Active Learning

John Langford @ Microsoft Research

Machine Learning the Future Class, May 1

An instrument of mass machine learning

How can we formalize it's use?

Exploiting unlabeled data

A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

But labeling can be expensive.

Exploiting unlabeled data

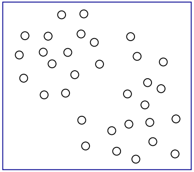
A lot of unlabeled data is plentiful and cheap, eg.

documents off the web

speech samples

images and video

But labeling can be expensive.



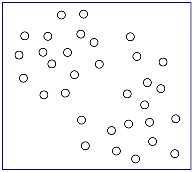
Unlabeled points

Exploiting unlabeled data

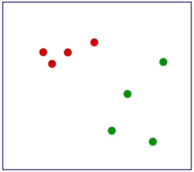
A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

But labeling can be expensive.



Unlabeled points



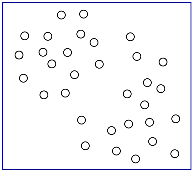
Supervised learning

Exploiting unlabeled data

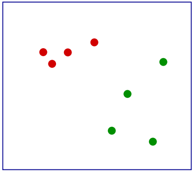
A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

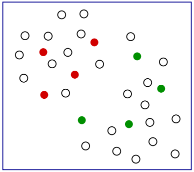
But labeling can be expensive.



Unlabeled points



Supervised learning



Semisupervised and active learning

Active Learning

Can interaction help us learn effectively?

The Active Learning Setting

Repeatedly:

1. Observe unlabeled example x.
2. Asking for label? Yes/no
3. If yes, observe label y.

Goal: Simultaneously optimize quality of learned classifier and minimize the number of labels requested.

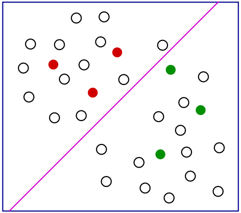
Typical heuristics for active learning

Start with a pool of unlabeled data

Pick a few points at random and get their labels

Repeat

- Fit a classifier to the labels seen so far
- Query the unlabeled point that is closest to the boundary (or most uncertain, or most likely to decrease overall uncertainty,...)



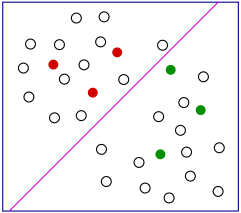
Typical heuristics for active learning

Start with a pool of unlabeled data

Pick a few points at random and get their labels

Repeat

Fit a classifier to the labels seen so far
Query the unlabeled point that is closest to the boundary (or most uncertain, or most likely to decrease overall uncertainty,...)



Biased sampling: the labeled points are not representative of the underlying distribution!

Sampling bias

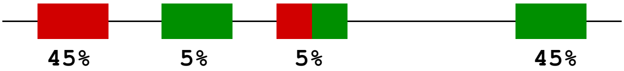
Start with a pool of unlabeled data

Pick a few points at random and get their labels

Repeat

- Fit a classifier to the labels seen so far
- Query the unlabeled point that is closest to the boundary (or most uncertain, or most likely to decrease overall uncertainty,...)

Example:



Sampling bias

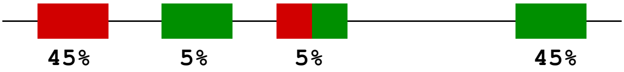
Start with a pool of unlabeled data

Pick a few points at random and get their labels

Repeat

Fit a classifier to the labels seen so far
Query the unlabeled point that is closest to the boundary
(or most uncertain, or most likely to decrease overall uncertainty,...)

Example:



Even with infinitely many labels, converges to a classifier with 5% error instead of the best achievable, 2.5%. *Not consistent!*

This problem occurs in practice.

Outline

1. Importance weighting
2. Rare Classes
3. Cost Sensitive

Importance Weighted Active Learning via Reduction

$S = \emptyset$

While (unlabeled examples remain)

1. Receive unlabeled example x.
2. Choose a probability of labeling p.
3. With probability p get label y, and add $(x, y, \frac{1}{p})$ to S.
4. Let $h = \text{Learn}(S)$.

Consistency Theorem: For all methods choosing $p > 0$, the algorithm is consistent.

How should p be chosen?

On the tth unlabeled point

let: $\hat{e}(h,S) = \frac{1}{t}\sum_{(x,y,i)\in S} i\mathbb{1}(h(x) \neq y)$ = importance weighted error rate.

How should p be chosen?

On the tth unlabeled point

let: $\hat{e}(h, S) = \frac{1}{t} \sum_{(x,y,i) \in S} i\mathbb{1}(h(x) \neq y)$ = importance weighted error rate.

Let h' = minimum error rate hypothesis choosing other label.

How should p be chosen?

On the tth unlabeled point

let: $\hat{e}(h, S) = \frac{1}{t} \sum_{(x,y,i) \in S} i \mathbb{1}(h(x) \neq y)$ = importance weighted error rate.
Let h' = minimum error rate hypothesis choosing other label.
Let $\Delta = \hat{e}(h', S) - \hat{e}(h, S)$ = error rate difference.

How should p be chosen?

On the tth unlabeled point

let: $\hat{e}(h,S) = \frac{1}{t}\sum_{(x,y,i)\in S} i\mathbb{1}(h(x)\neq y)$ = importance weighted error rate.

Let h' = minimum error rate hypothesis choosing other label.

Let $\Delta = \hat{e}(h',S) - \hat{e}(h,S)$ = error rate difference.

Choose $p=1$ if $\Delta \leq O\left(\sqrt{\frac{\log t}{t}}\right)$

Otherwise, let $p = O\left(\frac{\log t}{\Delta^2 t}\right)$

How should p be chosen?

On the tth unlabeled point

let: $\hat{e}(h,S) = \frac{1}{t}\sum_{(x,y,i)\in S} i\mathbb{1}(h(x) \neq y)$ = importance weighted error rate.
Let h' = minimum error rate hypothesis choosing other label.
Let $\Delta = \hat{e}(h',S) - \hat{e}(h,S)$ = error rate difference.
Choose $p = 1$ if $\Delta \leq O\left(\sqrt{\frac{\log t}{t}}\right)$
Otherwise, let $p = O\left(\frac{\log t}{\Delta^2 t}\right)$

Accuracy Theorem: With high probability, the IWAL reduction has a similar error rate to supervised learning on t points.

How should p be chosen?

On the tth unlabeled point

let: $\hat{e}(h,S) = \frac{1}{t}\sum_{(x,y,i)\in S} i\mathbb{1}(h(x)\neq y)$ = importance weighted error rate.

Let h' = minimum error rate hypothesis choosing other label.

Let $\Delta = \hat{e}(h',S) - \hat{e}(h,S)$ = error rate difference.

Choose $p=1$ if $\Delta \leq O\left(\sqrt{\frac{\log t}{t}}\right)$

Otherwise, let $p = O\left(\frac{\log t}{\Delta^2 t}\right)$

Accuracy Theorem: With high probability, the IWAL reduction has a similar error rate to supervised learning on t points.

Efficiency Theorem: If there is a small disagreement coefficient θ, the algorithm requires only $O\left(\theta\sqrt{t\log t}\right)$ + a minimum due to noise.

Disagreement Coefficient

Characterizes known examples where active learning can help.
Defined for any set of classifiers H and distribution D.

Disagreement Coefficient

Characterizes known examples where active learning can help.
Defined for any set of classifiers H and distribution D.

For any ϵ features x are of interest if there exists a hypothesis h:

1. With error rate less than ϵ larger than the best h^*.
2. That disagress with the best hypothesis, $h^*(x) \neq h(x)$.

Disagreement Coefficient

Characterizes known examples where active learning can help.
Defined for any set of classifiers H and distribution D.

For any ϵ features x are of interest if there exists a hypothesis h:

1. With error rate less than ϵ larger than the best h^*.
2. That disagress with the best hypothesis, $h^*(x) \neq h(x)$.

Disagreement coefficient is $\theta = \max_\epsilon \frac{\Pr(\text{interesting}_\epsilon\ x)}{\epsilon}$

Disagreement coefficient: examples

Disagreement coefficient: examples

- Thresholds in $\mathbb{R}$, any data distribution.

$$\theta = 2.$$

Disagreement coefficient: examples

- Thresholds in $\mathbb{R}$, any data distribution.

$$\theta = 2.$$

- Linear separators through the origin in $\mathbb{R}^d$, uniform data distribution.

$$\theta \leq \sqrt{d}.$$

Disagreement coefficient: examples

- Thresholds in $\mathbb{R}$, any data distribution.

$$\theta = 2.$$

- Linear separators through the origin in $\mathbb{R}^d$, uniform data distribution.

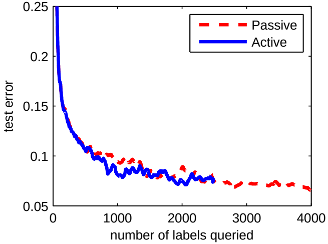
$$\theta \leq \sqrt{d}.$$

- Linear separators in $\mathbb{R}^d$, smooth data density bounded away from zero.

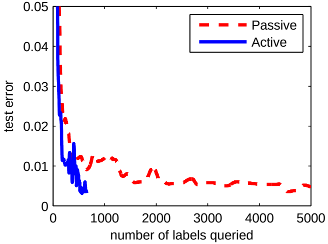
$$\theta \leq c(h^*)d$$

where $c(h^*)$ is a constant depending on the target h^*.

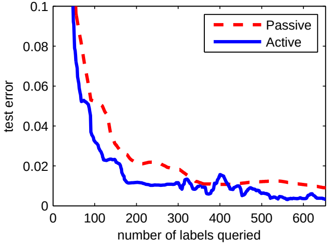
Decision Tree Experiments



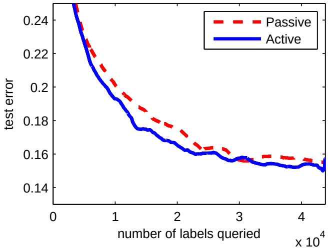
MNIST 3s vs 5s



KDDCUP99



KDDCUP99 (close-up)



MNIST multi-class (close-up)

An Approximate IWAL

Let $h(x) = \texttt{Learn}(S)$.
Let $h'(x) = \texttt{Learn}_{h(x) \neq y}(S)$.
Claim: If $\texttt{Learn}$ minimizes error rates, for all $\epsilon > 0$

$$\texttt{Learn}(S \cup (x, -h(x), t\Delta + \epsilon)) = h'(x)$$

In other words $t\Delta =$ importance weight required to change label for current x.

An Approximate IWAL

Let $h(x) = \texttt{Learn}(S)$.
Let $h'(x) = \texttt{Learn}_{h(x) \neq y}(S)$.
Claim: If $\texttt{Learn}$ minimizes error rates, for all $\epsilon > 0$

$$\texttt{Learn}(S \cup (x, -h(x), t\Delta + \epsilon)) = h'(x)$$

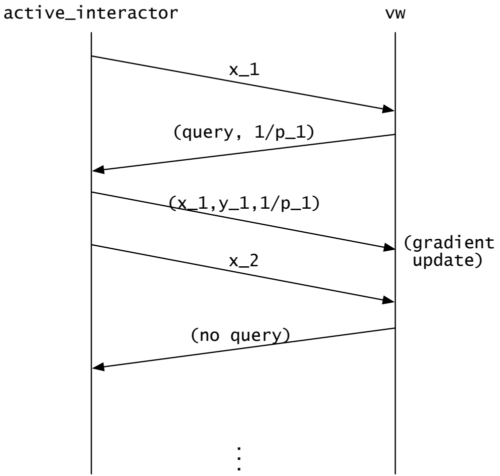
In other words $t\Delta =$ importance weight required to change label for current x.

Using Vowpal Wabbit as base learner, estimate $t \cdot \Delta$ as the number of gradient updates with x required for prediction to switch (from 0 to 1, or from 1 to 0).

e.g., for importance weight-aware square-loss update:

$$\Delta_t := \frac{1}{t \cdot \eta_t} \cdot \log \frac{\max\{h(x),\ 1 - h(x)\}}{0.5}$$

Active learning in Vowpal Wabbit



`active_interactor.cc` (in git repository) demonstrates how to implement this protocol.

Fringe Benefits

This approach has many nice properties.

Fringe Benefits

This approach has many nice properties.

1. Always consistent.

Fringe Benefits

This approach has many nice properties.

1. Always consistent.
2. Efficient.
    1. Label Efficient.
    2. Unlabeled data efficient.
    3. Computationally efficient.

Fringe Benefits

This approach has many nice properties.

1. Always consistent.
2. Efficient.
    1. Label Efficient.
    2. Unlabeled data efficient.
    3. Computationally efficient.
3. Compatible.
    1. With Online Algorithms
    2. With any optimization-style classification algorithm.
    3. With any Loss function
    4. With supervised learning
    5. With switching learning algorithms (!)

Fringe Benefits

This approach has many nice properties.

1. Always consistent.
2. Efficient.
   1. Label Efficient.
   2. Unlabeled data efficient.
   3. Computationally efficient.
3. Compatible.
   1. With Online Algorithms
   2. With any optimization-style classification algorithm.
   3. With any Loss function
   4. With supervised learning
   5. With switching learning algorithms (!)
4. It works, empirically.

Outline

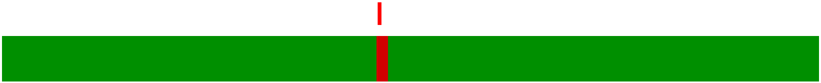
1. Importance weighting
2. Rare Classes
3. Cost Sensitive

How many labels for a rare class?

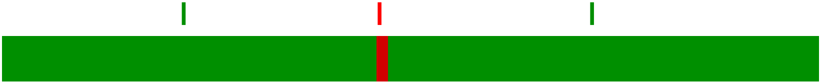
How many labels for a rare class?

Attenberg & Provost 2010: Search and insertion of labeled rare class examples helps.

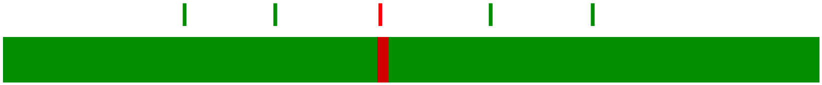
Why does search help?



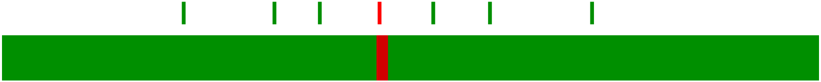
Why does search help?



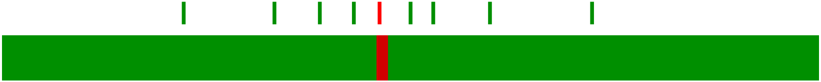
Why does search help?



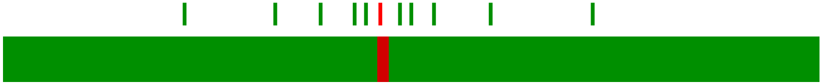
Why does search help?



Why does search help?



Why does search help?



Why does search help?

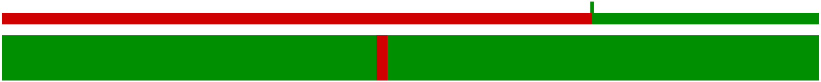
Potentially: exponential improvement in label complexity!

When does Search help?

Does searching for counterexamples help?

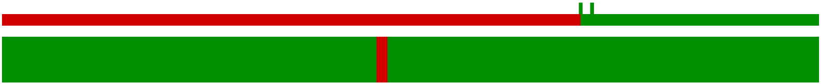
When does Search help?

Does searching for counterexamples help?



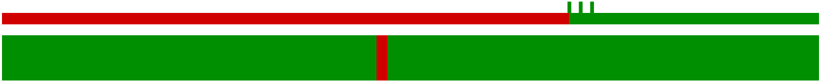
When does Search help?

Does searching for counterexamples help?



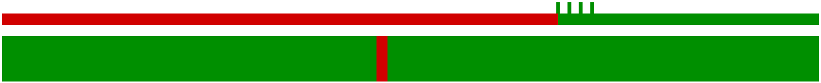
When does Search help?

Does searching for counterexamples help?



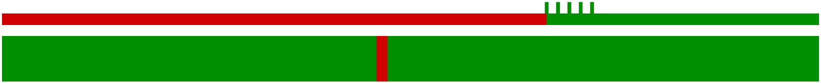
When does Search help?

Does searching for counterexamples help?



When does Search help?

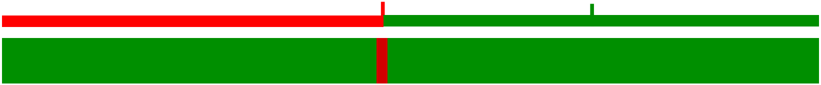
Does searching for counterexamples help?



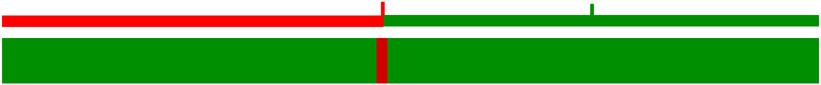
No!

Counterexample to version space instead!

Counterexample to version space instead!



Counterexample to version space instead!



Theorem: Search for version space counterexample can reduce Label calls exponentially by starting with simple set of classifiers and moving to more complex as they are proved inadequate.

Outline

1. Importance weighting
2. Rare Classes
3. Cost Sensitive

How to do Active Cost Sensitive Classification?

Cost-sensitive multi-class classification

Distribution D over $X \times [0,1]^k$, where a vector in $[0,1]^k$ specifies the cost of each of the k choices.

Find a classifier $h : X \to \{1, \ldots, k\}$ minimizing the expected cost

$$\text{cost}(h, D) = \mathbf{E}_{(x,c)\sim D}[c_{h(x)}].$$

How to do Active Cost Sensitive Classification?

Cost-sensitive multi-class classification

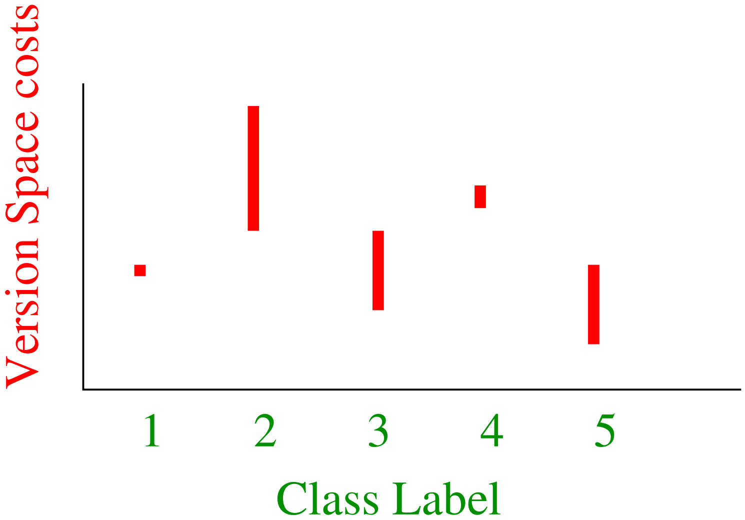
Distribution D over $X \times [0,1]^k$, where a vector in $[0,1]^k$ specifies the cost of each of the k choices.

Find a classifier $h : X \to \{1, \ldots, k\}$ minimizing the expected cost

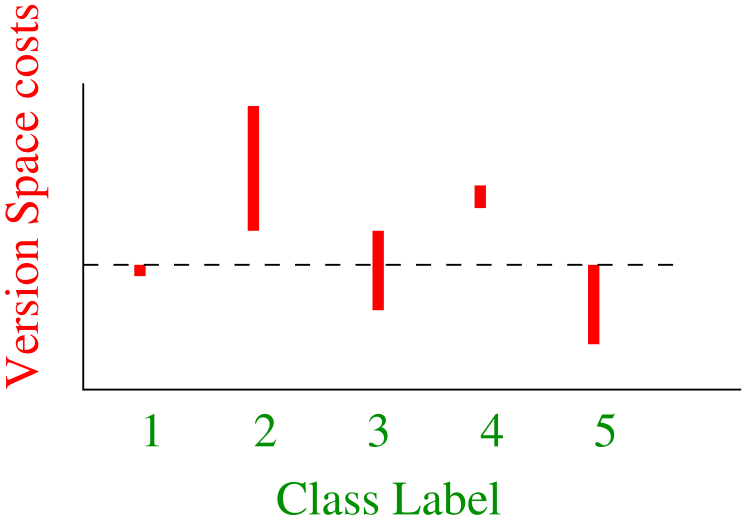
$$\text{cost}(h, D) = \mathbf{E}_{(x,c)\sim D}[c_{h(x)}].$$

Should queries be per-example or per-cost?

Which class costs should be queried?

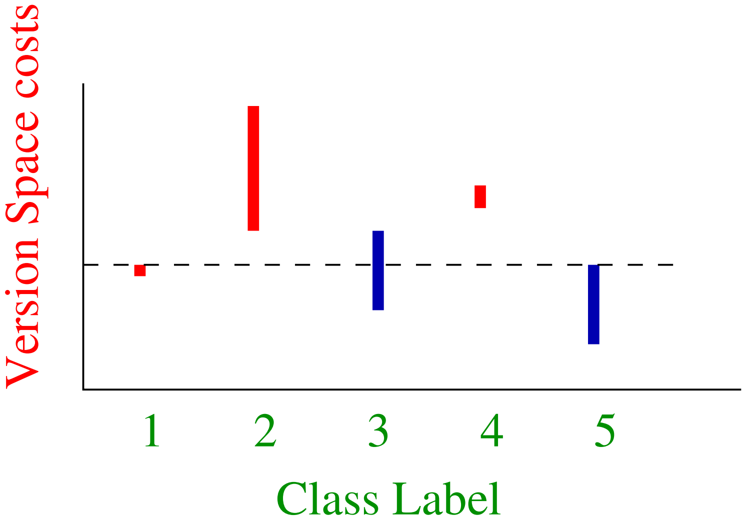


Which class costs should be queried?



1. Minimum cost < smallest maximum cost

Which class costs should be queried?



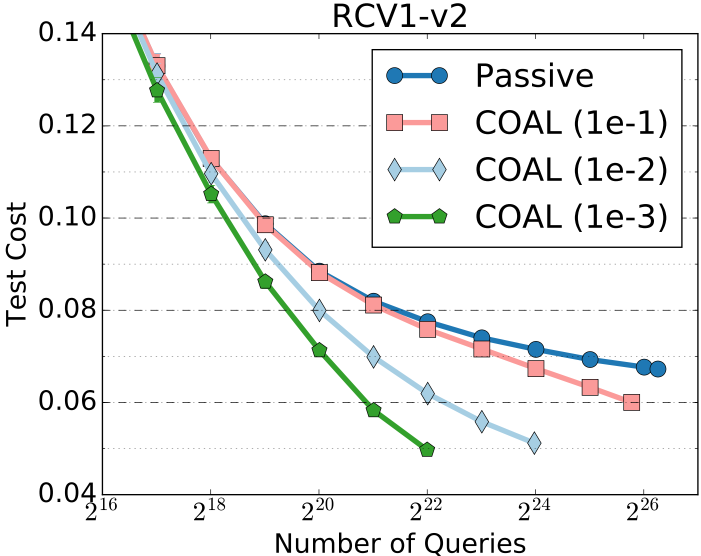
1. Minimum cost $<$ smallest maximum cost
2. Cost difference matters

Cost Overlapped Active Learning Results

Theorem: Works efficiently if (1) cost predictors factorize (2) squared loss optimizer is efficient (3) world is IID.

Cost Overlapped Active Learning Results

Theorem: Works efficiently if (1) cost predictors factorize (2) squared loss optimizer is efficient (3) world is IID.



Bibliography: Agnostic Active Learning history

Possibility N Balcan, A Beygelzimer, J Langford, Agnostic Active Learning. ICML 2006.

Noise M Kaariainen, Active Learning in the Non-realizable Case, ALT 2006.

Disagree S Hanneke. A Bound on the Label Complexity of Agnostic Active Learning. ICML 2007.

Online S Dasgupta, D Hsu, and C Monteleoni. A general agnostic active learning algorithm. NIPS 2007.

Weights F Bach. Active learning for misspecified generalized linear models. NIPS 2007.

Bibliography AAL algorithms

IWAL I A Beygelzimer, S Dasgupta, and J Langford, Importance Weighted Active Learning, ICML 2009.

IWAL II A Beygelzimer, D Hsu, J Langford, T Zhang, Agnostic Active Learning Without Constraints, NIPS 2010.

Search I J Attenberg and F Provost, Why Label when you can Search? Alternatives to Active Learning for Applying Human Resources to Build Classification Models Under Extreme Class Imbalance, KDD 2010.

Search II A Beygelzimer, D Hsu, J Langford, C Zhang, Search Improves Label for Active Learning, NIPS 2016.

COAL A Krishnamurthy, A Agarwal, TK Huang, H Daume, J Langford, Active Learning for Cost-Sensitive Classification, https://arxiv.org/abs/1703.01014