Active Learning

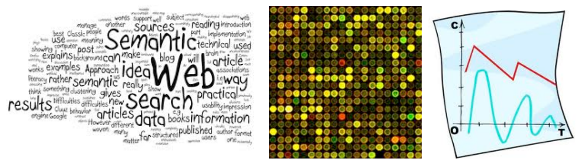
Burr Settles

Machine Learning 10-701 / 15-781
April 19, 2011

some slides adapted from: Aarti Singh, Rui Castro, Rob Nowak

MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science

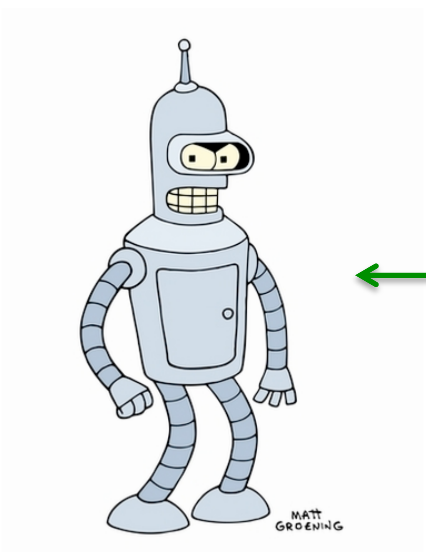
Supervised Learning



raw unlabeled data

$x_1, x_2, x_3, \ldots$

random sample



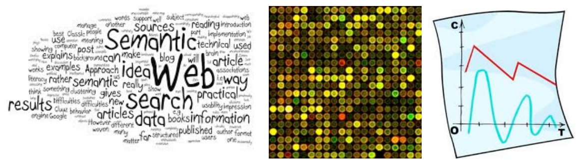
labeled training instances

$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \langle x_3, y_3 \rangle, \ldots$

supervised learner
induces a classifier

expert / oracle
analyzes experiments
to determine labels

Semi-Supervised Learning



exploit the structure in unlabeled data

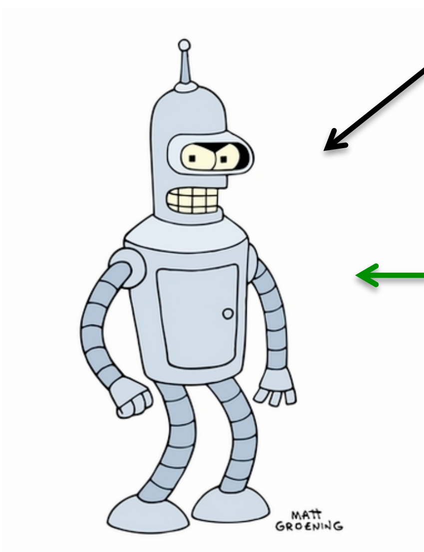
raw unlabeled data

$x_1, x_2, x_3, \ldots$

random sample

labeled training instances

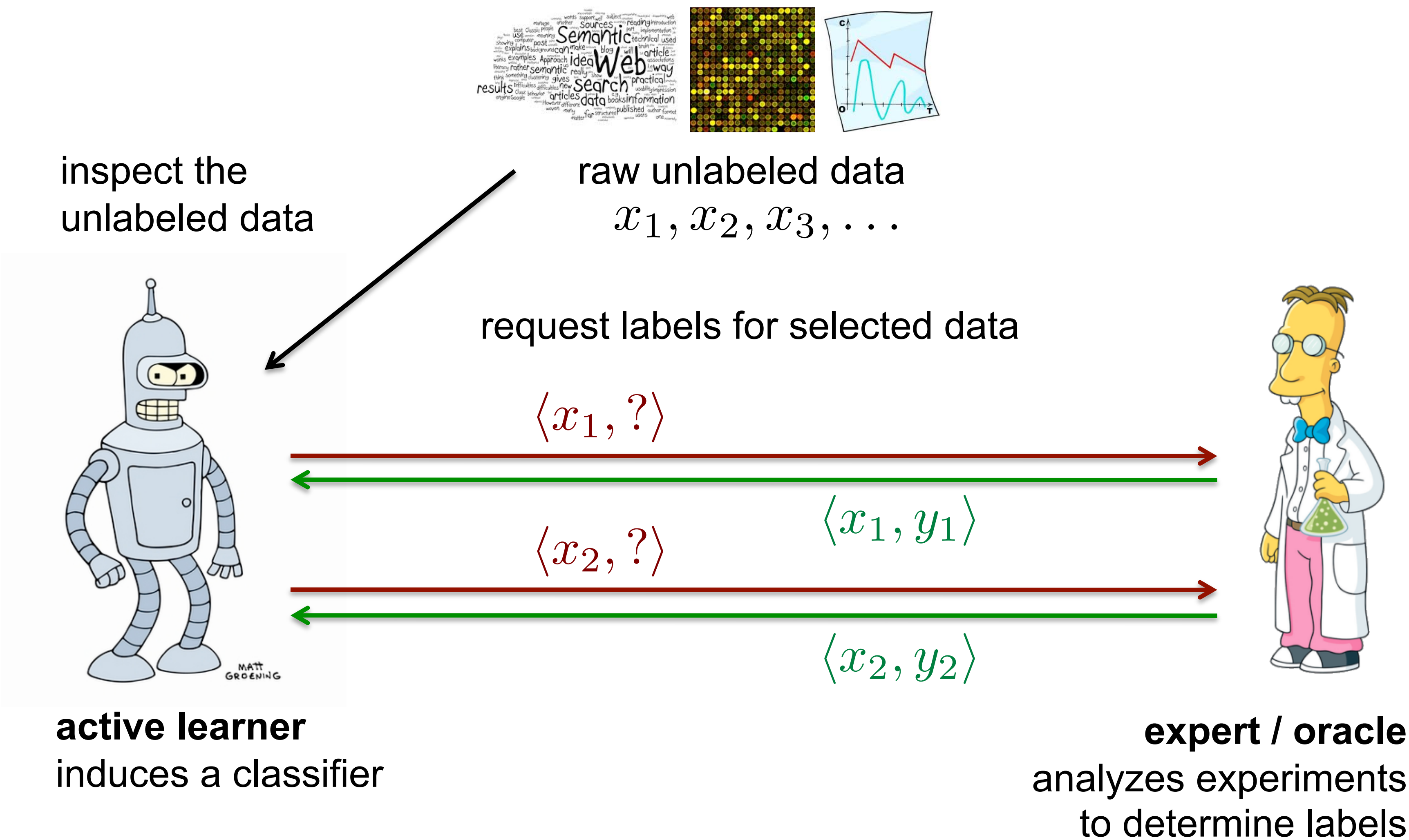
$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \langle x_3, y_3 \rangle, \ldots$



semi-supervised learner
induces a classifier

expert / oracle
analyzes experiments to determine labels

Active Learning



The 20 Questions Game

“Are you female?”
“No.”

“Do you have a moustache?”
“Yes.”

our goal is to pose the
most informative “queries”

how can we automate this process?

Thought Experiment

- suppose you are on an Earth convoy sent to colonize planet Zelgon

people who ate the smooth Zelgian fruits found them *tasty!*

people who ate the spikey Zelgian fruits ***got sick****!*

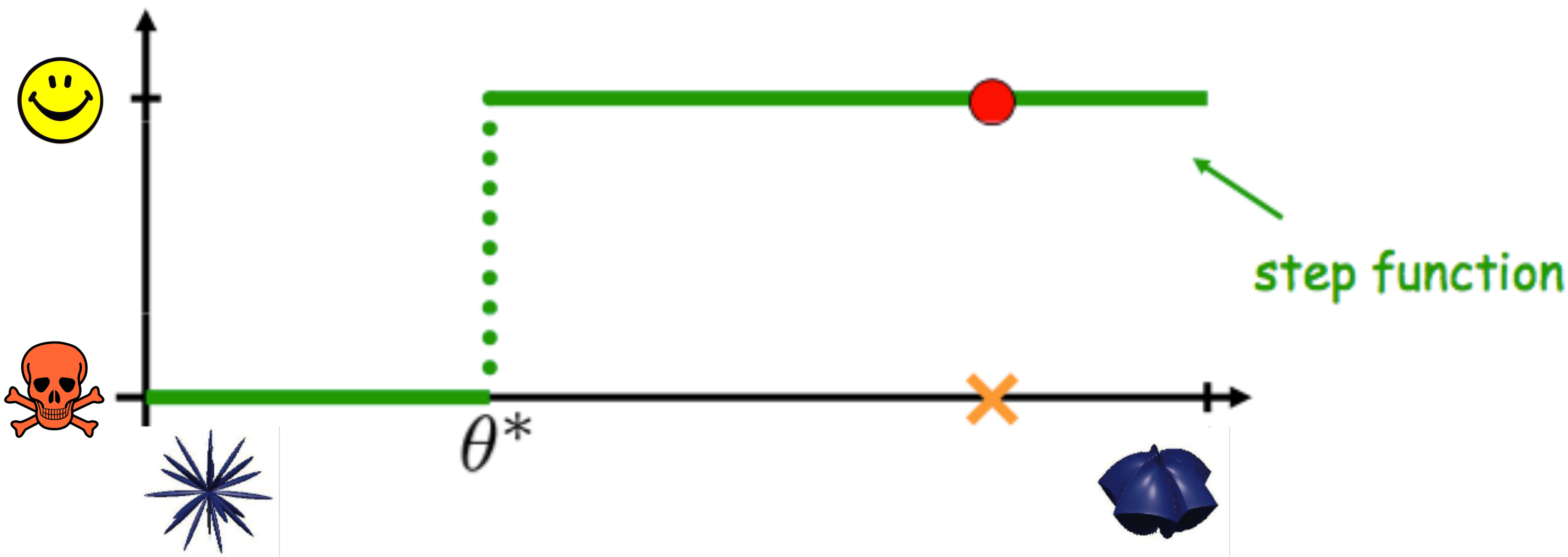
Determining Poison vs. Yummy Fruits

- there is a continuous range of spikey-to-smooth fruit shapes on Zelgon:

you need to learn how to recognize fruits as poisonous or safe

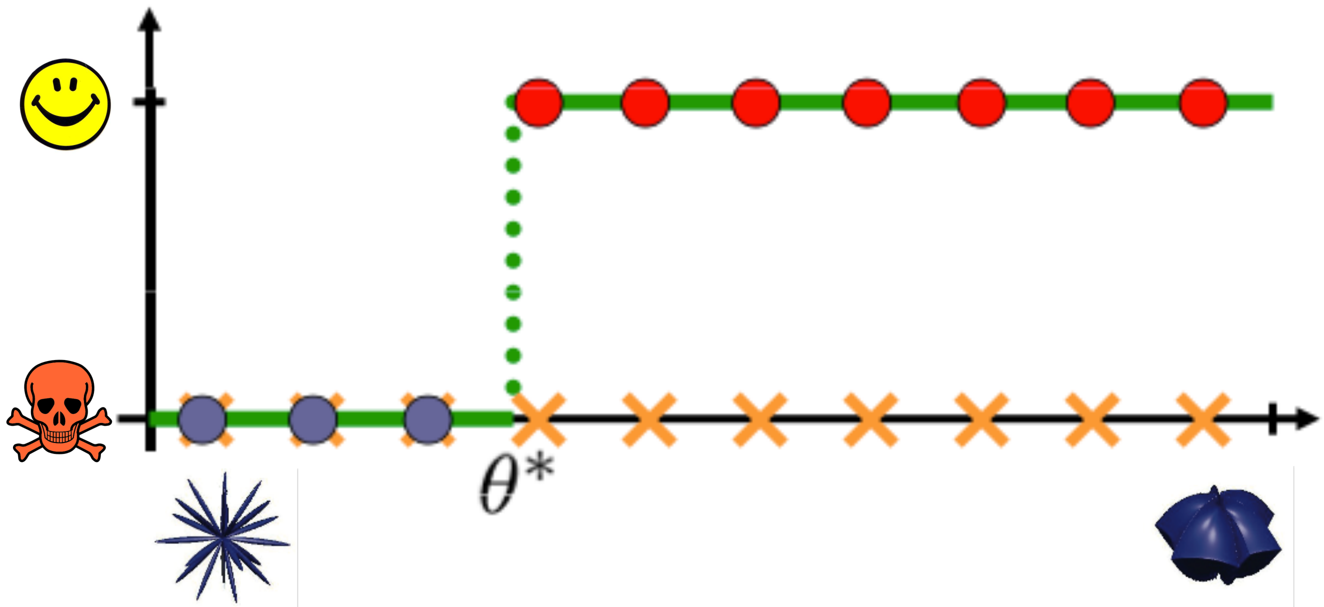
and you need to do this while risking as little as possible (i.e., colonist health)

Learning a Change Point



goal: learn threshold θ* as accurately as possible,
using as few labeled instances as possible.

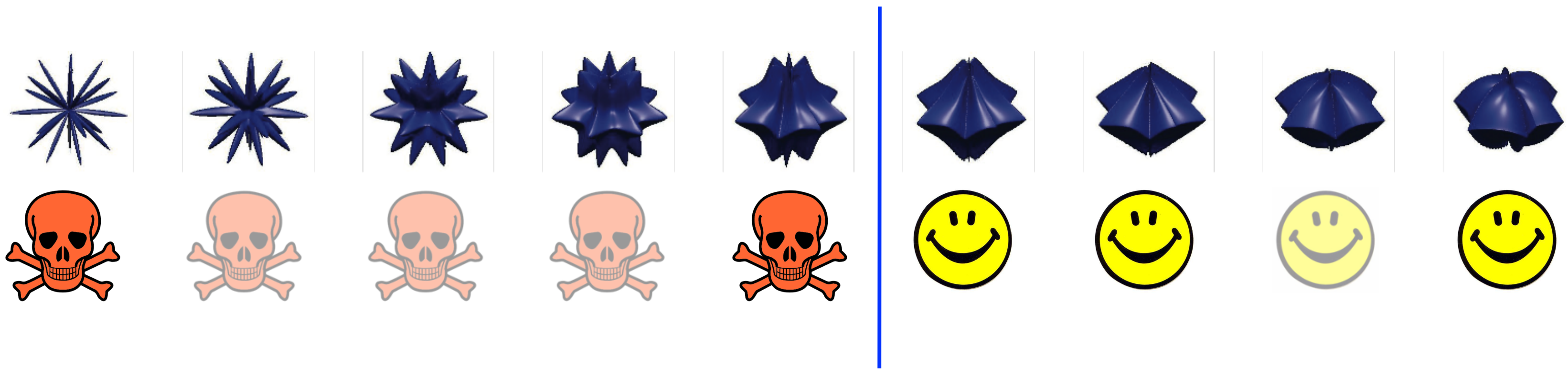
The Problem with Passive Learning



in passive supervised learning, the instances must be chosen before any “tests” are done!

error rate ε requires us to risk $O(1/\varepsilon)$ people’s health!

Can We Do Better?



this is just a **binary search**...

requiring $O(1/\varepsilon)$ fruits (samples)
and only $O(\log_2 1/\varepsilon)$ tests (queries)

your first "active learning" algorithm!

Relationship to Active Learning

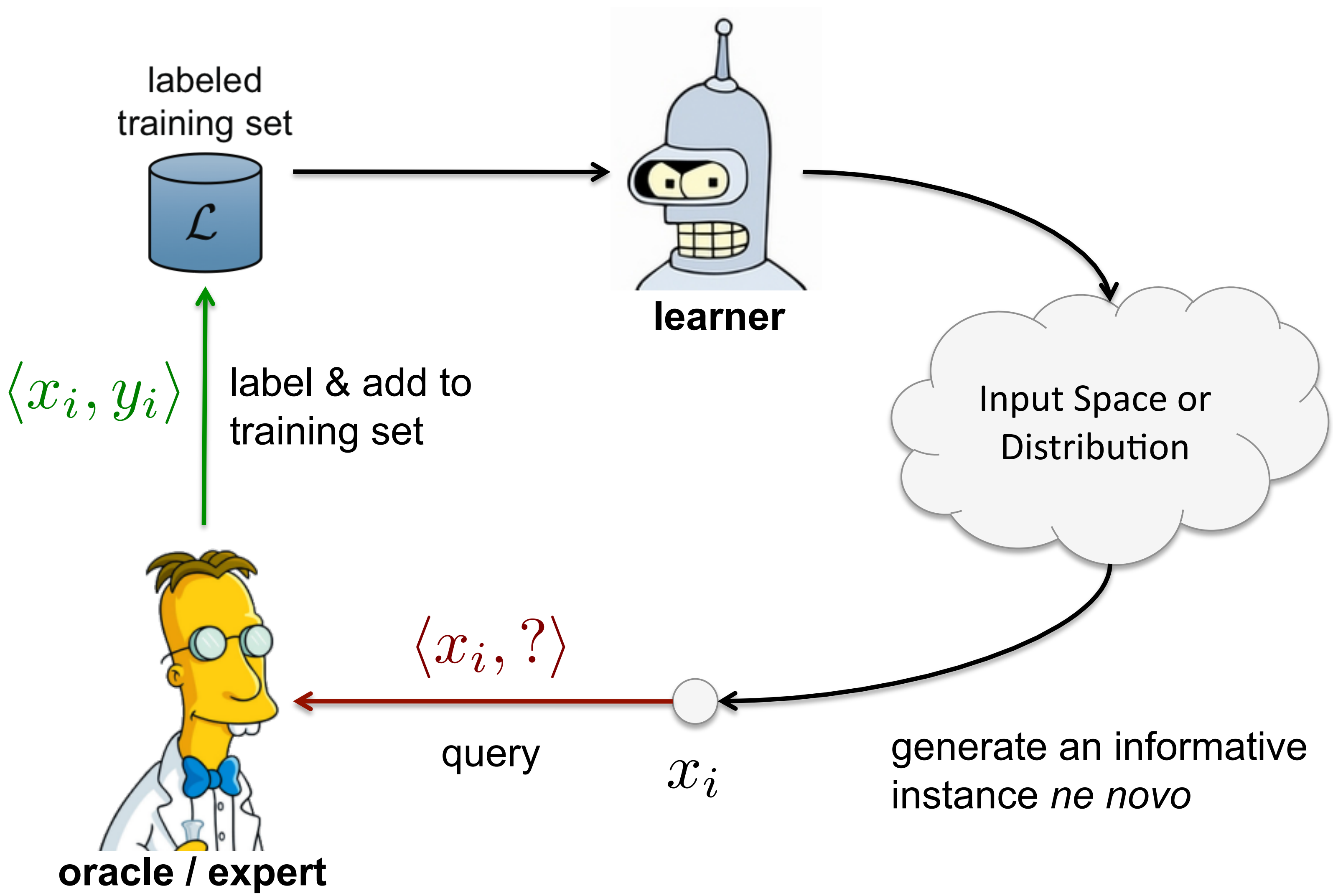
- **key idea:** the learner chooses the training data
  - on Zelgon: whether a fruit was poisonous/safe
  - *in general*: the true label of some instance

- **goal:** reduce the training costs
  - on Zelgon: the number of “lives at risk”
  - *in general*: the number of “queries”
    (=> labor costs, disk storage space, etc.)

Practical Query Scenarios

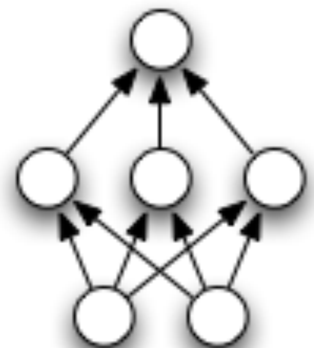
- query synthesis [Anguin, 1988]
- selective sampling [Atlas et al., 1989]
- pool-based active learning [Lewis & Gale, 1994]

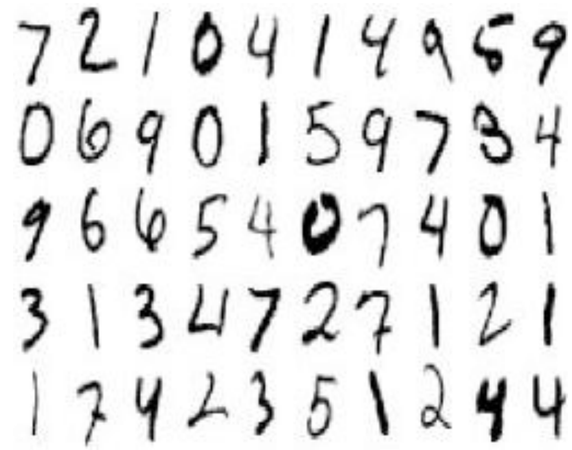
Query Synthesis

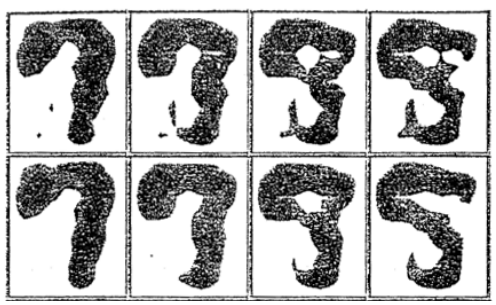


Problems with Query Synthesis

an early real-world application: neural-net queries synthesized for handwritten digits [Lang & Baum, 1992]



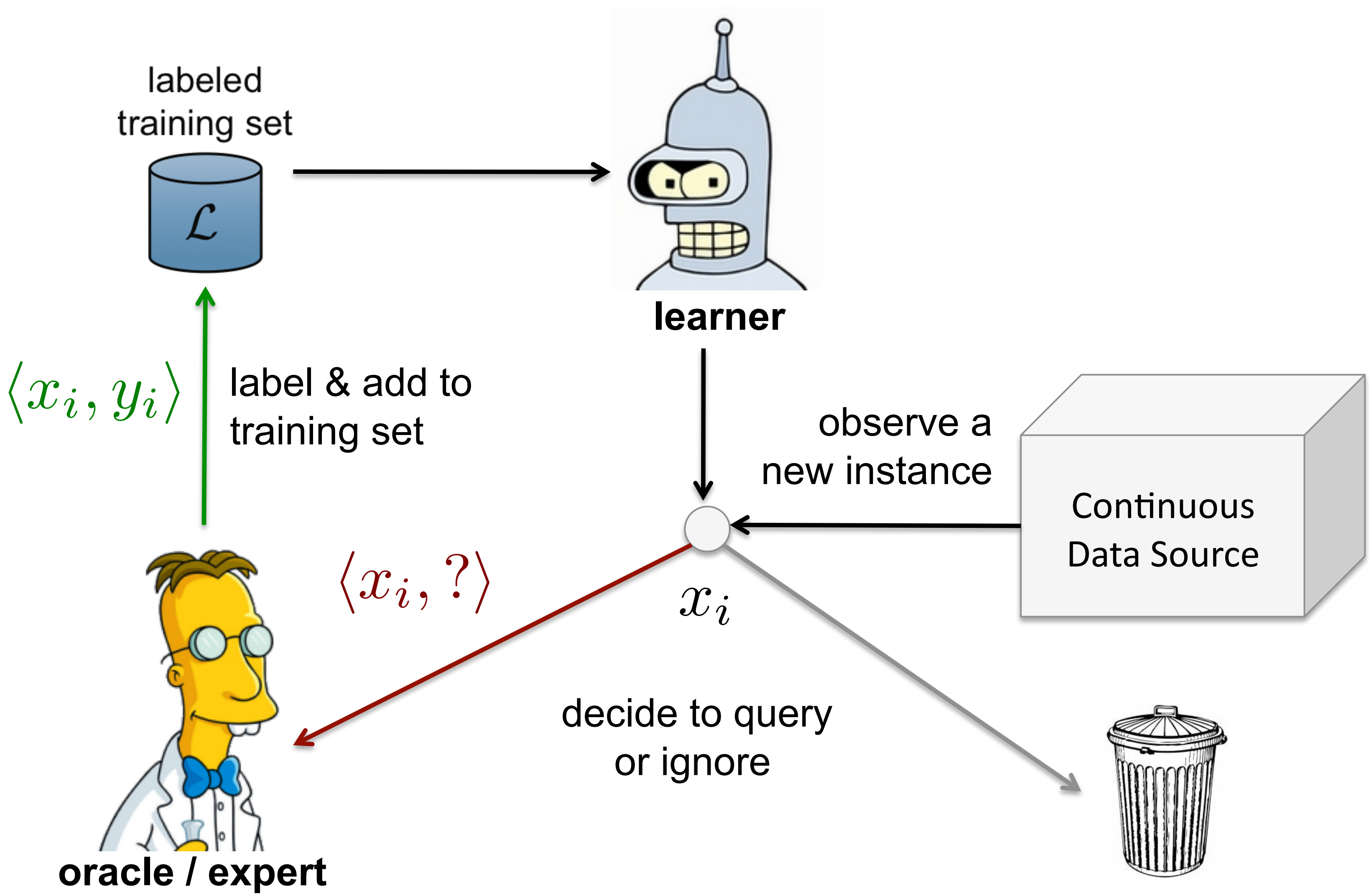




problem: humans couldn't interpret the queries!

ideally, we can ensure that the queries come from the underlying "natural" distribution

Selective Sampling

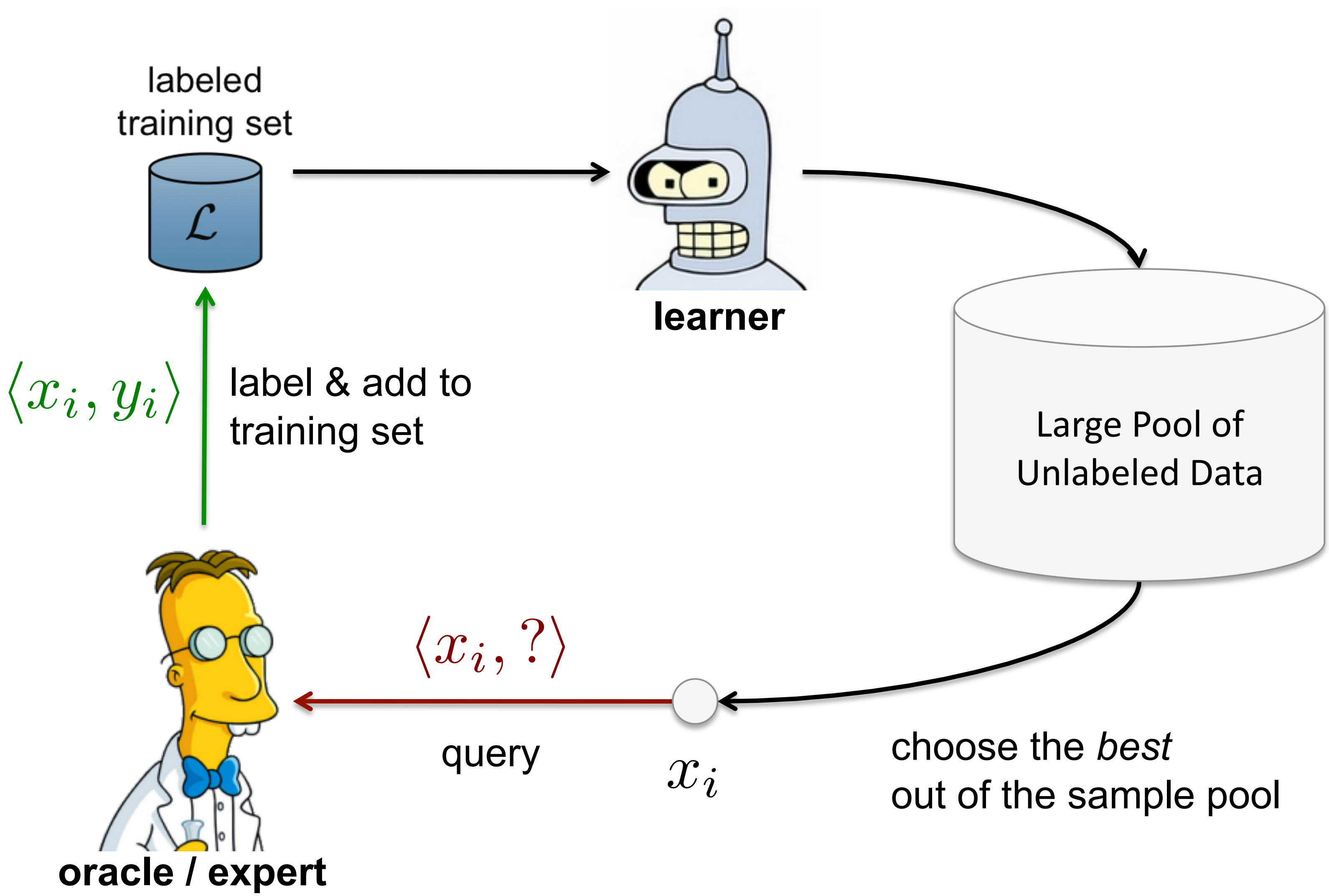


Selective Sampling (2)

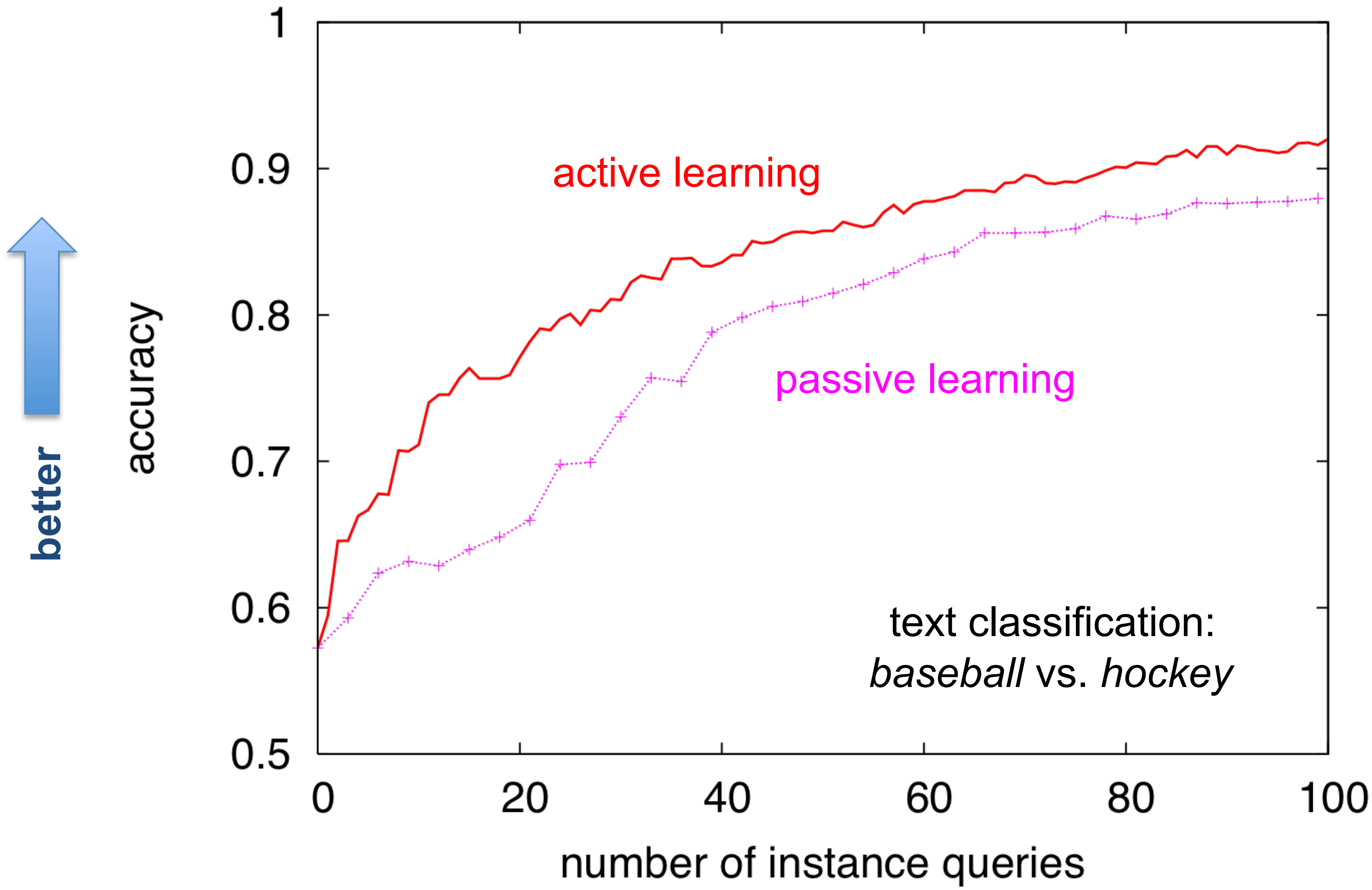
- **advantage**: ensures that query instances come from the true underlying data distribution

- **assumption**: drawing an instance from the distribution is significantly less expensive than obtaining its label
  – often true in practice, e.g., downloading Web documents vs. assigning topic labels to them

Pool-Based Active Learning



Learning Curves



Who Uses Active Learning?

Sentiment analysis for blogs; Noisy relabeling
– *Prem Melville*

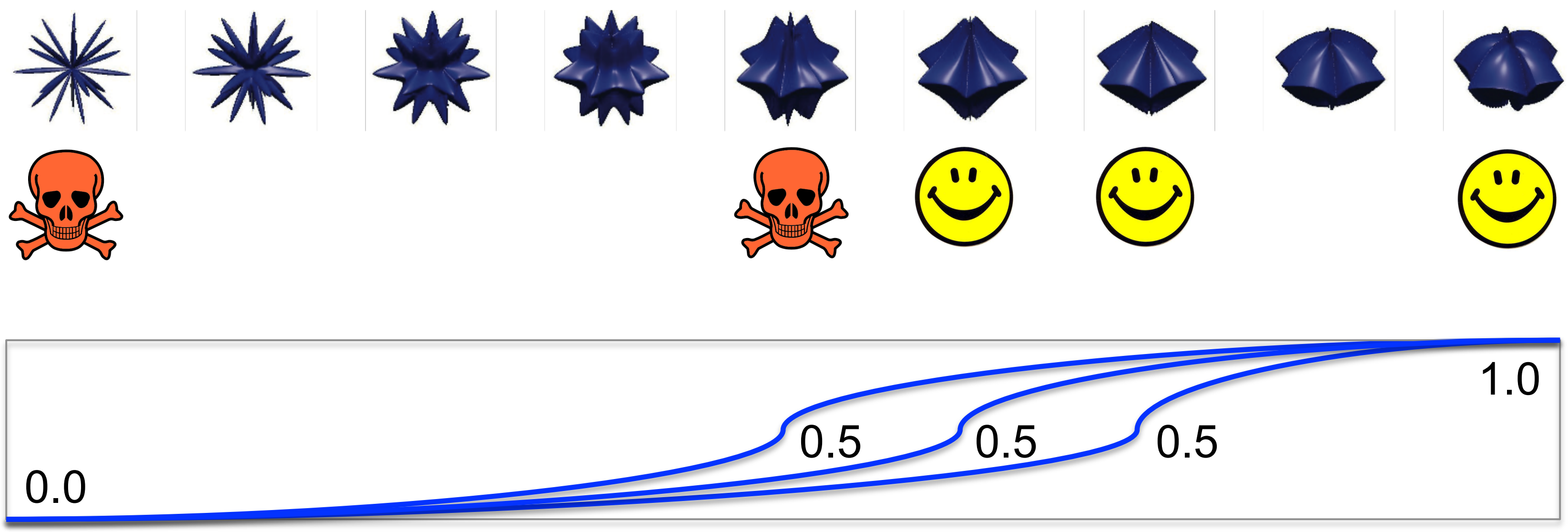
Biomedical NLP & IR; Computer-aided diagnosis
– *Balaji Krishnapuram*

MS Outlook voicemail plug-in [Kapoor et al., IJCAI'07]; "A variety of prototypes that are in use throughout the company." – *Eric Horvitz*

"While I can confirm that we're using active learning in earnest on many problem areas… I really can't provide any more details than that. Sorry to be so opaque!"
– *David Cohn*

OK, How Do We Select Queries?

- let's interpret our Zelgian fruit binary search in terms of a *probabilistic* classifier:

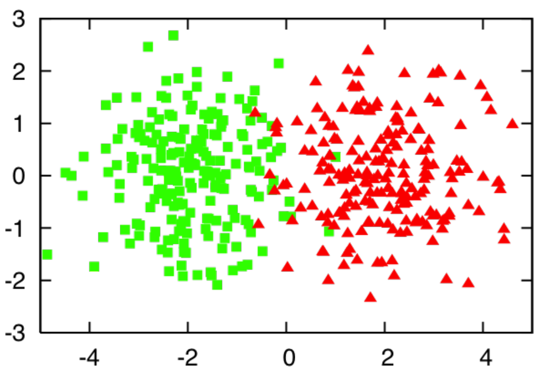


$P(Y = \text{☺} | X)$

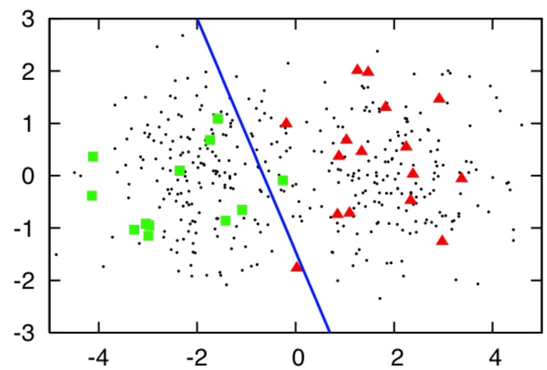
[Lewis & Gale, SIGIR'94]

Uncertainty Sampling

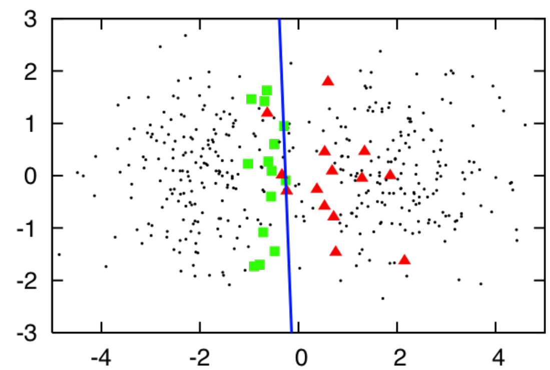
- query instances the learner is *most uncertain* about



400 instances sampled from 2 class Gaussians



random sampling
30 labeled instances
(accuracy=0.7)



uncertainty sampling
30 labeled instances
(accuracy=0.9)

Uncertainty Measures

least confident

$$\phi_{LC}(x) = 1 - P_\theta(y^*|x)$$

smallest-margin

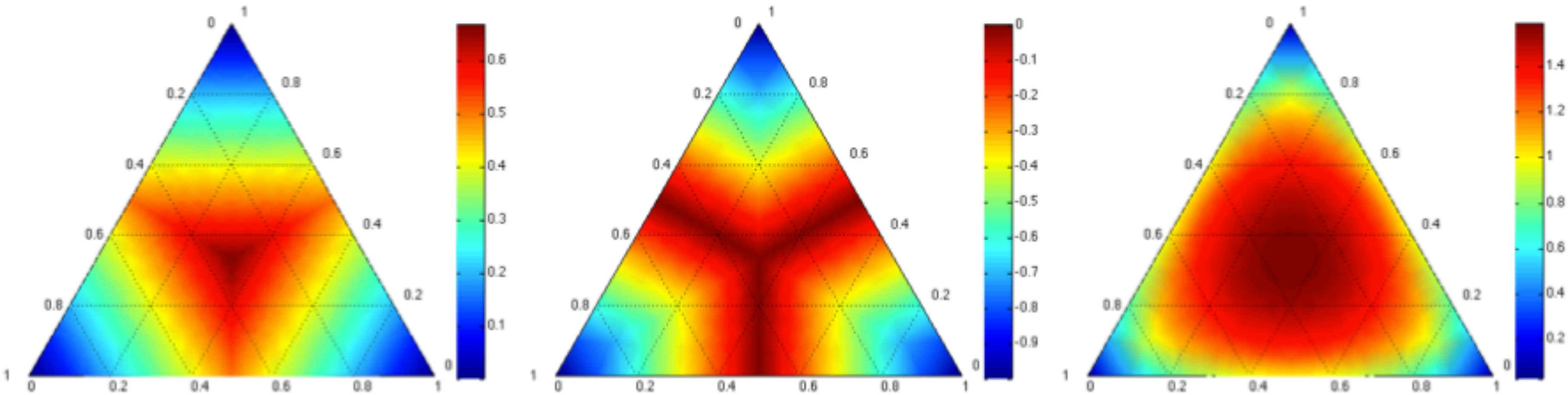
$$\phi_M(x) = P_\theta(y_1^*|x) - P_\theta(y_2^*|x)$$

entropy

$$\phi_{ENT}(x) = -\sum_y P_\theta(y|x)\log_2 P_\theta(y|x)$$

note: for binary tasks, these are equivalent!

Multi-Class Uncertainty



(a) least confident (b) margin (c) entropy

illustration of preferred (dark red) posterior
distributions in a 3-label classification task

note: for multi-class tasks, these are not equivalent!

Information-Theoretic Interpretation

- the "surprisal" $\mathcal{I}$ is a measure (in bits, nats, etc.) of the information content for outcome y of variable Y:

$$\mathcal{I}(y) = \log \frac{1}{P(y)} = -\log P(y)$$

- so this is how "informative" the oracle's label y will be
- but the learner doesn't know the oracle's answer yet! we can estimate it as an *expectation* over all possible labels:

$$E_y \left[-\log P_\theta(y|x)\right] = -\sum_y P_\theta(y|x) \log P_\theta(y|x)$$

- which is **entropy**-based uncertainty sampling

Uncertainty Sampling in Practice

- pool-based active learning:
  - evaluate each x in $\mathcal{U}$
  - rank and query the top K instances
  - retrain, repeat
- selective sampling:
  - threshold a "region of uncertainty," e.g., [0.2, 0.8]
  - observe new instances, but only query those that fall within the region
  - retrain, repeat

Simple and Widely-Used

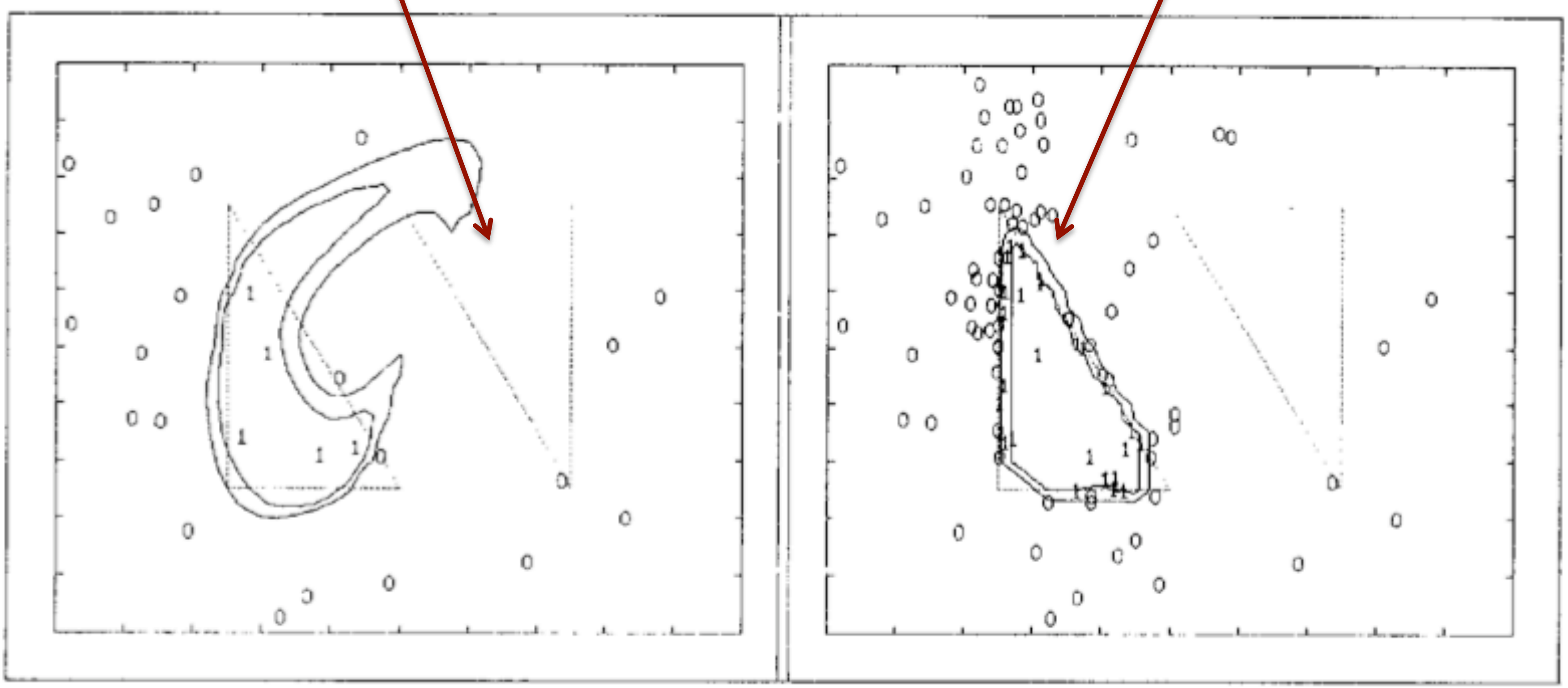
- **text classification**
  - Lewis & Gale ICML'94;
- **POS tagging**
  - Dagan & Engelson, ICML'95; Ringger et al., ACL'07
- **disambiguation**
  - Fujii et al., CL'98;
- **parsing**
  - Hwa, CL' 04
- **information extraction**
  - Scheffer et al., CAIDA'01; Settles & Craven, EMNLP'08
- **word segmentation**
  - Sassano, ACL'02
- **speech recognition**
  - Tur et al., SC'05
- **transliteration**
  - Kuo et al., ACL'06
- **translation**
  - Haffari et al., NAACL'09

[Cohn et al., ML'94]

Uncertainty Sampling FAIL!

initial random sample
misses the right triangle

neural net uncertainty sampling
only queries the left side



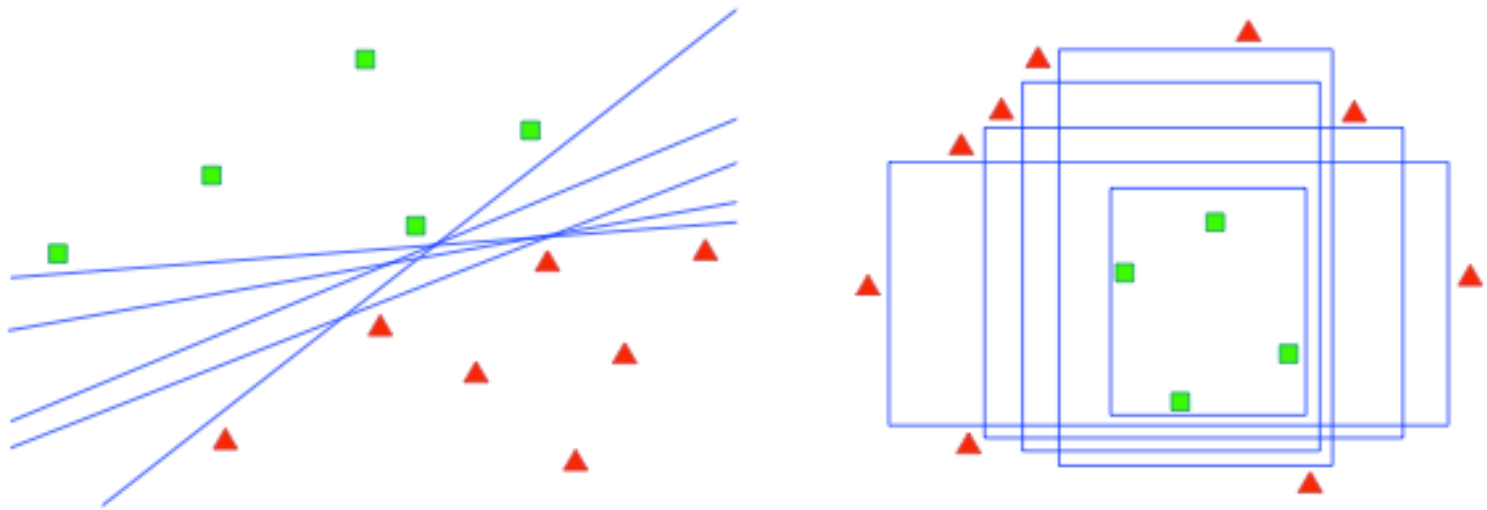
What To Do?

- plain uncertainty sampling only uses the confidence of a *single* classifier
  - sometimes called a “point estimate” for parametric models
  - this classifier can become overly confident about instances is really knows nothing about!

- instead, let’s consider a different notion of “uncertainty”… about the *classifier itself*

Remember Version Spaces?

- the set of all classifiers that are consistent with the labeled training data

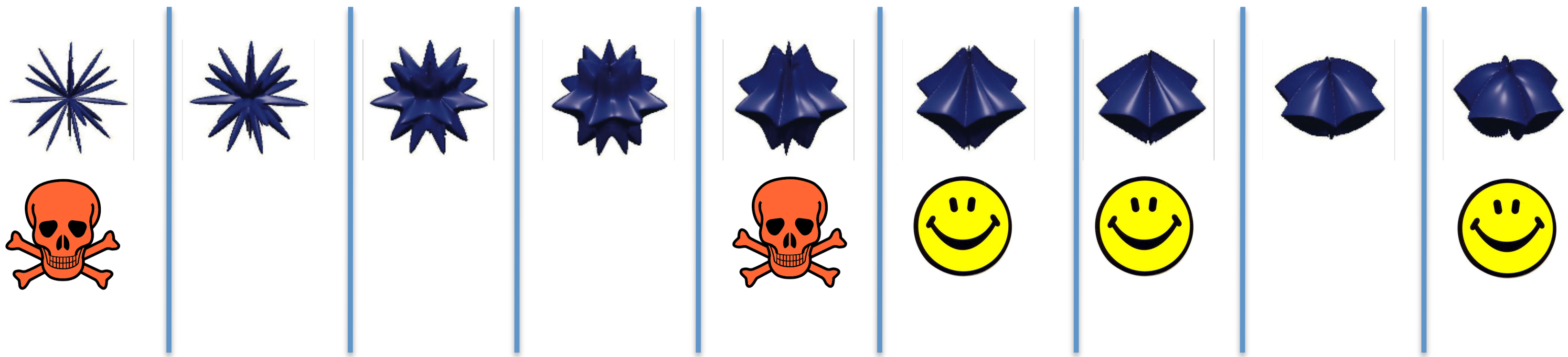


(a) (b)

- the larger the version space $\mathcal{V}$, the less likely each possible classifier is… we want queries to *reduce* $|\mathcal{V}|$

Alien Fruits Revisited

- let's try interpreting our binary search in terms of a version-space search:



possible classifiers (thresholds): 1

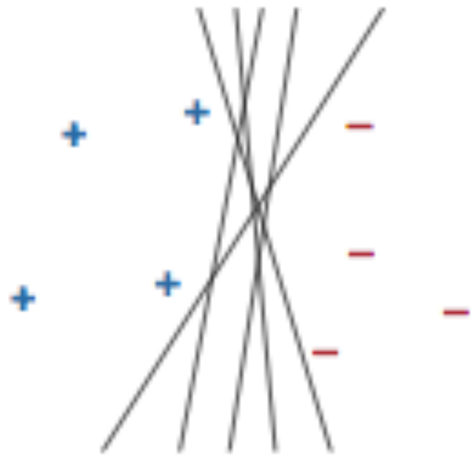
Simple Version Space Algorithm

- enumerate all legal hypotheses
  - or compute $|\mathcal{V}|$ analytically
- the optimal query is the one that most reduces the size of $\mathcal{V}$ (in expectation over y):

$$x^*_{VS} = \arg\min_x E_y \left| \mathcal{V}^{\mathcal{L} \cup \langle x, y \rangle} \right|$$

- ideally we can *halve* the size of the version space
- binary search does this in 1D (e.g., Zelgian fruits)

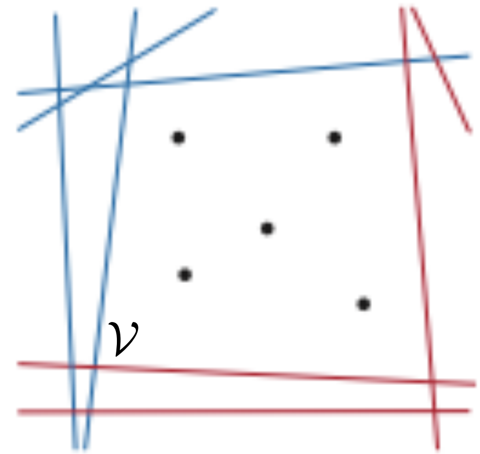
Version Spaces for SVMs



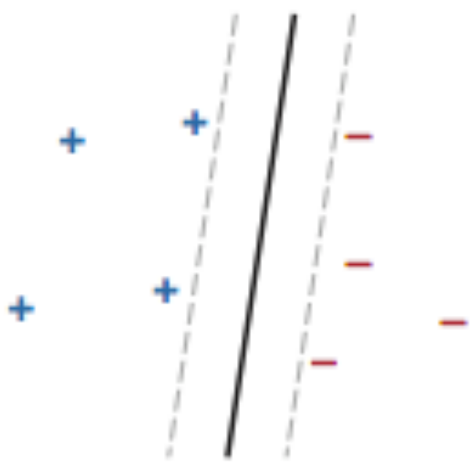
$\mathcal{F}$ (feature space)

"version space duality"
(Vapnik, 1998)

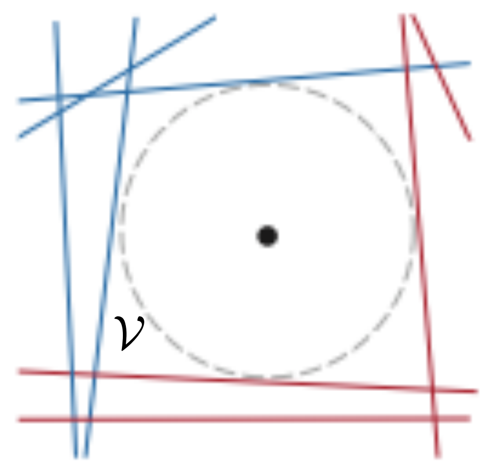
points in $\mathcal{F}$ correspond to hyperplanes in $\mathcal{H}$ and *vice versa*



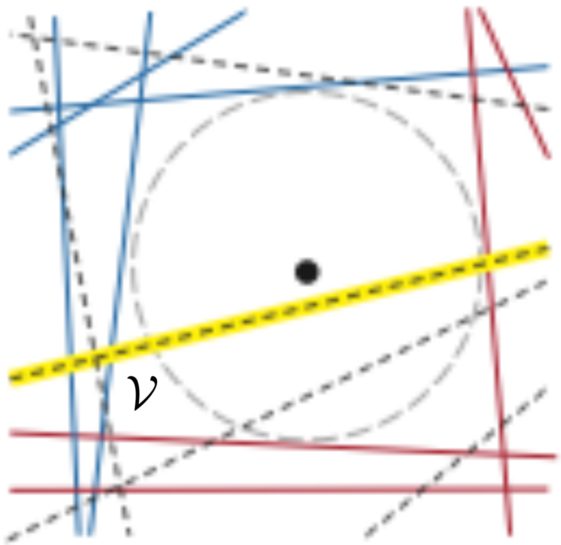
$\mathcal{H}$ (hypothesis space)



SVM with largest margin is the center of the largest hypersphere in $\mathcal{V}$



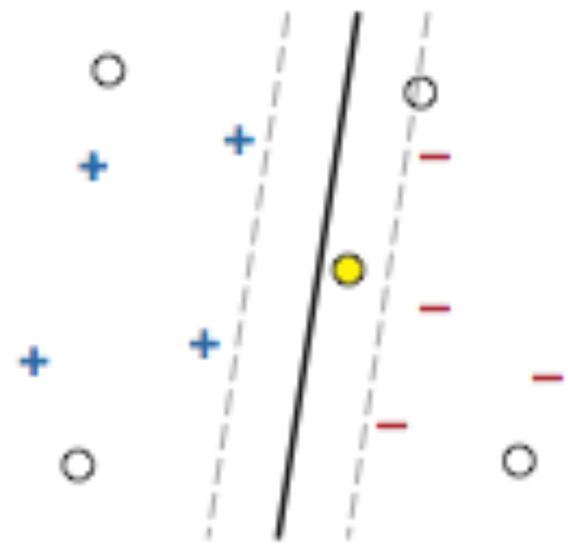
Bisecting the SVM Version Space



given a choice of unlabeled instances (planes in $\mathcal{H}$), we want to query one that mostly "bisects" $\mathcal{V}$

i.e., the instance that comes closest to the SVM weight vector

this corresponds to the instance closest to the SVM decision boundary, i.e., smallest-margin uncertainty sampling

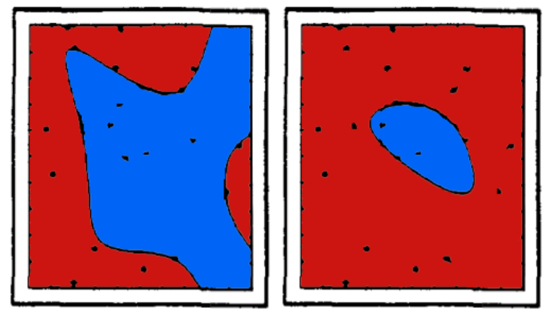


special case for SVMs: the best classifier is (hopefully) the *center* of the version space

Problem: $\mathcal{V}$ Can Be a Big Space

- in general, $\mathcal{V}$ may be too large to enumerate or measure $|\mathcal{V}|$ through analysis or trickery

- idea: train two classifiers G and S which represent the two "extremes" of the version space



- if these two models disagree, the instances falls within the "region of uncertainty"

[Atlas et al., NIPS'89]

Toy Example: Learning A Square

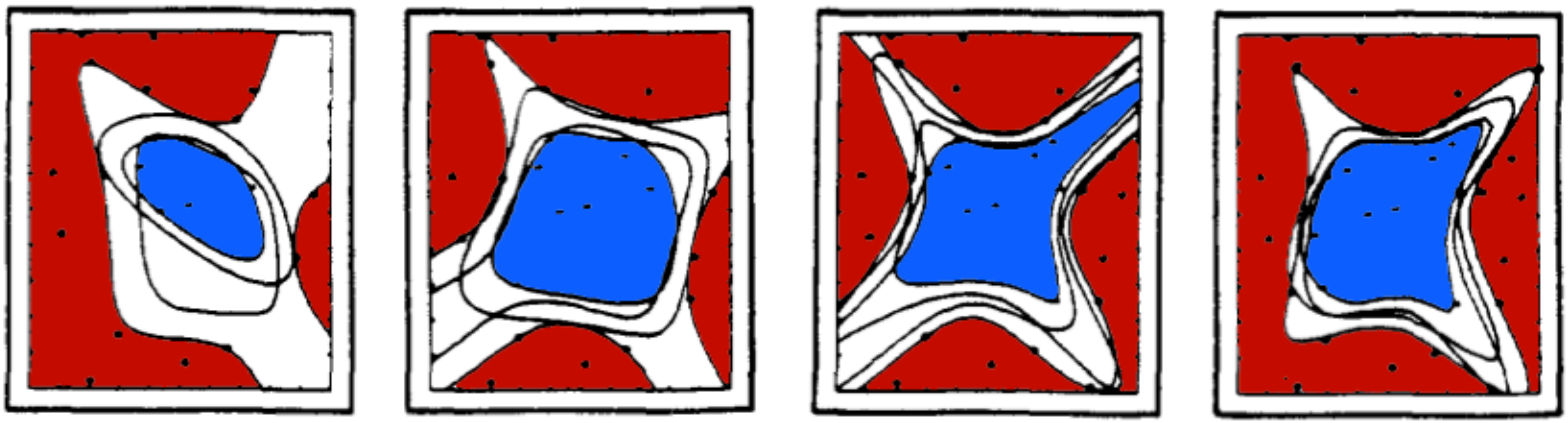


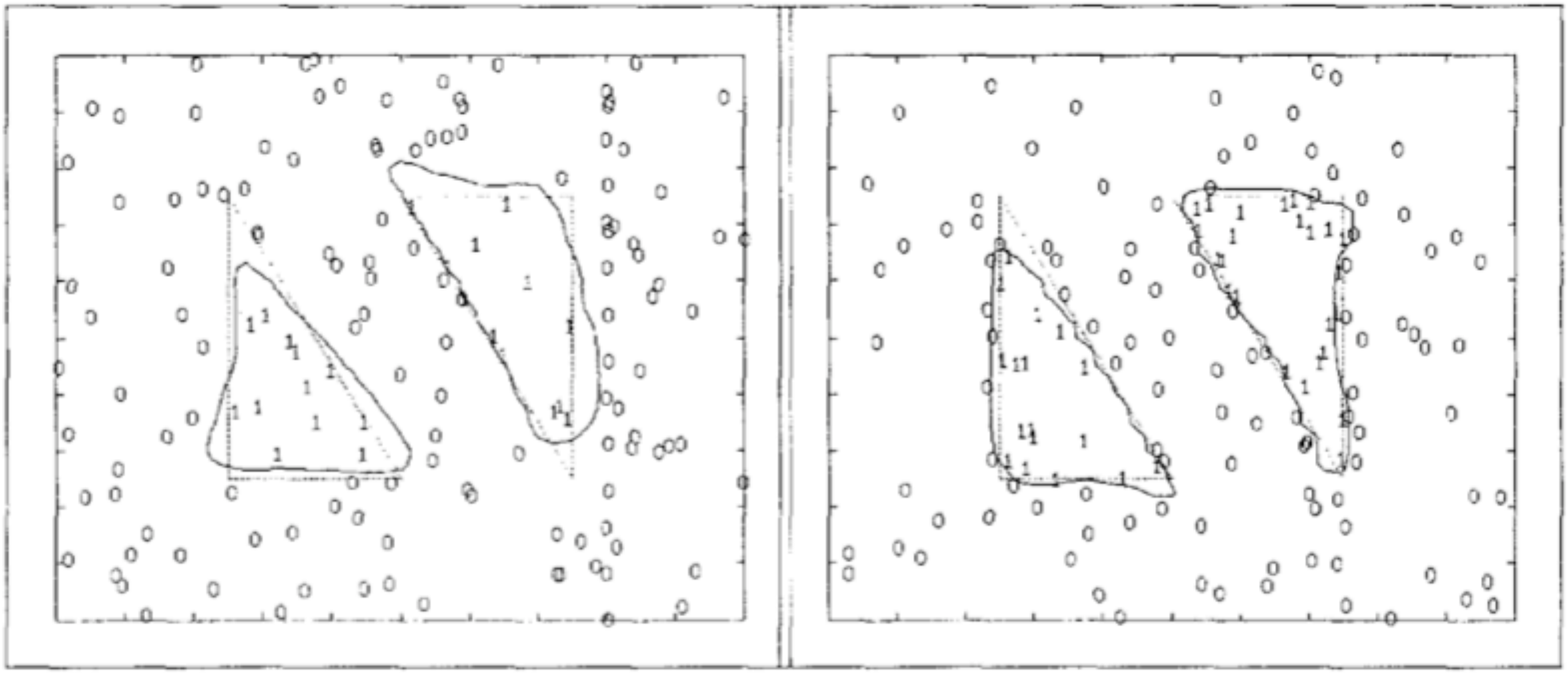
Figure 4: Learning a square by selective sampling

[Cohn et al., ML'94]

Triangles Revisited

150 random samples

150 queries (G & S disagreement)



[Seung et al., COLT'92]

Query-By-Committee (QBC)

- simpler, more general approach

- train a committee of classifiers $\mathcal{C}$
  - no need to maintain G and S
  - committee can be any size (but often just 2)

- query instances for which committee members disagree

[Freund et al., ML'97]

QBC Guarantees

- let d be the VC dimension of hypothesis space
- under certain conditions, QBC achieves prediction error ε with high probability using:
  - $O(d/\varepsilon)$ unlabeled instances
  - $O(\log_2 d/\varepsilon)$ queries
- an exponential improvement!

QBC in Practice

- selective sampling:
  - train a committee $\mathcal{C}$
  - observe new instances, but only query those for which there is disacreement (or a lot of disagreement)
  - retrain, repeat
- pool-based active learning:
  - train a committee C
  - measure disagreement for each x in U
  - rank and query the top K instances
  - retrain, repeat

QBC Design Decisions

- how to build a committee:
  - "sample" models from $P(\theta|\mathcal{L})$
    - [Dagan & Engelson, ICML'95; McCallum & Nigam, ICML'98]
  - standard ensembles (e.g., bagging, boosting)
    - [Abe & Mamitsuka, ICML'98]
- how to measure disagreement (many):
  - "XOR" committee classifications
  - view vote distribution as probabilities, use uncertainty measures (e.g., entropy)

Bayesian Interpretation

- we can use Bayes' rule to derive an estimate of the *ensemble* prediction for a new x:

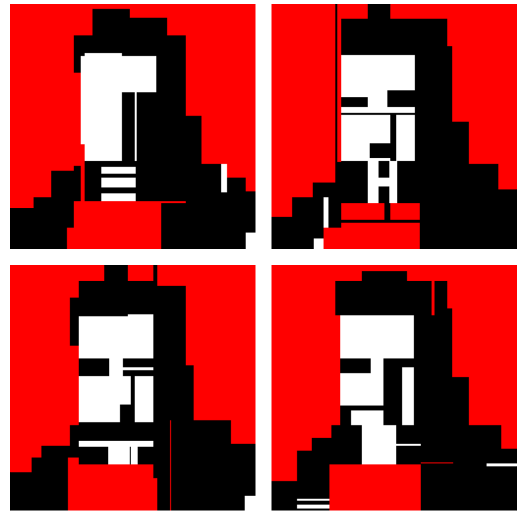
$$P_{\mathcal{C}}(y|x) = \sum_{\theta \in \mathcal{C}} P_\theta(y|x)P(\theta)$$

- QBC attempts to reduce uncertainty over both:
  - the label y
  - the classifier θ

$$\phi_{VE}(x) = -\sum_{y}\sum_{\theta \in \mathcal{C}} \Big[P_\theta(y|x)P(\theta)\Big] \log \Big[P_\theta(y|x)P(\theta)\Big]$$

If Andy Warhol Were Bayesian…

2-dimensional
3-class
problem



4 example
decision trees
from 50 labeled
instances

Bayesian
prediction
from 10-tree
ensemble

vote entropy
among
committee
of 10 trees

Tangent: Active vs. Semi-Supervised

- both try to attack the same problem: making the most of unlabeled data $\mathcal{U}$

uncertainty sampling
query instances the model is least confident about

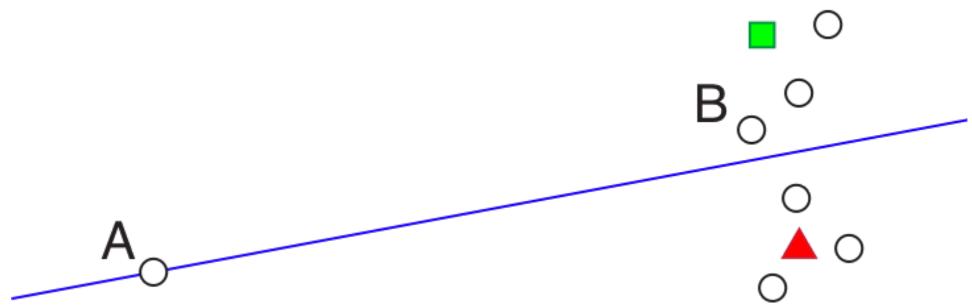
self-training
expectation-maximization (EM)
entropy regularization (ER)
propagate confident labelings among unlabeled data

query-by-committee (QBC)
use ensembles to rapidly reduce the version space

co-training
multi-view learning
use ensembles with multiple views to constrain the version space w.r.t. unlabeled data

Problem: Outliers

- an instance may be uncertain or controversial (for QBC) simply because it's an *outlier*



- querying outliers is not likely to help us reduce error on more typical data

Solution 1: Density Weighting

- weight the uncertainty ("informativeness") of an instance by its density w.r.t. the pool $\mathcal{U}$
  [Settles & Craven, EMNLP'08]

$$\phi_{ID}(x) = \underbrace{\boxed{H_\theta(Y|x)}}_{\text{"base" informativeness}} \times \underbrace{\boxed{\left(\frac{1}{U} \sum_{u \in \mathcal{U}} \mathrm{sim}(x, u) \right)^{\beta}}}_{\text{density term}}$$

- use $\mathcal{U}$ to approximate $P(x)$ and avoid outliers

[McCallum & Nigam, ICML'98; Nguyen & Smeulders, ICML'04; Xu et al., ECIR'07]

[Roy & McCallum, ICML'01; Zhu et al., ICML-WS'03]

Solution 2: Estimated Error Reduction

- minimize the risk $R(x)$ of a query candidate
  - expected uncertainty over $\mathcal{U}$ if x is added to $\mathcal{L}$

expectation over possible labelings of x

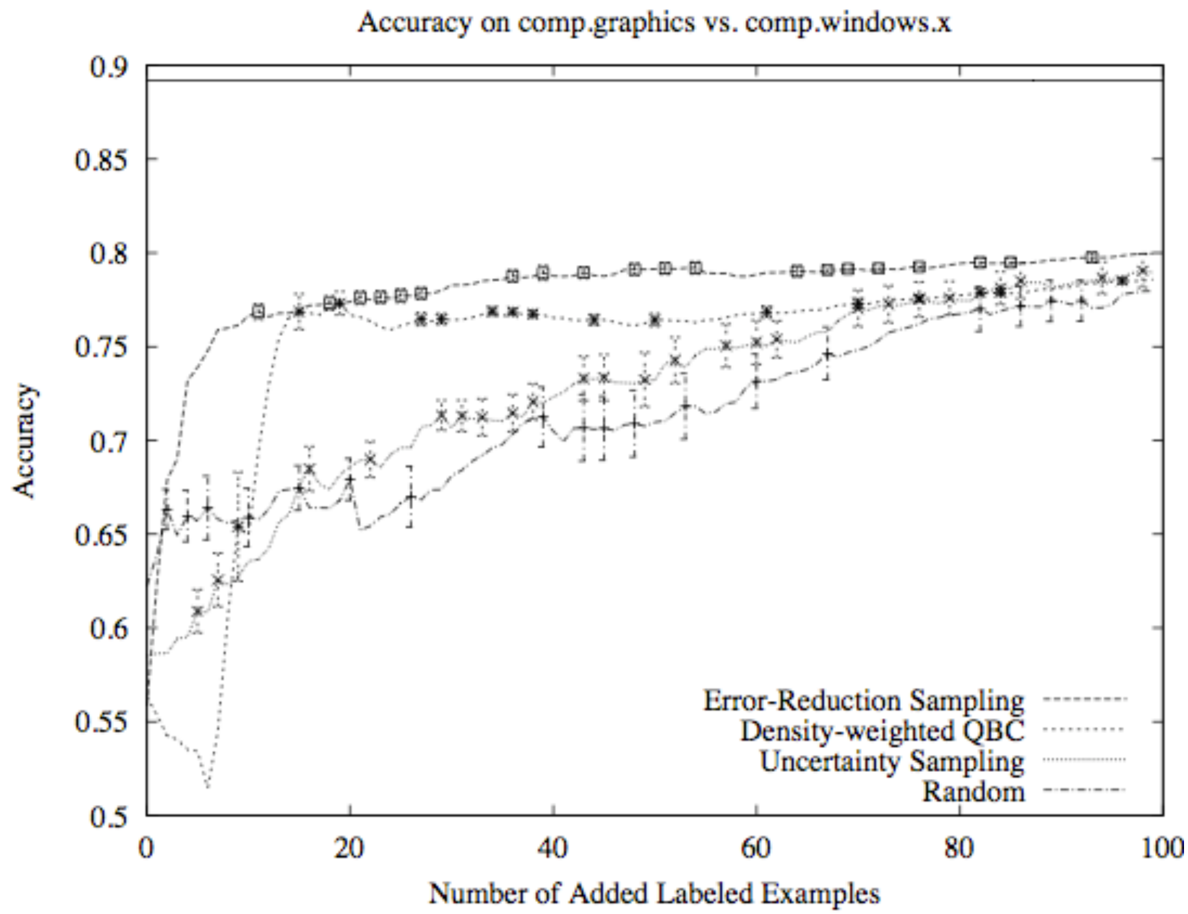
$$R(x) = \sum_{u \in \mathcal{U}} E_y \left[H_{\theta + \langle x, y \rangle} (Y | u) \right]$$

sum over unlabeled instances

uncertainty of u after retraining with x

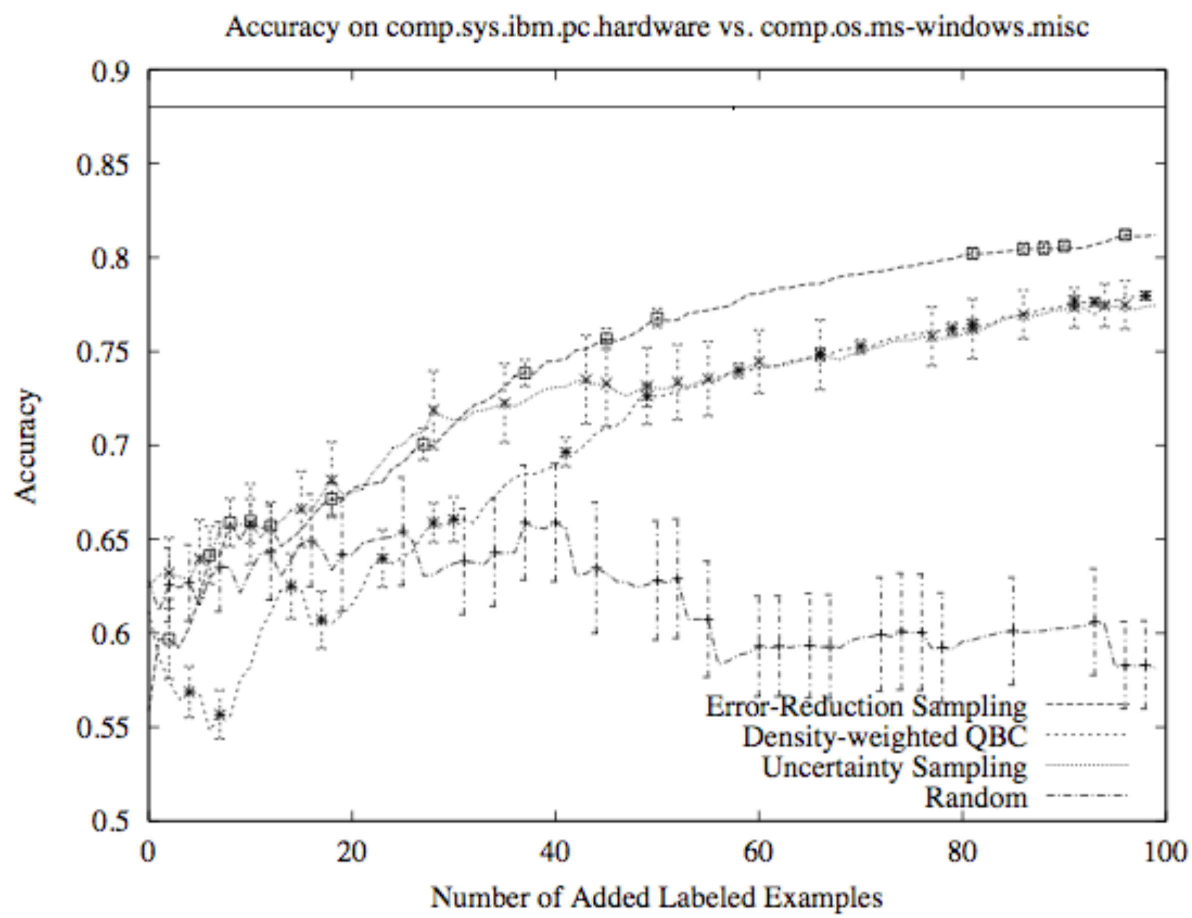
[Roy & McCallum, ICML'01]

Text Classification Examples



[Roy & McCallum, ICML'01]

Text Classification Examples



Relationship to Uncertainty Sampling

- a different perspective: aim to maximize the *information gain* over $\mathcal{U}$

uncertainty *before* query

risk term

$$\phi_{IG}(x) = \sum_{u \in \mathcal{U}} H_\theta(Y|u) - E_y\left[H_{\theta+\langle x,y\rangle}(Y|u)\right]$$

assume x is representative of $\mathcal{U}$

$$\approx H_\theta(Y|x) - E_y\left[H_{\theta+\langle x,y\rangle}(Y|x)\right]$$

$$\approx H_\theta(Y|x)$$

assume this evaluates to zero

…reduces to uncertainty sampling!

“Error Reduction” Scoresheet

- pros:
  - more principled query strategy
  - can be model-agnostic
    - literature examples: naïve Bayes, LR, GP, SVM
- cons:
  - too expensive for most model classes
    - some solutions: subsample $\mathcal{U}$; use approximate training
  - intractable for multi-class and structured outputs

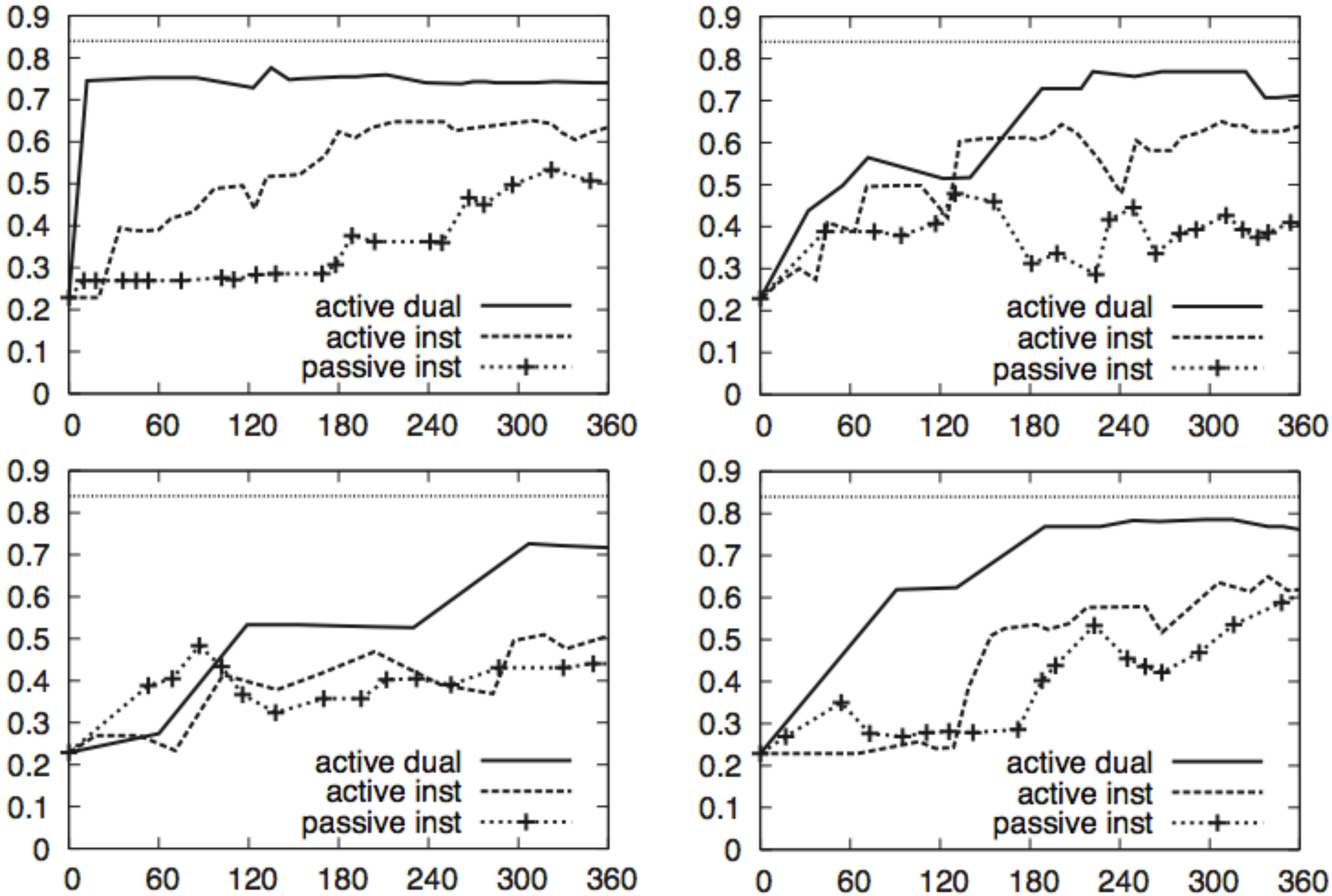
Alternative Query Types

- for some tasks, we can often intuitively label *features*
  - the feature word "*puck*" indicates the label **hockey**
  - the feature word "*strike*" indicates the label **baseball**
- **dual supervision** exploits this domain knowledge using both instance- and feature labels
  [Settles, 2011; Attenberg et al., 2010; Druck et al., 2009]
  - e.g., "does *puck* indicate the class **hockey**?"
- does it help to *actively* solicit domain knowledge?

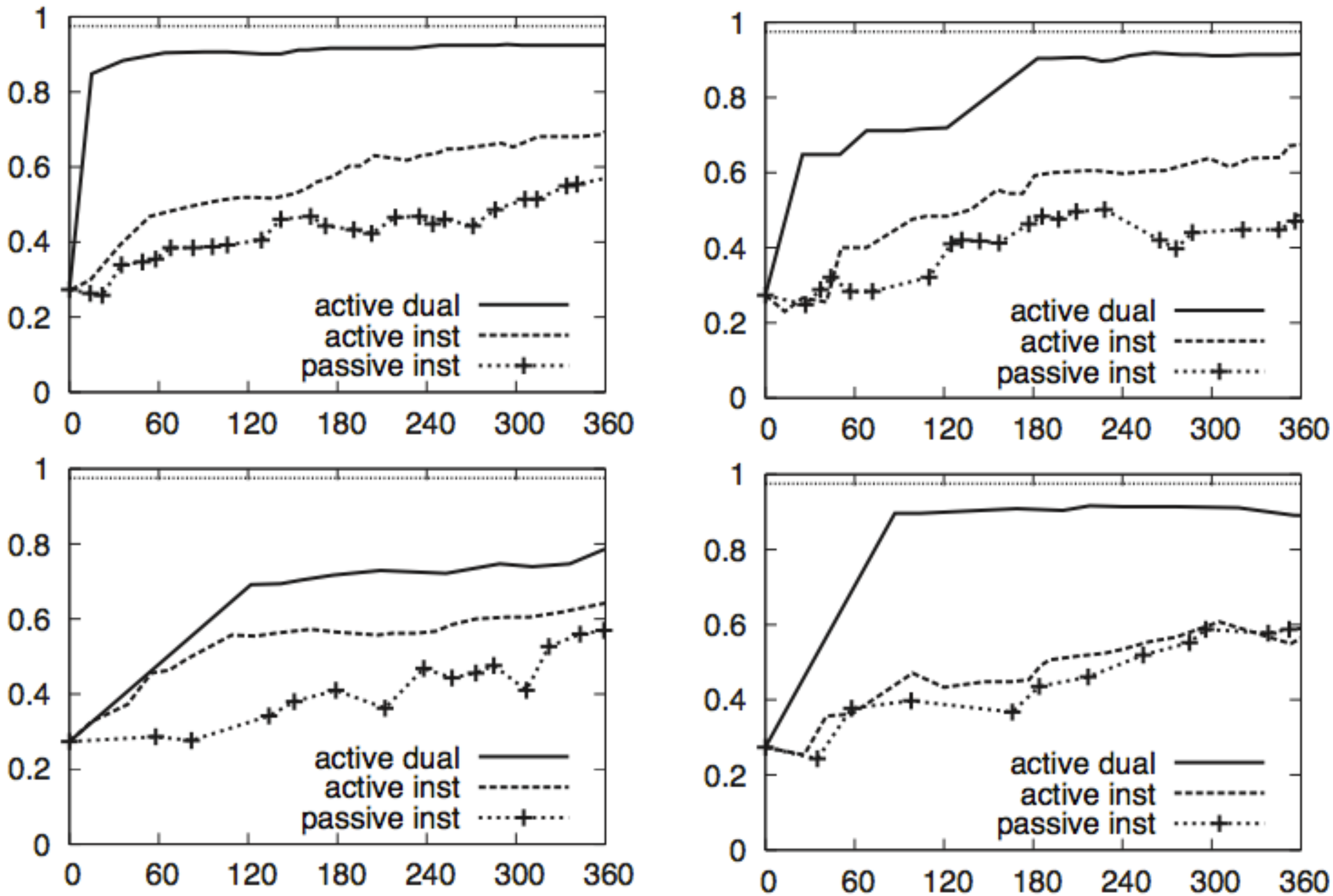
DUALIST

- open-source software project for interactive text annotation which combines:
  - semi-supervised learning
    - naïve Bayes + EM
  - domain knowledge
    - i.e., priors on P(word|y) parameters
  - active learning
    - instance queries using uncertainty sampling
    - feature queries using mutual information

Results: University Web Pages



Results: Science Newsgroups



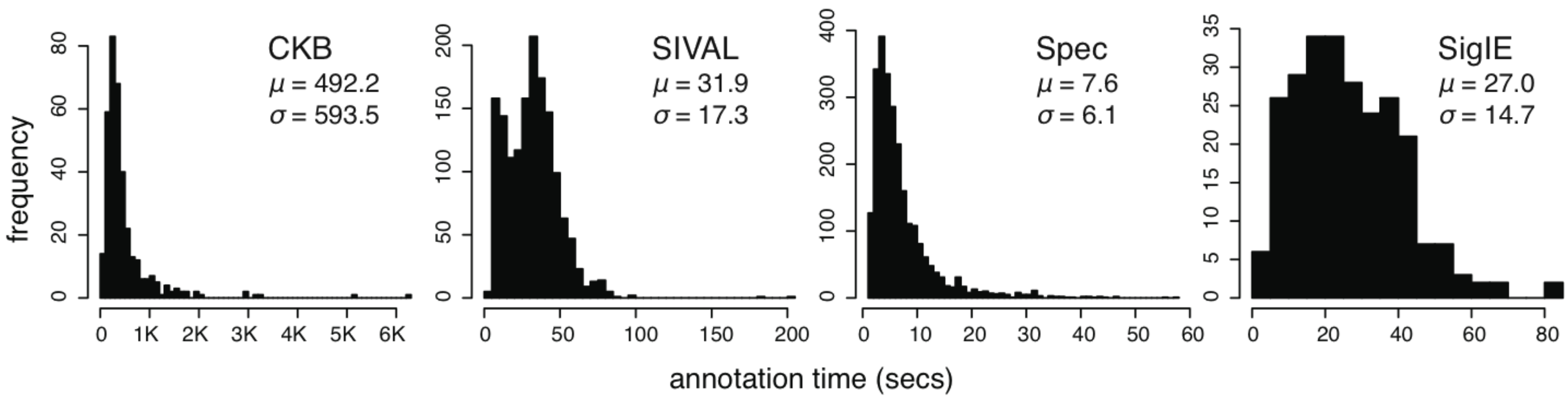
Real-World Annotation Costs

- so far, we've assumed that queries are equally expensive to label
  - for many tasks, labeling "costs" vary

[Settles et. al, NIPS'08]

Example: Annotation Time As Cost

- do annotation times vary among instances?

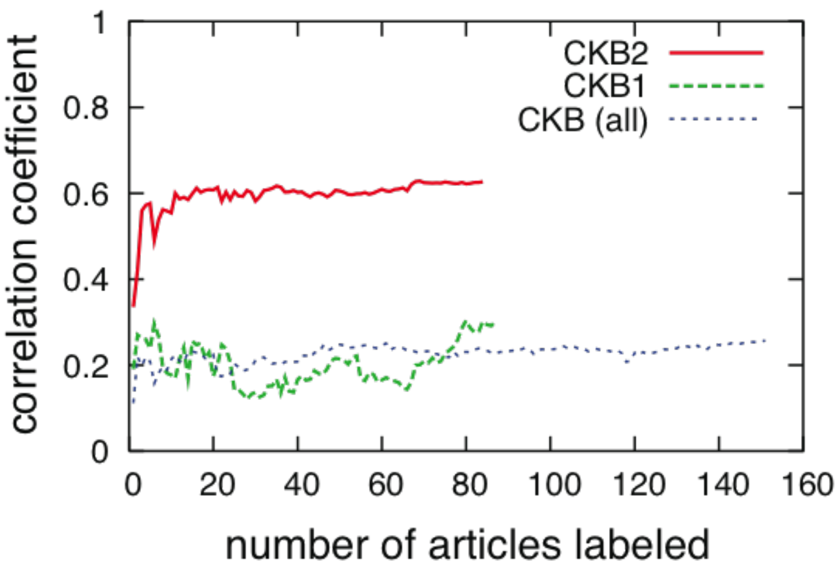


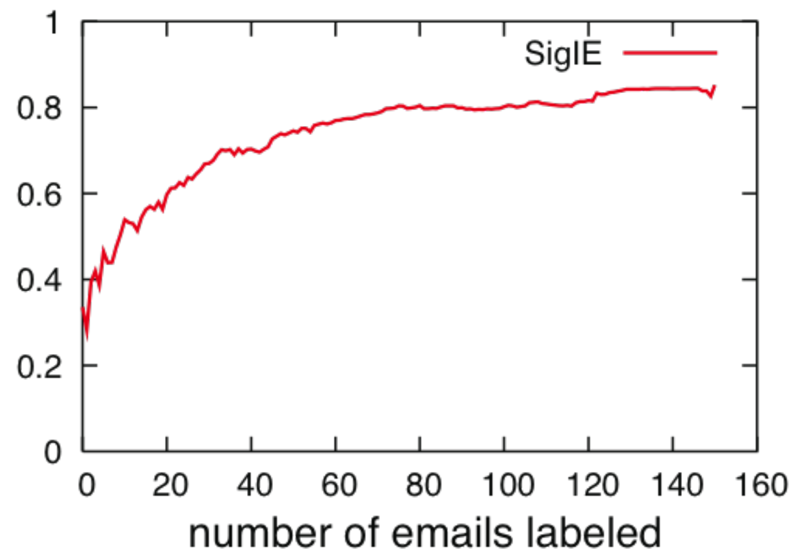
- where does this variance come from?
  - sometimes annotator-dependent
  - stochastic effects

[Settles et. al, NIPS'08]

Can Labeling Times be Predicted?

cost predictor: regression model using meta-features





Interesting Open Issues

- better cost-sensitive approaches
- “crowdsourced” labels (noisy oracles)
- batch active learning (many queries at once)
- multi-task active learning
- HCI / user interface issues
- data reusability