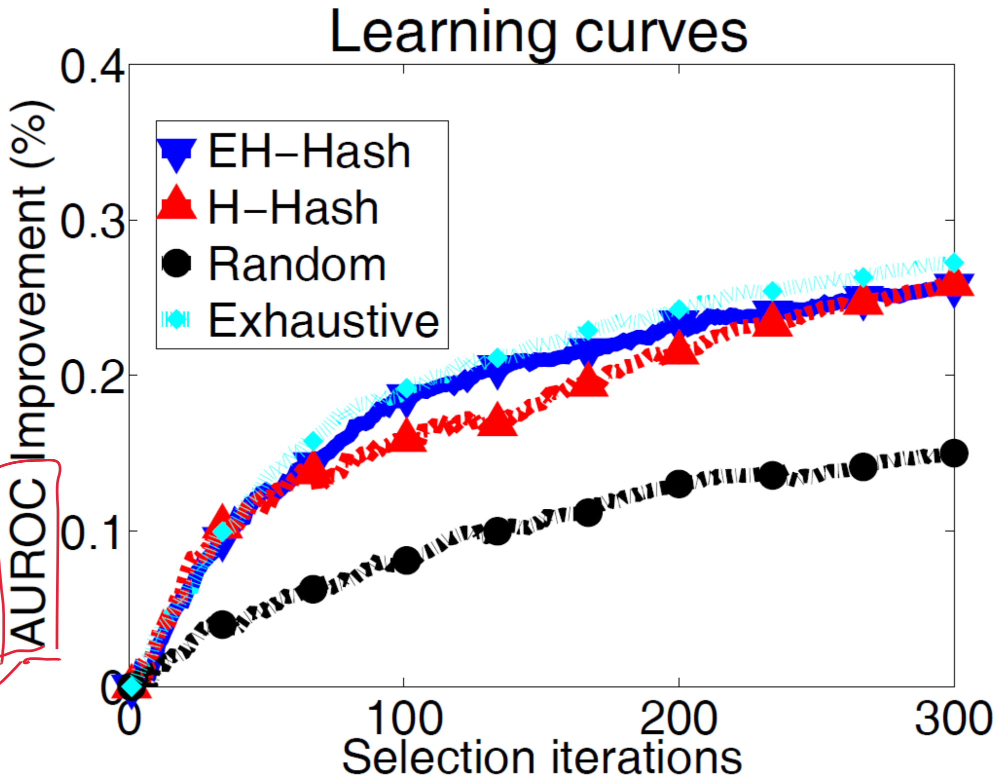
Newsgroups dataset (20.000 documents from 20 categories)

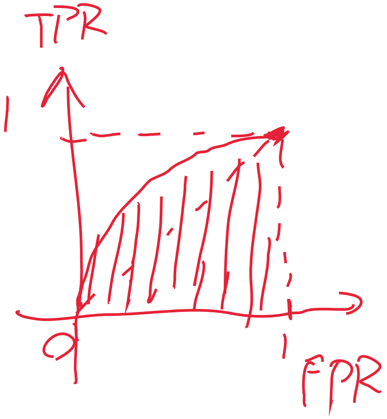
		True	
		P	N
Pre	P	TP	FP
	N	FN	TN

$$FPR = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{TP + FN}$$

Area Under ROC





$f(x) \in [0, 1]$

$f(x) = 0.8$

$Th = 0.5$

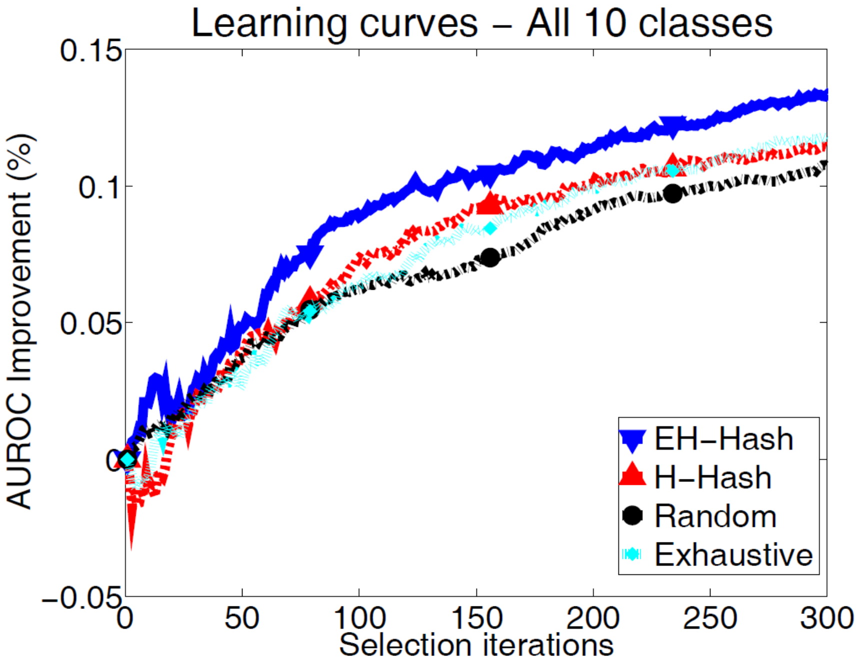
0.5

Common Technique in Practice

Active SVM seems to be quite useful in practice.

E.g., Jain, Vijayanarasimhan & Grauman, NIPS 2010

CIFAR-10 image dataset (60.000 images from 10 categories)



Active SVM/Uncertainty Sampling

- Works sometimes….

- **However, we need to be very very very careful!!!**

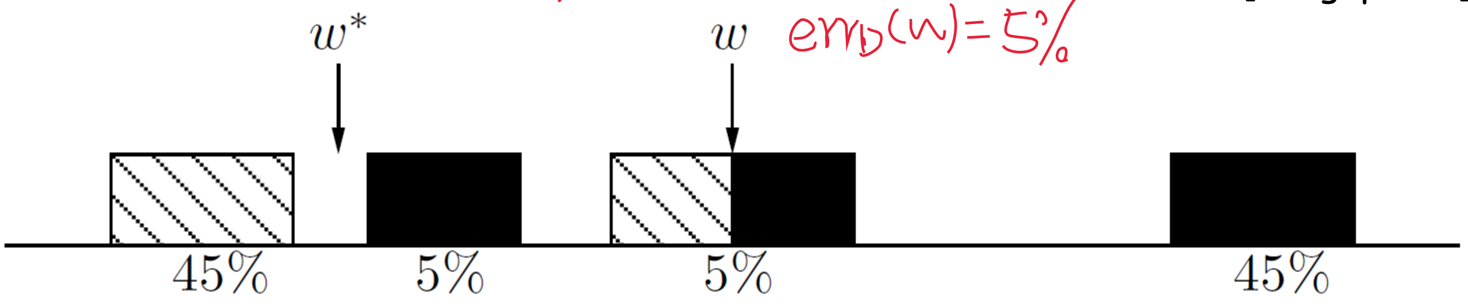
  - Myopic, greedy technique can suffer from sampling bias.

  - A bias created because of the querying strategy; as time goes on the sample is less and less representative of the true data source.

$\min_{h \in H} err_D(h)$

$err_D(w^*) = 2.5\%$

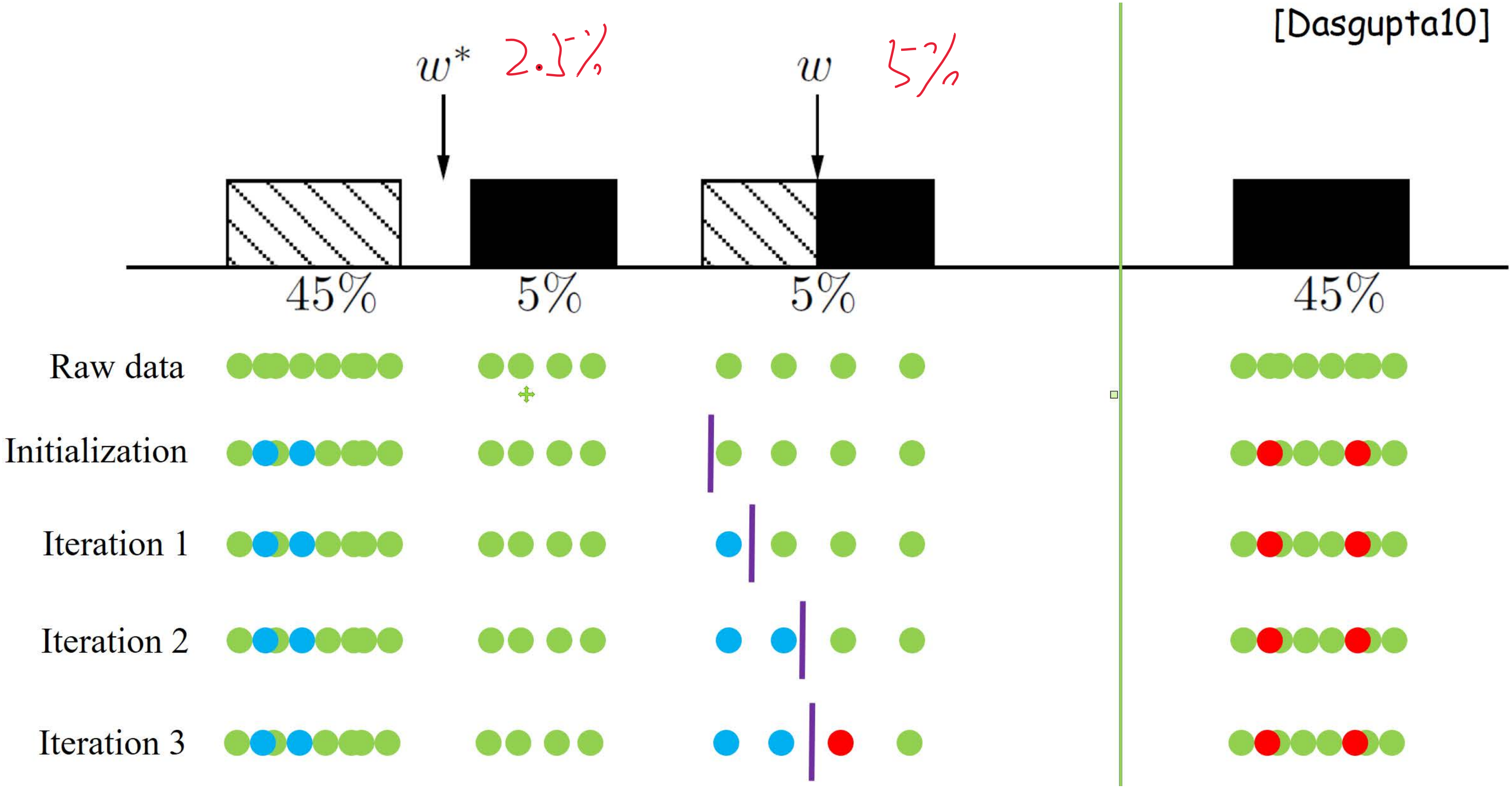
[Dasgupta10]



$err_D(w) = 5\%$

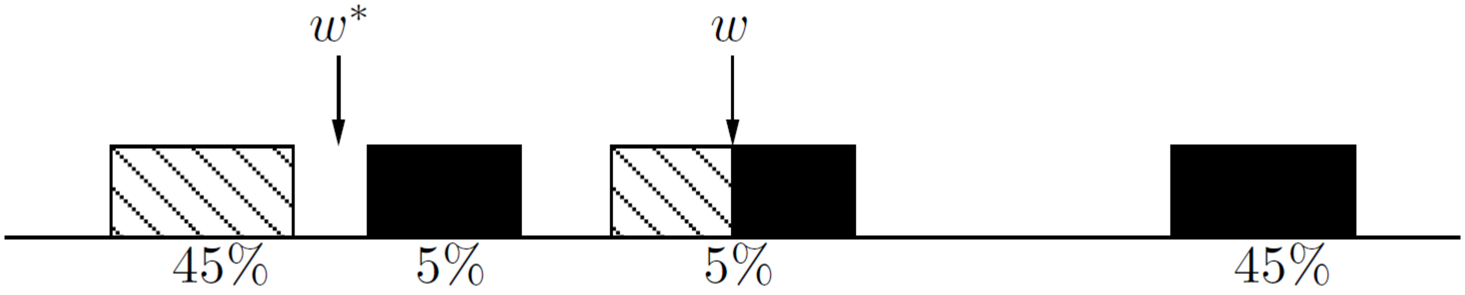
Active SVM/Uncertainty Sampling

- Works sometimes….
- **However, we need to be very very careful!!!**



Active SVM/Uncertainty Sampling

- Works sometimes….
- **However, we need to be very very careful!!!**
  - Myopic, greedy technique can suffer from sampling bias.
  - Bias created because of the querying strategy; as time goes on the sample is less and less representative of the true source.
  - Observed in practice too!!!!
- **Main tension**: want to choose informative points, but also want to guarantee that the classifier we output does well on true random examples from the underlying distribution.



Safe Active Learning Schemes

Disagreement Based Active Learning

Hypothesis Space Search

[CAL92] [BBL06]

[Hanneke'07, DHM'07, Wang'09 , Fridman'09, Kolt10, BHW'08, BHLZ'10, H'10, Ailon'12, …]

Version Spaces

- X - feature/instance space; distr. D over X; c^* target fnc
- Fix hypothesis space H.

Definition (Mitchell'82) Assume realizable case: $\underline{c^* \in H.}$

Given a set of labeled examples $S = (x_1, y_1), \ldots, (x_{m_l}, y_{m_l}), y_i = c^*(x_i)$

Version space of H: part of H consistent with labels so far.

I.e., $h \in VS(H)$ iff $h(x_i) = c^*(x_i)\ \forall i \in \{1, \ldots, m_l\}$.

$err_S(h) = 0$

Version Spaces

- X – feature/instance space; distr. D over X; c^* target fnc
- Fix hypothesis space H.

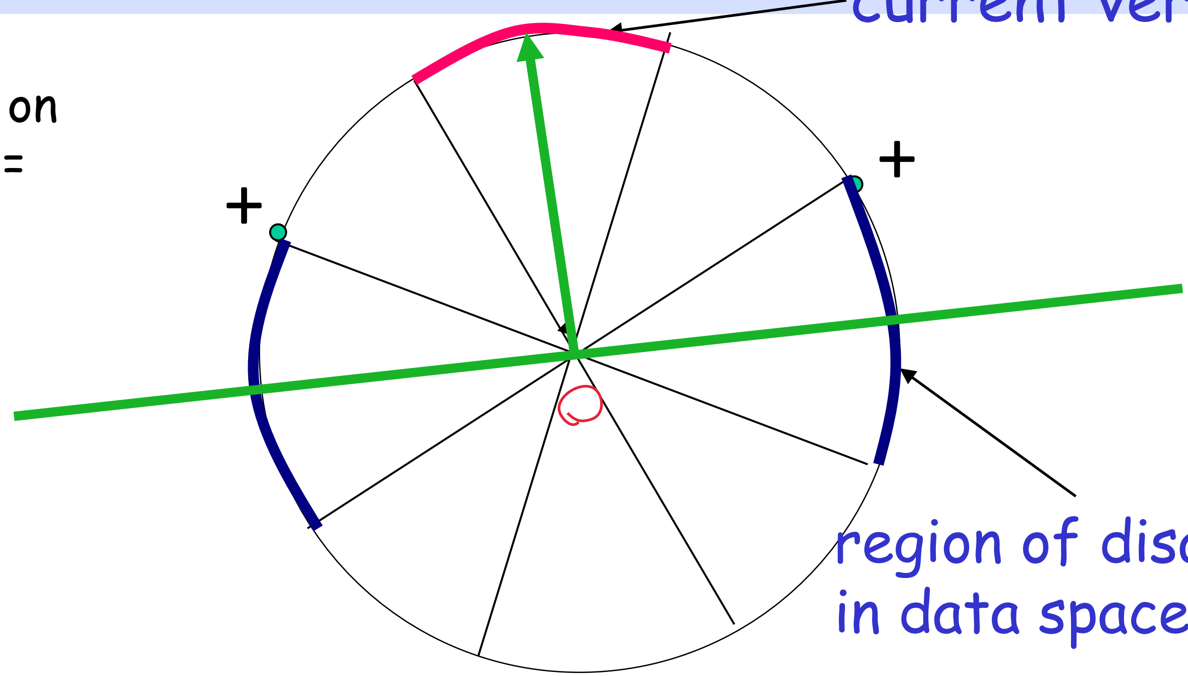
Definition (Mitchell'82) Assume realizable case: $c^* \in H$.

Given a set of labeled examples $(x_1, y_1), \ldots, (x_{m_l}, y_{m_l}), y_i = c^*(x_i)$

Version space of H: part of H consistent with labels so far.

E.g.,: data lies on circle in R^2, H = homogeneous linear seps.

$h(x) = w^T x$



current version space

region of disagreement in data space

Version Spaces. Region of Disagreement

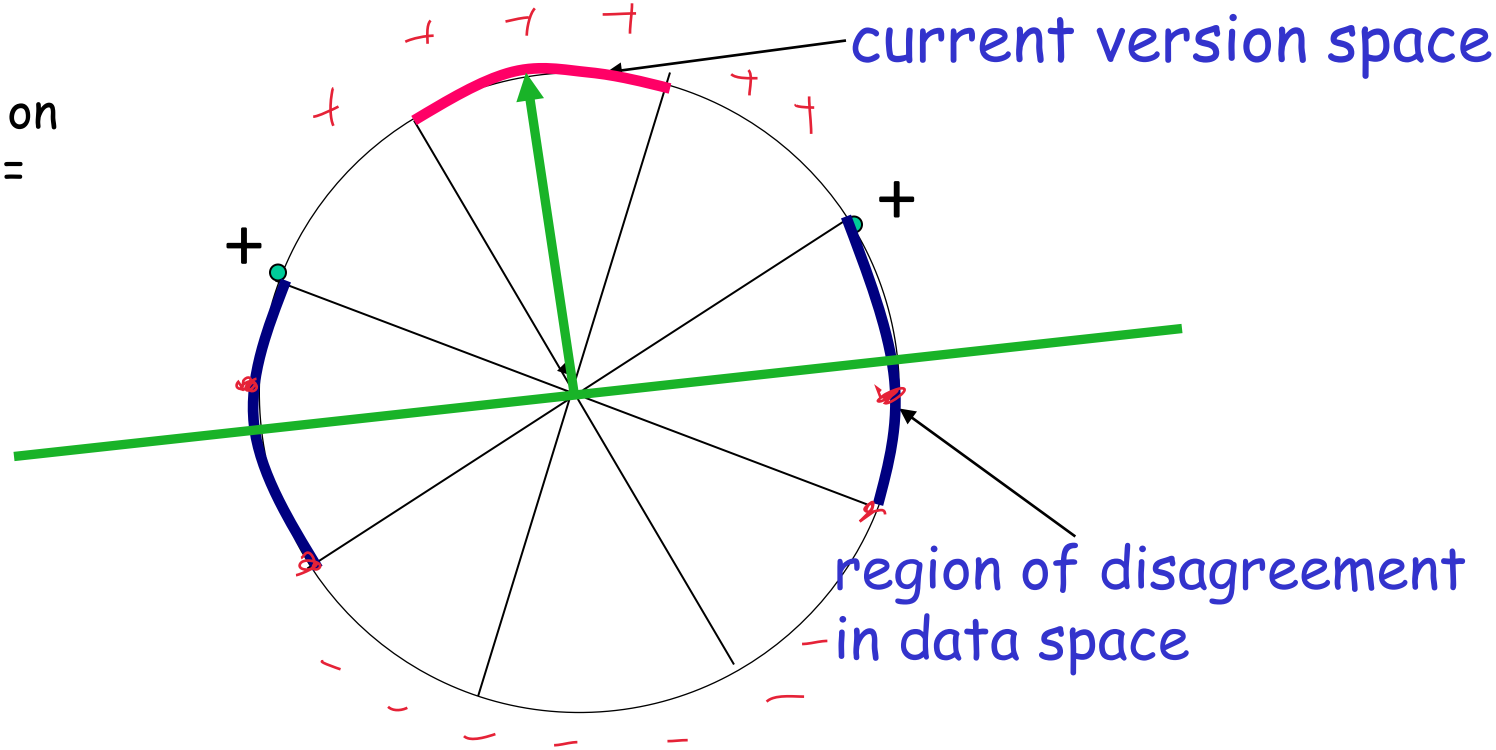
Definition (CAL'92)

Version space: part of H consistent with labels so far.

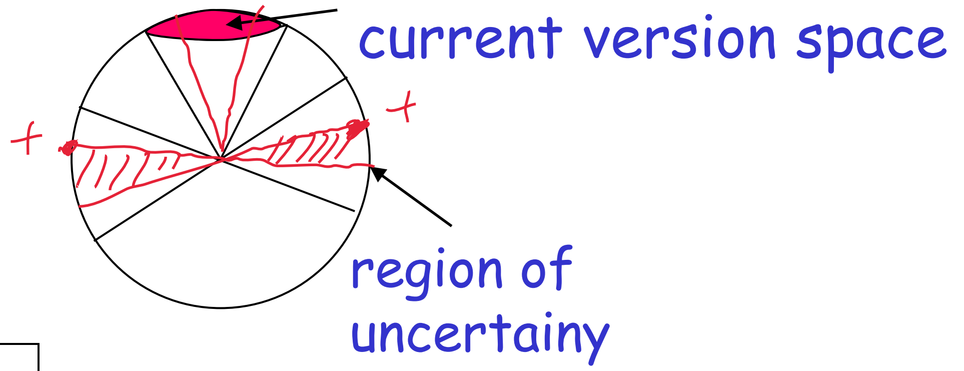
Region of disagreement = part of data space about which there is still some uncertainty (i.e. disagreement within version space)

$x \in X, x \in \underline{DIS(VS(H))}$ iff $\underline{\exists} h_1, h_2 \in VS(H), h_1(x) \neq h_2(x)$

E.g.,: data lies on circle in R^2, H = homogeneous linear seps.



Disagreement Based Active Learning [CAL92]



current version space

region of uncertainy

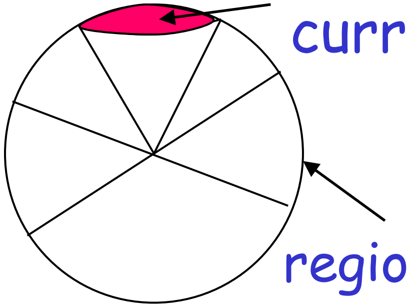
Algorithm:

Pick a few points at random from the current region of uncertainty and query their labels.

Stop when region of uncertainty is small.

Note: it is active since we do not waste labels by querying in regions of space we are certain about the labels.

Disagreement Based Active Learning [CAL92]



current version space

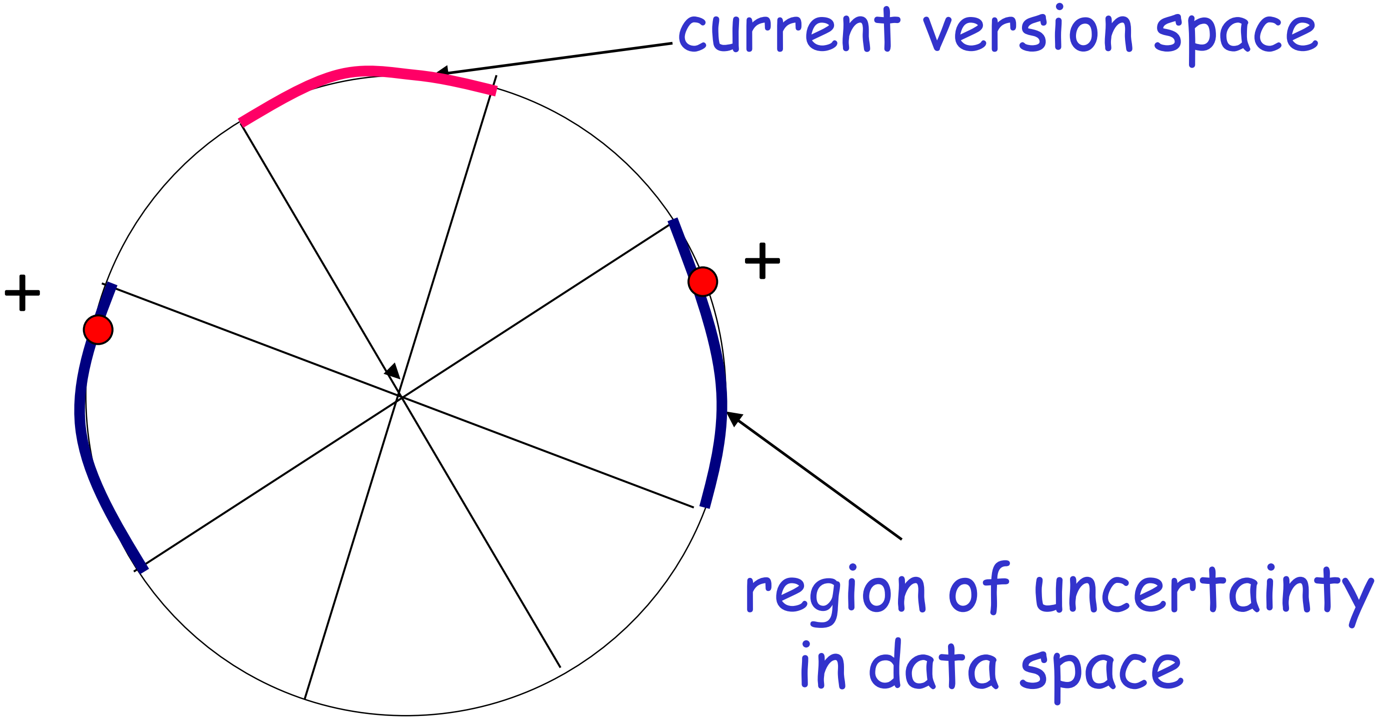
region of uncertainy

Algorithm:

```
Query for the labels of a few random $x_i$s.
Let $\mathrm{H}_1$ be the current version space.
For $t = 1, \ldots,$
    Pick a few points at random from the current region of disagreement $\mathrm{DIS}(\mathrm{H}_t)$ and query their labels.
    Let $\mathrm{H}_{t+1}$ be the new version space.
```

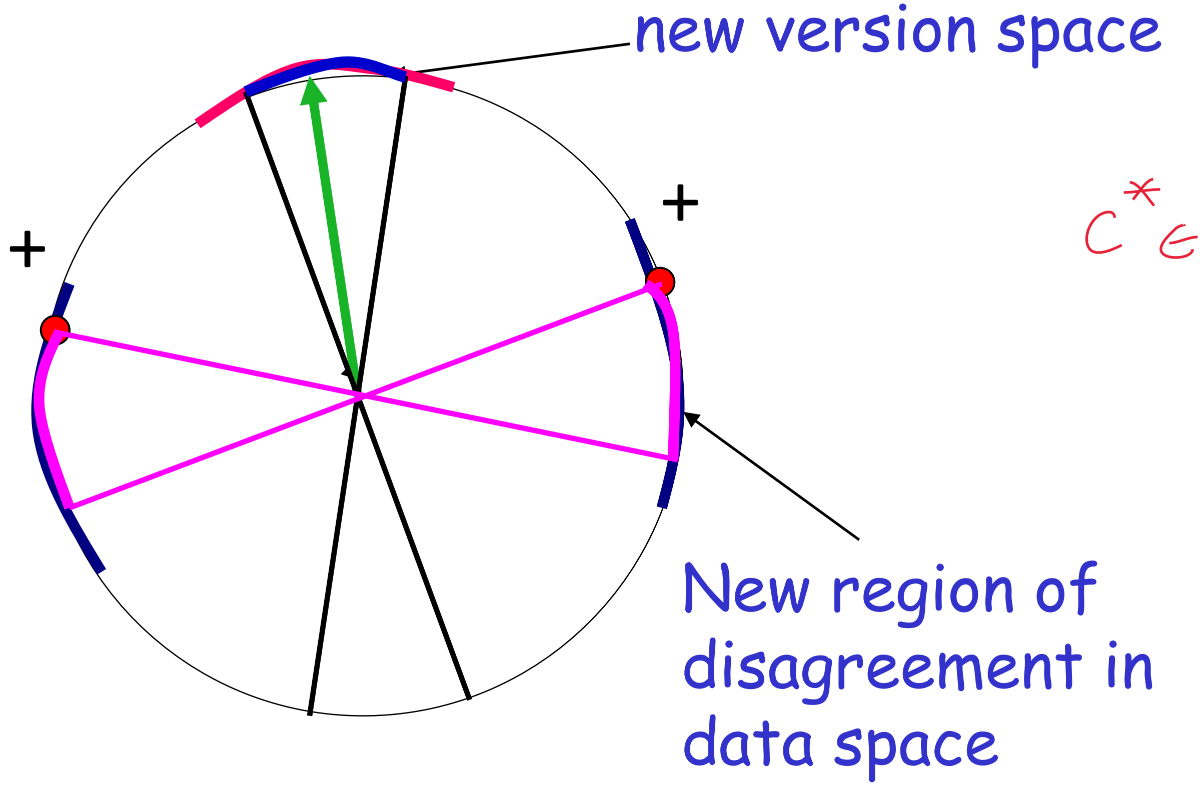
Region of uncertainty [CAL92]

- Current version space: part of C consistent with labels so far.
- "Region of uncertainty" = part of data space about which there is still some uncertainty (i.e. disagreement within version space)



Region of uncertainty [CAL92]

- Current version space: part of *C* consistent with labels so far.
- "Region of uncertainty" = part of data space about which there is still some uncertainty (i.e. disagreement within version space)



new version space

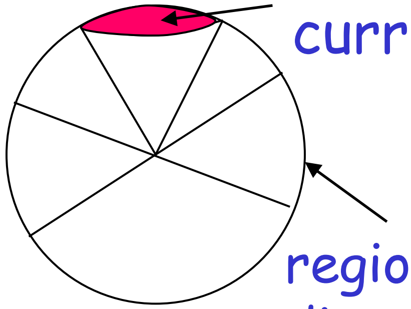
New region of disagreement in data space

$c^* \in H$

$c^* \notin H$

How about the agnostic case where the target might not belong the H?

A^2 Agnostic Active Learner [BBL'06]



current version space

region of disagreement

Algorithm:

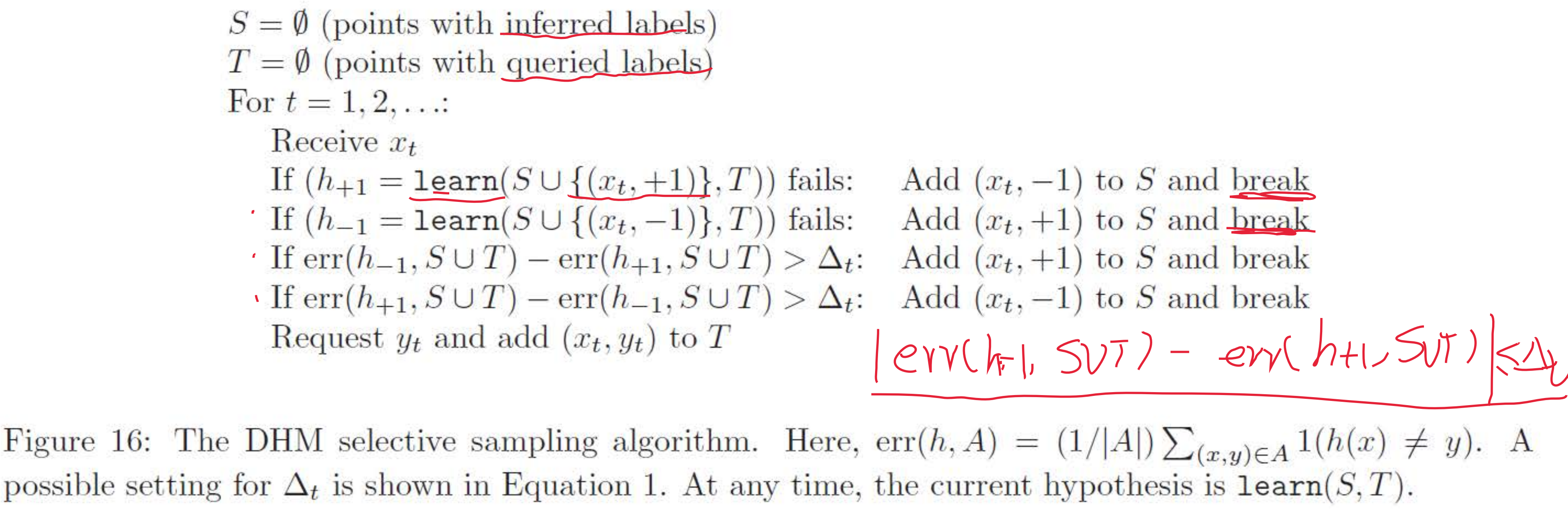
Careful use of generalization bounds; Avoid the sampling bias!!!!

Let $H_1 = H$.

For $t = 1, \ldots.,$

- Pick a few points at random from the current region of disagreement $DIS(H_t)$ and query their labels.
- Throw out hypothesis if you are statistically confident they are suboptimal.

The DHN Agnostic Active Learner [DHN'07]



```
$S = \emptyset$ (points with inferred labels)
$T = \emptyset$ (points with queried labels)
For $t = 1, 2, \ldots$:
    Receive $x_t$
    If ($h_{+1} = \texttt{learn}(S \cup \{(x_t, +1)\}, T)$) fails:    Add $(x_t, -1)$ to $S$ and break
    If ($h_{-1} = \texttt{learn}(S \cup \{(x_t, -1)\}, T)$) fails:    Add $(x_t, +1)$ to $S$ and break
    If $\mathrm{err}(h_{-1}, S \cup T) - \mathrm{err}(h_{+1}, S \cup T) > \Delta_t$:    Add $(x_t, +1)$ to $S$ and break
    If $\mathrm{err}(h_{+1}, S \cup T) - \mathrm{err}(h_{-1}, S \cup T) > \Delta_t$:    Add $(x_t, -1)$ to $S$ and break
    Request $y_t$ and add $(x_t, y_t)$ to $T$
```

$|\mathrm{err}(h_{-1}, S \cup T) - \mathrm{err}(h_{+1}, S \cup T)| \le \Delta_t$

Figure 16: The DHM selective sampling algorithm. Here, $\mathrm{err}(h, A) = (1/|A|) \sum_{(x,y) \in A} 1(h(x) \neq y)$. A possible setting for Δ_t is shown in Equation 1. At any time, the current hypothesis is $\texttt{learn}(S, T)$.

$\texttt{learn}(A, B)$ returns a hypothesis $h \in \mathcal{H}$ consistent with A, and with minimum error on B. If there is no hypothesis consistent with A, a failure flag is returned.

$$\Delta_t = \beta_t^2 + \beta_t \left(\sqrt{\mathrm{err}(h_{+1}, S \cup T)} + \sqrt{\mathrm{err}(h_{-1}, S \cup T)} \right), \qquad \beta_t = C \sqrt{\frac{d \log t + \log(1/\delta)}{t}}$$

When Active Learning Helps. Agnostic case

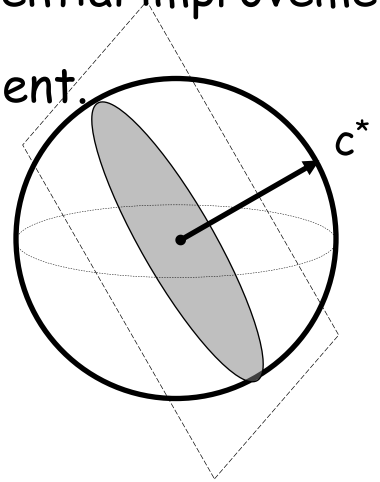
A^2 the first algorithm which is robust to noise.

[Balcan, Beygelzimer, Langford, ICML'06] [Balcan, Beygelzimer, Langford, JCSS'08]

"Region of disagreement" style: Pick a few points at random from the current region of disagreement, query their labels, throw out hypothesis if you are statistically confident they are suboptimal.

Guarantees for A^2 [BBL'06,'08]:

- It is safe (never worse than passive learning) & exponential improvements.
  - C – thresholds, low noise, exponential improvement.
  - C - homogeneous linear separators in R^d,
    D - uniform, low noise, only $\underline{d^2 \log (1/\varepsilon)}$ labels.



A lot of subsequent work.

[Hanneke'07, DHM'07, Wang'09 , Fridman'09, Kolt10, BHW'08, BHLZ'10, H'10, Ailon'12, …]

General guarantees for A^2 Agnostic Active Learner

"Disagreement based": Pick a few points at random from the current region of uncertainty, query their labels, throw out hypothesis if you are statistically confident they are suboptimal. [BBL'06]

How quickly the region of disagreement collapses as we get closer and closer to optimal classifier

Supremum (maximum)

Guarantees for A^2 [Hanneke'07]:

Disagreement coefficient $\theta_{c^*} = \sup_{r \geq \eta + \epsilon} \frac{\Pr(DIS(B(c^*, r)))}{r}$

$\eta = \inf_{h \in H} err_D(h)$

infimum (minimum)

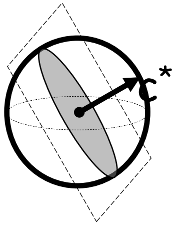
Theorem

$$m = \left(1 + \frac{\eta^2}{\epsilon^2}\right) VCdim(C)\theta_{c^*}^2 \log(\frac{1}{\varepsilon})$$

labels are sufficient s.t. with prob. $\geq 1 - \delta$ output h with $err(h) \leq \eta + \epsilon$.

Realizable case: $m = VCdim(C)\theta_{c^*} \log(\frac{1}{\varepsilon})$

Linear Separators, uniform distr.: $\theta_{c^*} = \sqrt{d}$



Disagreement Based Active Learning

"Disagreement based " algos: query points from current region of disagreement, throw out hypotheses when statistically confident they are suboptimal.

- Generic (any class), adversarial label noise.
- Computationally efficient for classes of small VC-dimension

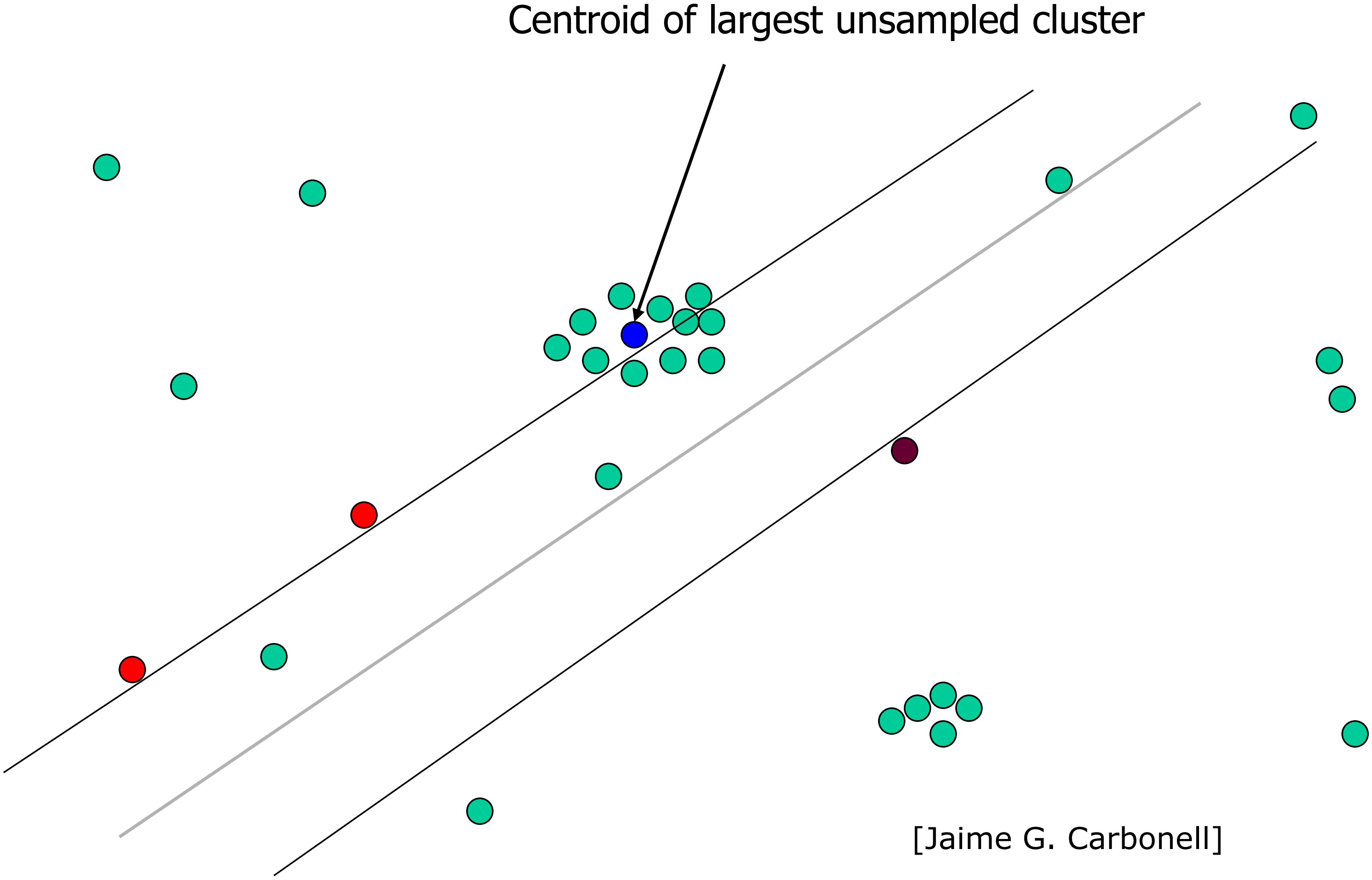
Still, could be suboptimal in label complex & computationally inefficient in general.

Lots of subsequent work trying to make is more efficient computationally and more aggressive too: [Hanneke07, DasguptaHsuMontleoni'07, Wang'09 , Fridman'09, Koltchinskii10, BHW'08, BeygelzimerHsuLangfordZhang'10, Hsu'10, Ailon'12, …]

Other Interesting AL Techniques used in Practice

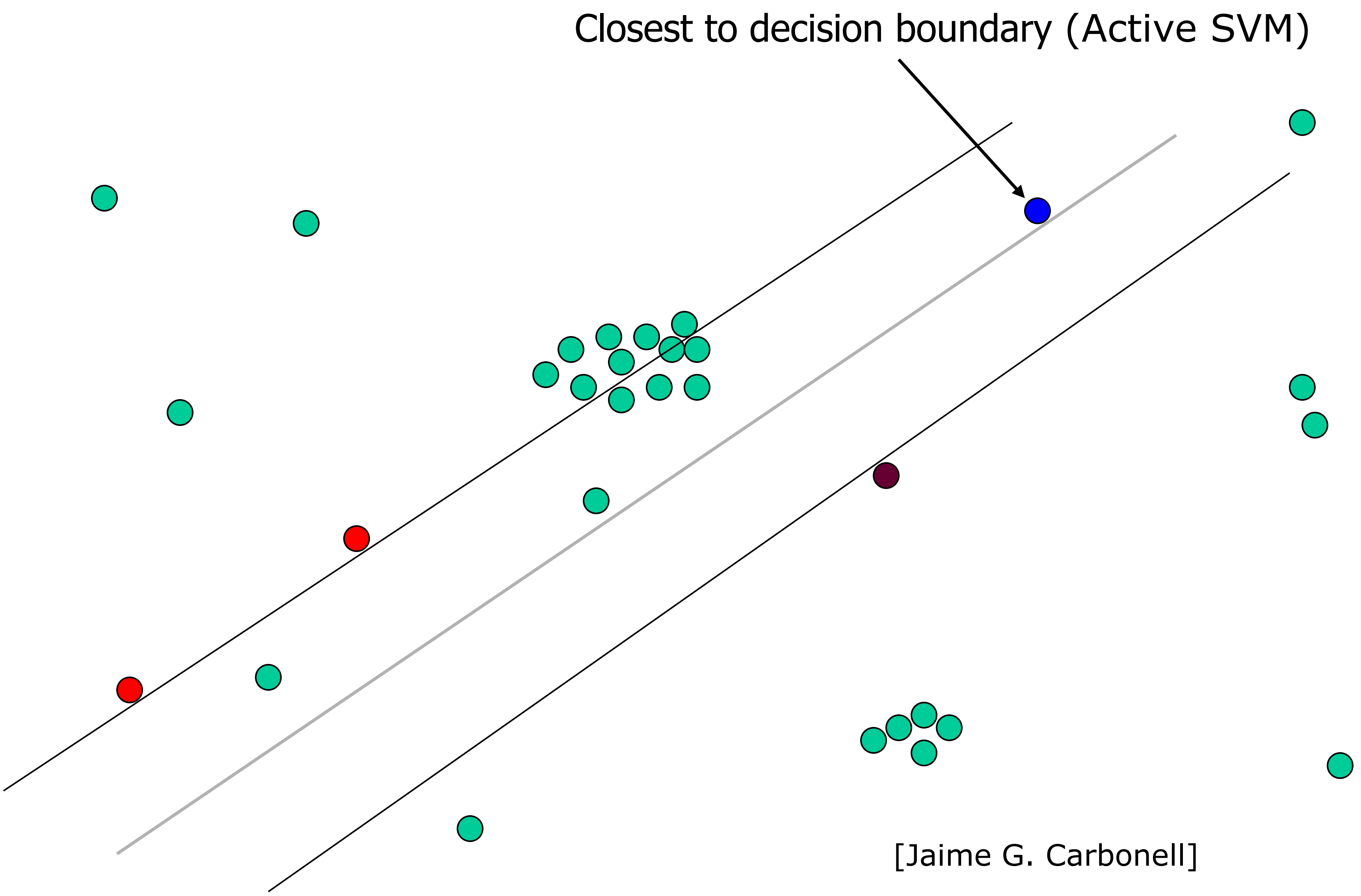
Interesting open question to analyze under what conditions they are successful.

Density-Based Sampling



[Jaime G. Carbonell]

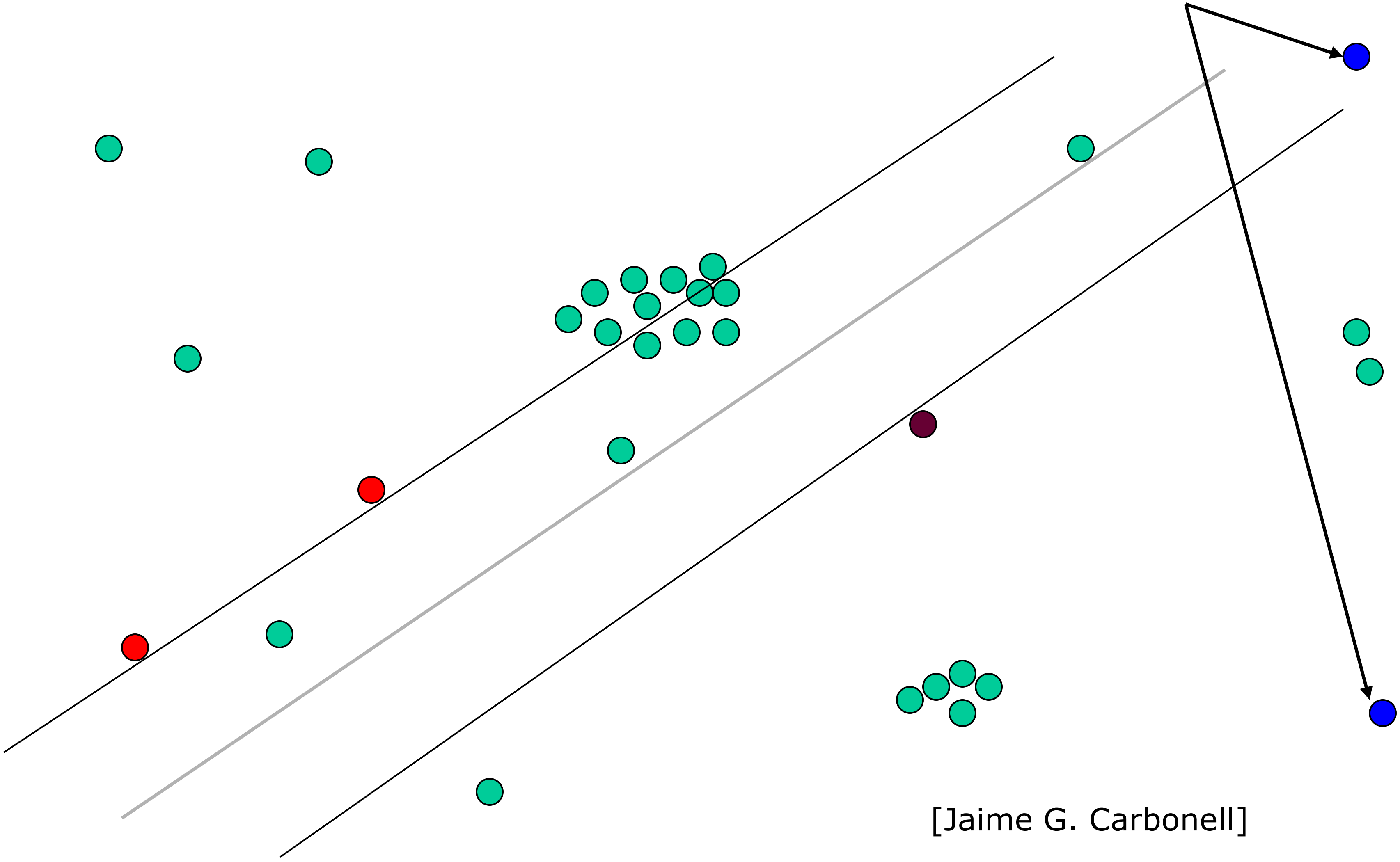
Uncertainty Sampling



Closest to decision boundary (Active SVM)

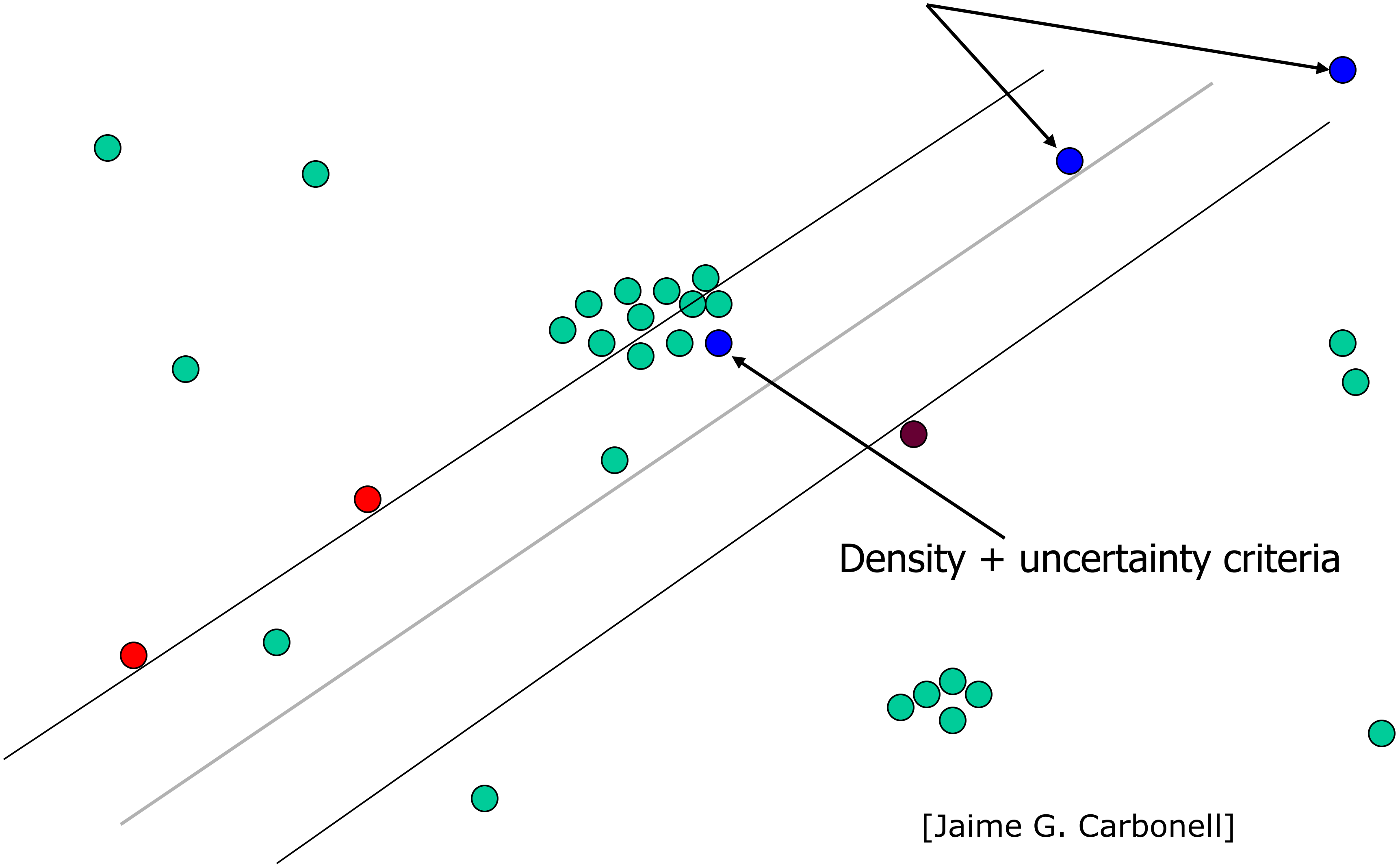
[Jaime G. Carbonell]

Maximal Diversity Sampling


Maximally distant from labeled x’s

[Jaime G. Carbonell]

Ensemble-Based Possibilities



Uncertainty + Diversity criteria

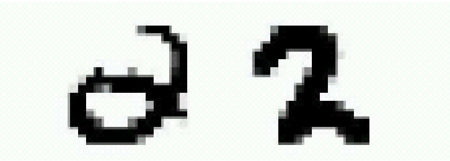
Density + uncertainty criteria

[Jaime G. Carbonell]

Graph-based Active and Semi-Supervised Methods

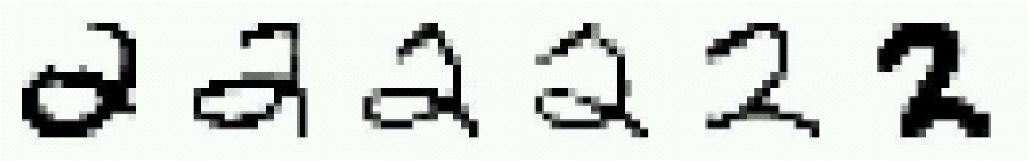
Graph-based Methods

- Assume we are given a pairwise similarity fnc and that very similar examples probably have the same label.
- If we have a lot of labeled data, this suggests a Nearest-Neighbor type of algorithm.
- If you have a lot of unlabeled data, perhaps can use them as "stepping stones".



not similar

E.g., handwritten digits [Zhu07]:

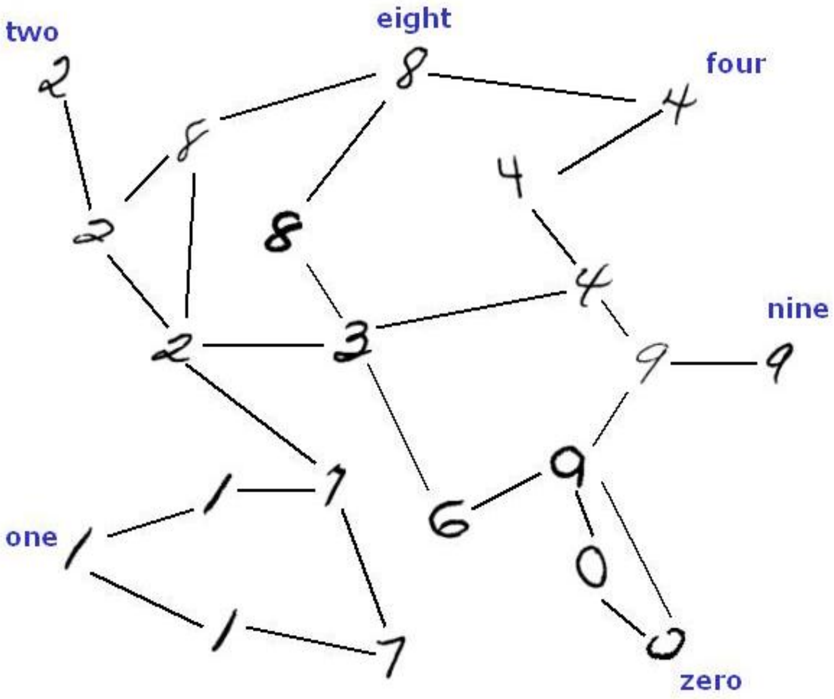


'indirectly' similar
with stepping stones

Graph-based Methods

Idea: construct a graph with edges between very similar examples.

Unlabeled data can help “glue” the objects of the same class together.



Graph-based Methods

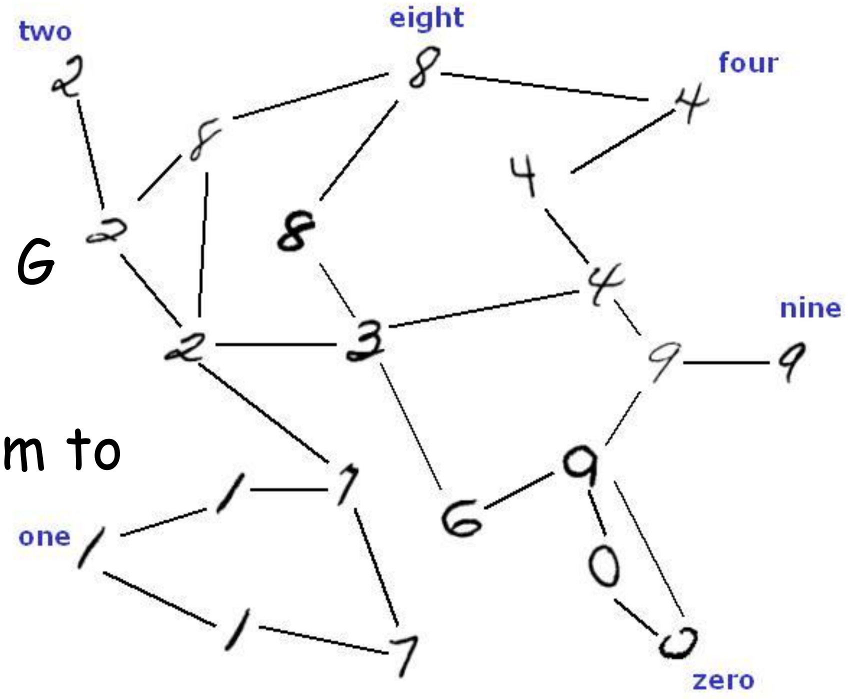
Often, transductive approach. (Given L + U, output predictions on U). Are alllowed to output any labeling of $L \cup U$.

Main Idea:

- Construct graph G with edges between very similar examples.

- Might have also glued together in G examples of different classes.

- Run a graph partitioning algorithm to separate the graph into pieces.



Several methods:

- Minimum/Multiway cut [Blum&Chawla01]
- Minimum "soft-cut" [ZhuGhahramaniLafferty'03]
- Spectral partitioning
- …

SSL using soft cuts

[ZhuGhahramaniLafferty'03]

$f(x) \in \{-1, 1\}$

Solve for label function $\underline{f(x) \in [\overset{-1}{\cancel{0}},1]}$ to minimize:

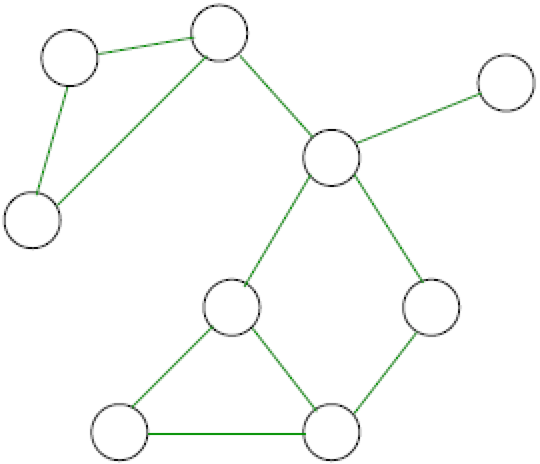
$$\min_f \quad J(f) = \underbrace{\sum_{edges\ (i,j)} \overset{\underline{x_i \leftarrow x_j}}{\underline{w_{ij}}}\left(f(x_i) - f(x_j)\right)^2}_{\text{Similar nodes get similar labels (weighted similarity)}} + \underbrace{\sum_{x_i \in L} \lambda\left(f(x_i) - y_i\right)^2}_{\text{Agreement with labels (agreement not strictly enforces)}}$$

Similar nodes get
similar labels
(weighted similarity)

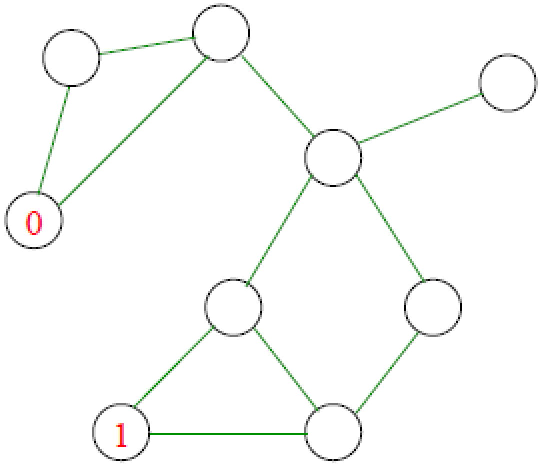
Agreement with labels
(agreement not strictly enforces)

Active learning with label propagation

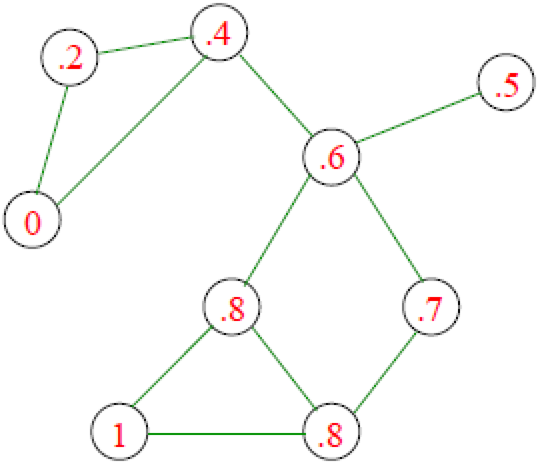
(1) Build neighborhood graph



(2) Query some random points

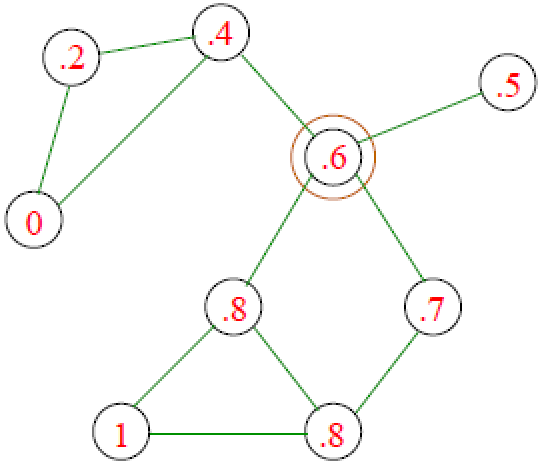


(3) Propagate labels (using soft-cuts)



How to choose which node to query?

(4) Make query and go to (3)



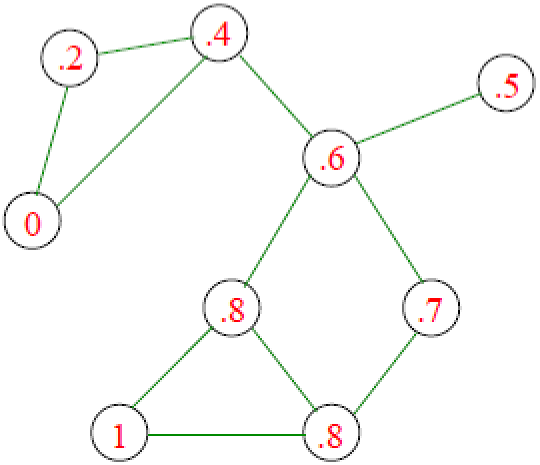
Active learning with label propagation

One natural idea: query the most uncertain point.

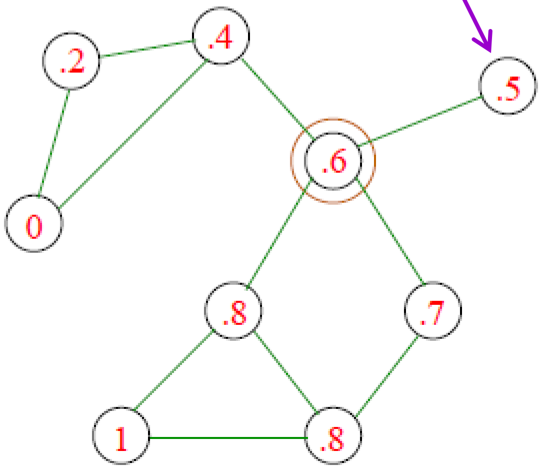
But this has only one edge. Query won't have much impact!

(even worse: a completely isolated node)

(3) Propagate labels (using soft-cuts)



(4) Make query and go to (3)



Active learning with label propagation

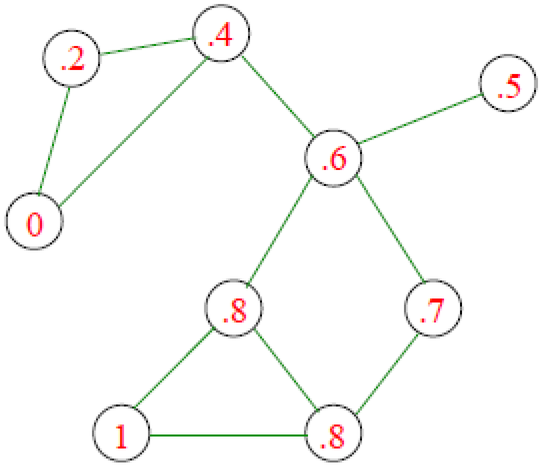
Instead, use a 1-step-lookahead heuristic:

- For a node with label p, assume that querying will have prob p of returning answer 1, $1-p$ of returning answer 0.
- Compute "average confidence" after running soft-cut in each case:

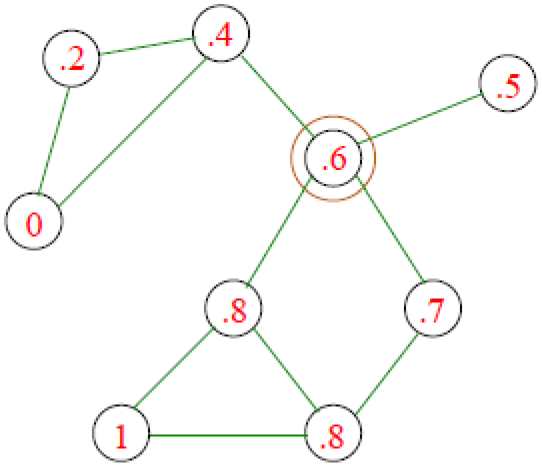
$$p\frac{1}{n}\sum_{x_i}\max(f_1(x_i), 1-f_1(x_i)) + (1-p)\frac{1}{n}\sum_{x_i}\max(f_0(x_i), 1-f_0(x_i))$$

- Query node s.t. this quantity is highest (you want to be more confident on average).

(3) Propagate labels (using soft-cuts)

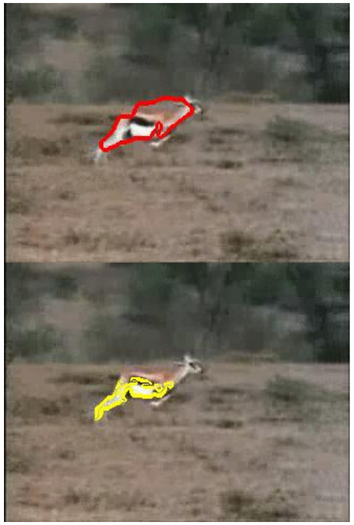


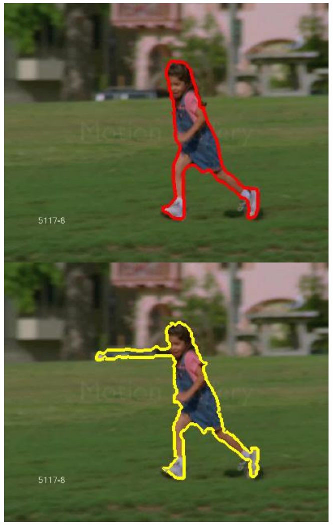
(4) Make query and go to (3)



Active Learning with Label Propagation in Practice

- Does well for Video Segmentation (Fathi-Balcan-Ren-Regh, BMVC 11).





What You Should Know

- Active learning could be really helpful, could provide exponential improvements in label complexity (both theoretically and practically)!

- Common heuristics (e.g., those based on uncertainty sampling). Need to be very careful due to sampling bias.

- Safe Disagreement Based Active Learning Schemes.
    - Understand how they operate precisely in noise free scenarios.