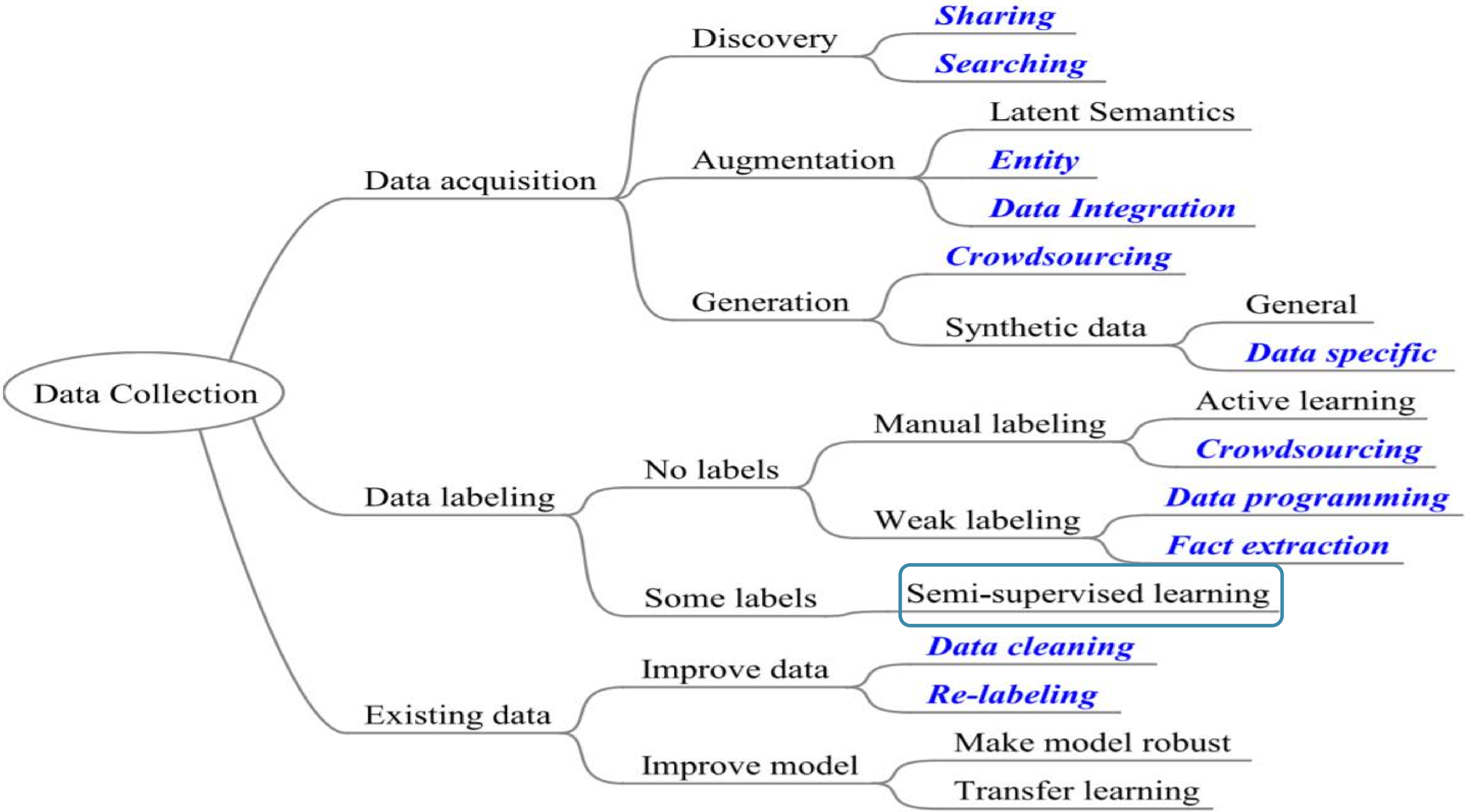
# DSE 203 (Fall 2020)

Human-In-the-Loop Data Integration (Active Learning)


Data Collection
Data acquisition
Discovery
Sharing
Searching
Augmentation
Latent Semantics
Entity
Data Integration
Generation
Crowdsourcing
Synthetic data
General
Data specific
Data labeling
No labels
Manual labeling
Active learning
Crowdsourcing
Weak labeling
Data programming
Fact extraction
Some labels
Semi-supervised learning
Existing data
Improve data
Data cleaning
Re-labeling
Improve model
Make model robust
Transfer learning

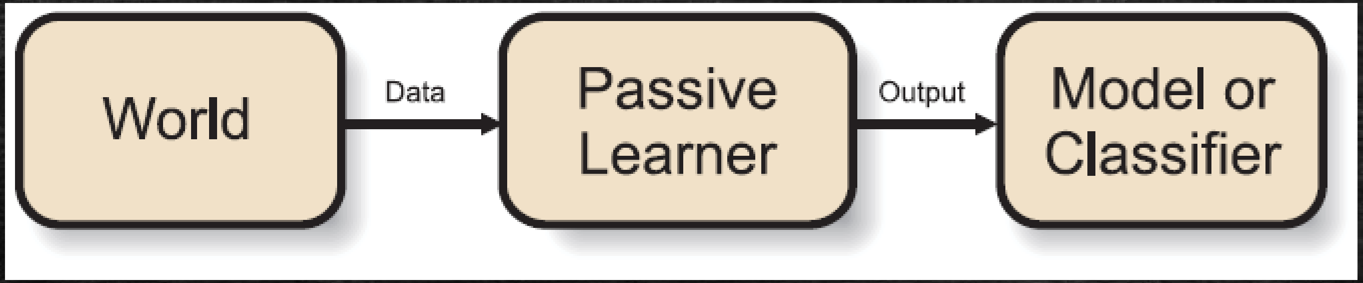
# Definition

***Active learning*** is a special case of semi-supervised machine learning in which a learning algorithm is able to ***interactively query the user*** (or some other information source) to obtain the desired outputs at new data points. In statistics literature it is sometimes also called optimal ***Experimental design***.

*Settles, Burr (2010)*
*Olsson, Fredrik (2009)*

# Passive Learning vs. Active Learning

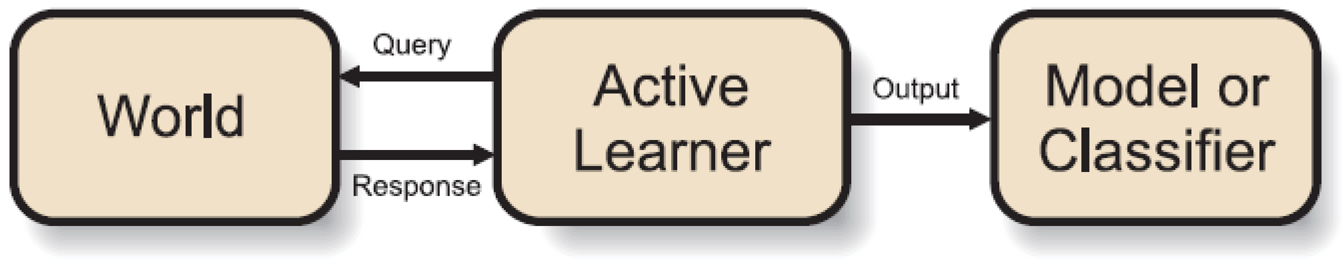
*General Scheme of a passive learner*



- For all supervised and unsupervised learning tasks, it needs to gather significant amount of data randomly sampled from the underlying population distribution and then induce a classifier or model
- Cons:
  - Gathering of labeled data often too expensive: time, money

# Passive Learning vs. Active Learning

*General Schema of an active learner*



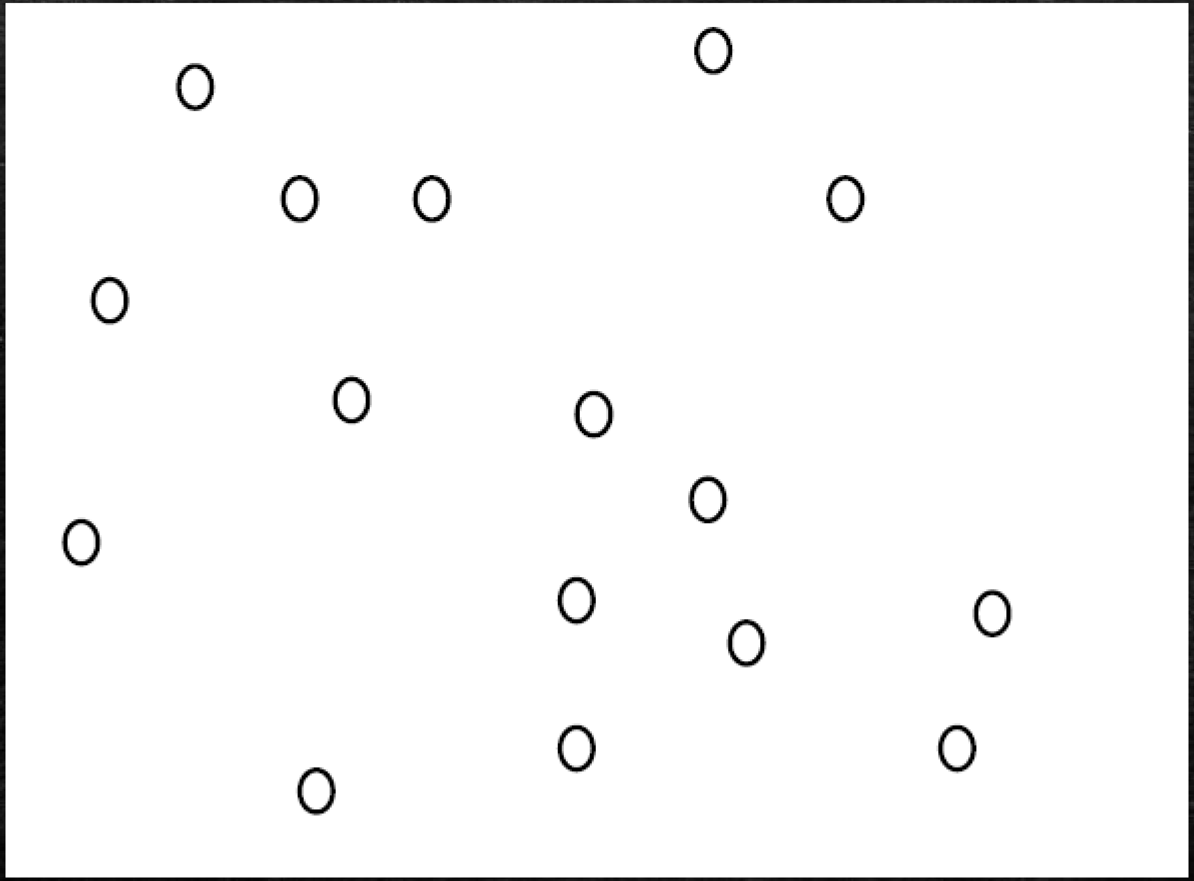
- Difference: the ability to ask queries about the world based upon the past queries and responses
- The notion of what exactly a query is and what response it receives will depend upon the exact task at hand
- Pro:
  - The entire data need not be labeled, it is the task of the learner to request for relevant label

# Active Learning Heuristic

- Start with a pool of unlabeled data (*U*)
- Pick a few points at random and get their labels using an oracle (e.g. human annotator)
- Repeat the following:
    1. Fit a classifier to the labels seen so far
    2. Pick the BEST unlabeled point to get a label for
        - (closest to the boundary?)
        - (most uncertain?)
        - (most likely to decrease overall uncertainty?)

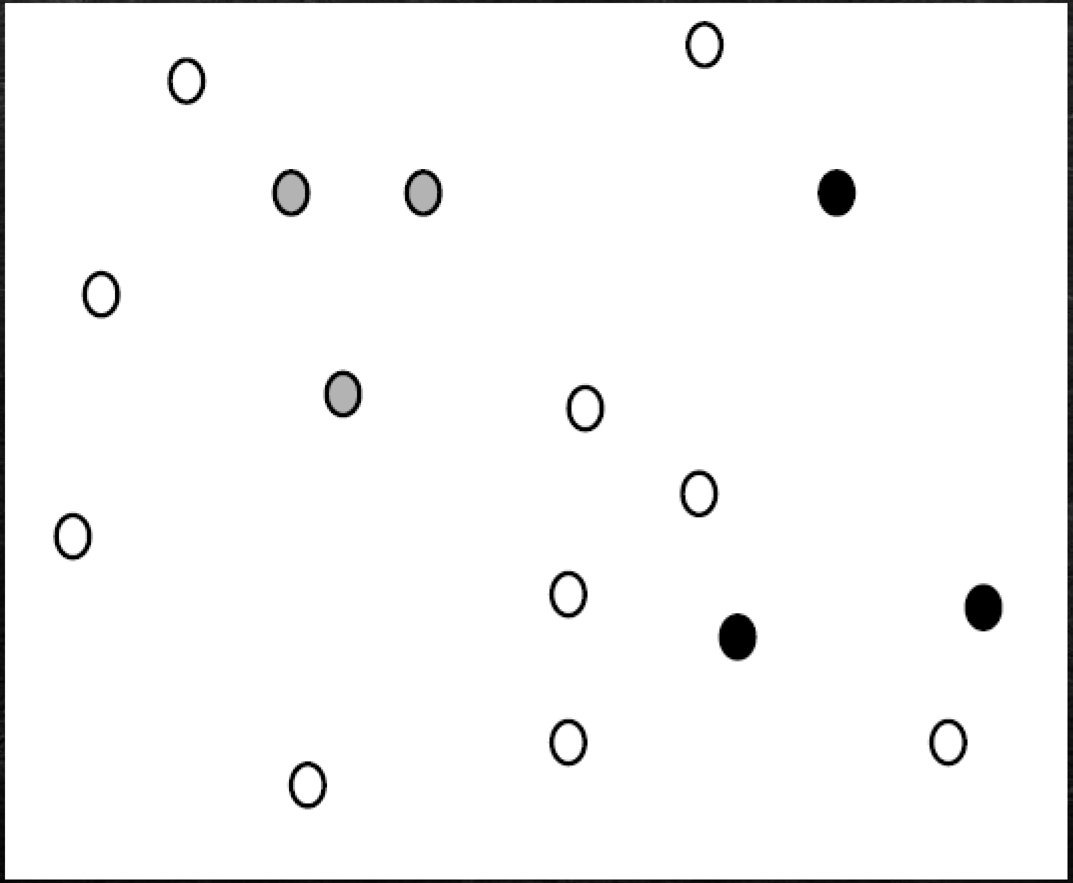
# Active Learning (II)

Start: Unlabeled data



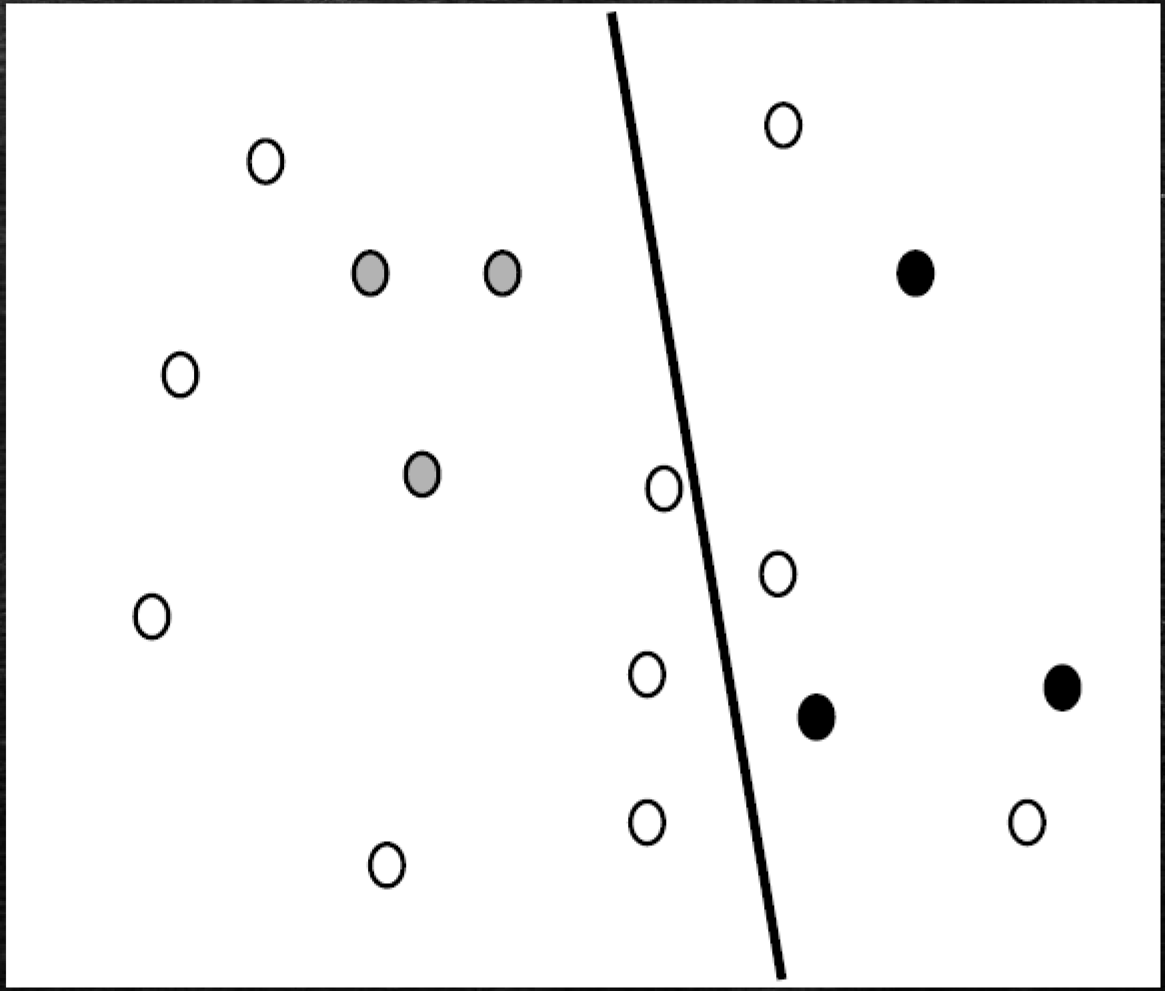
# Active Learning (III)

Label a random subset (Can we be smarter?)



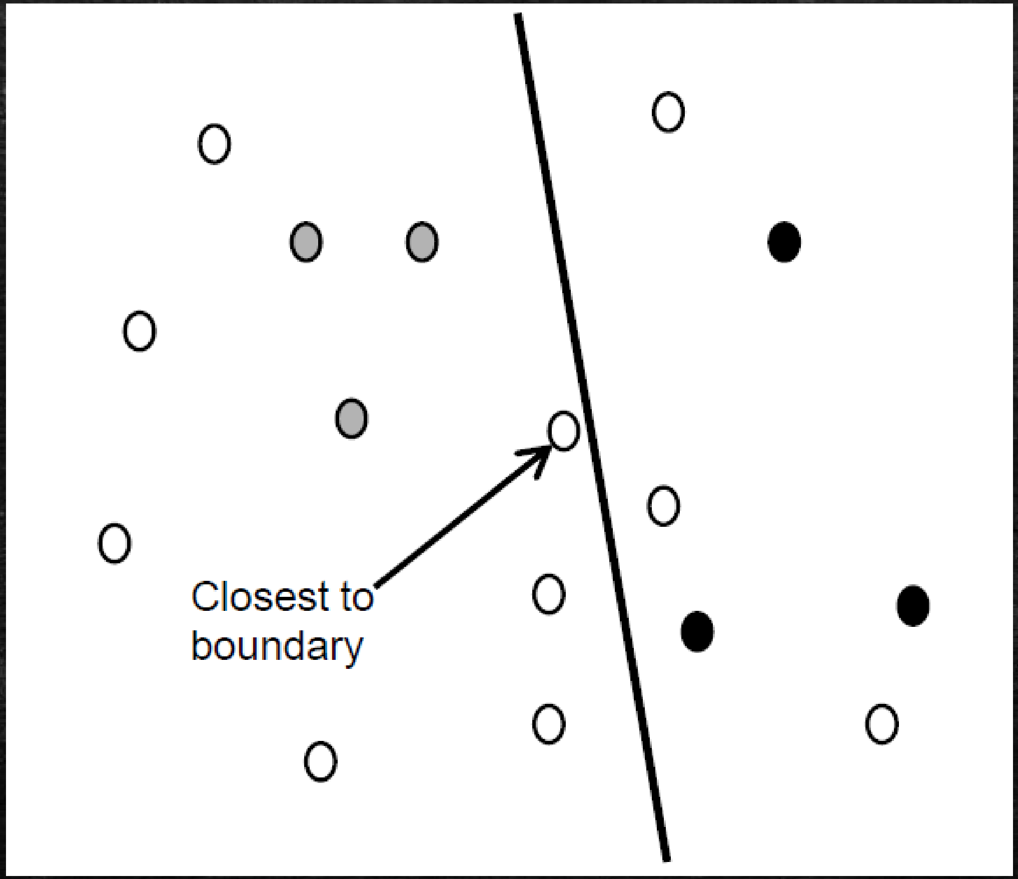
# Active Learning (IV)

Fit a classifier to labeled data



# Active Learning (V)

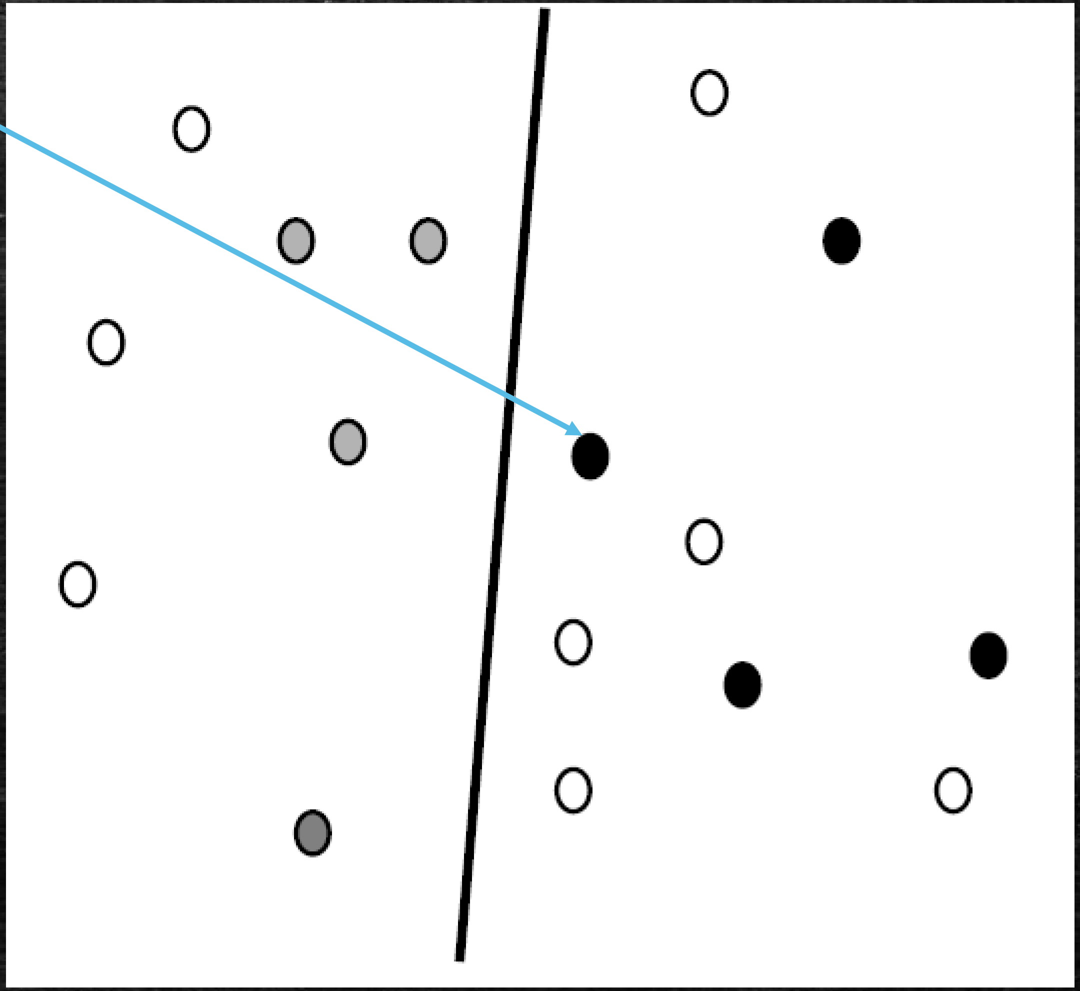
Pick the BEST next point to label (e.g. closest to boundary)



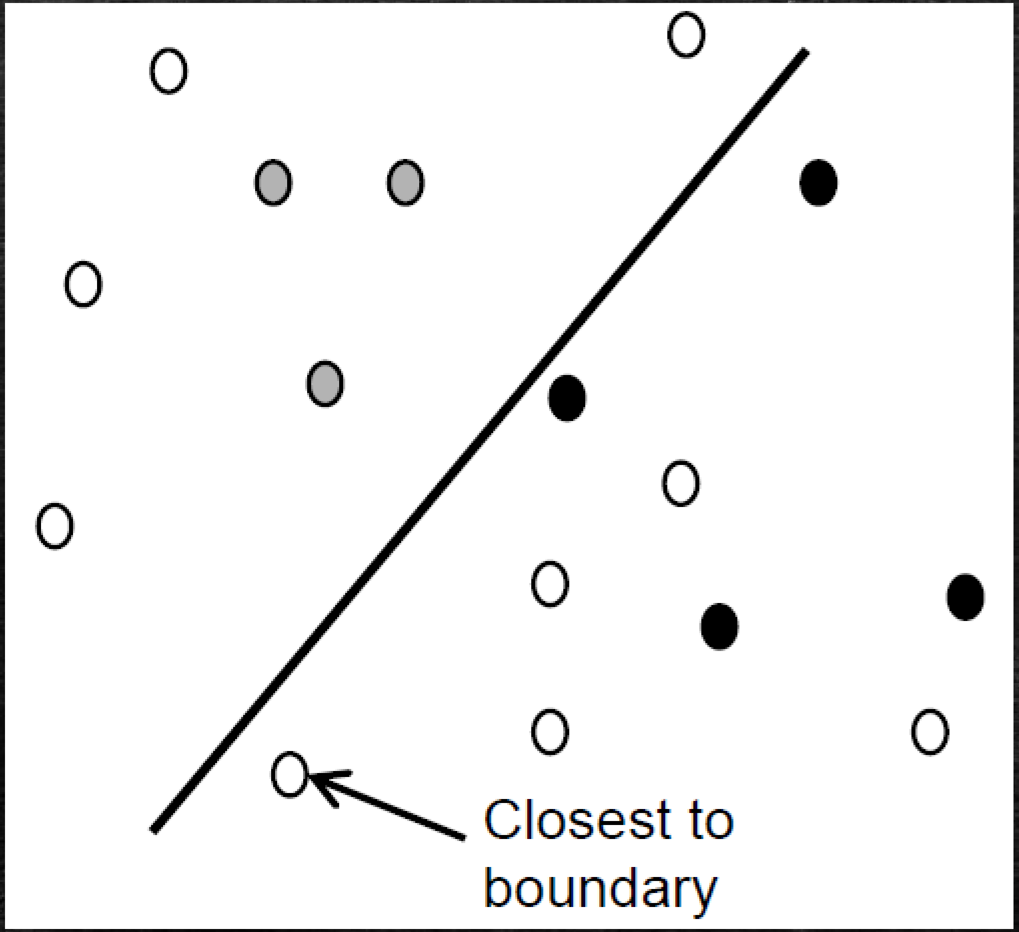
# Active Learning (VIII)

Human labels a data point

Fit a classifier to labeled data



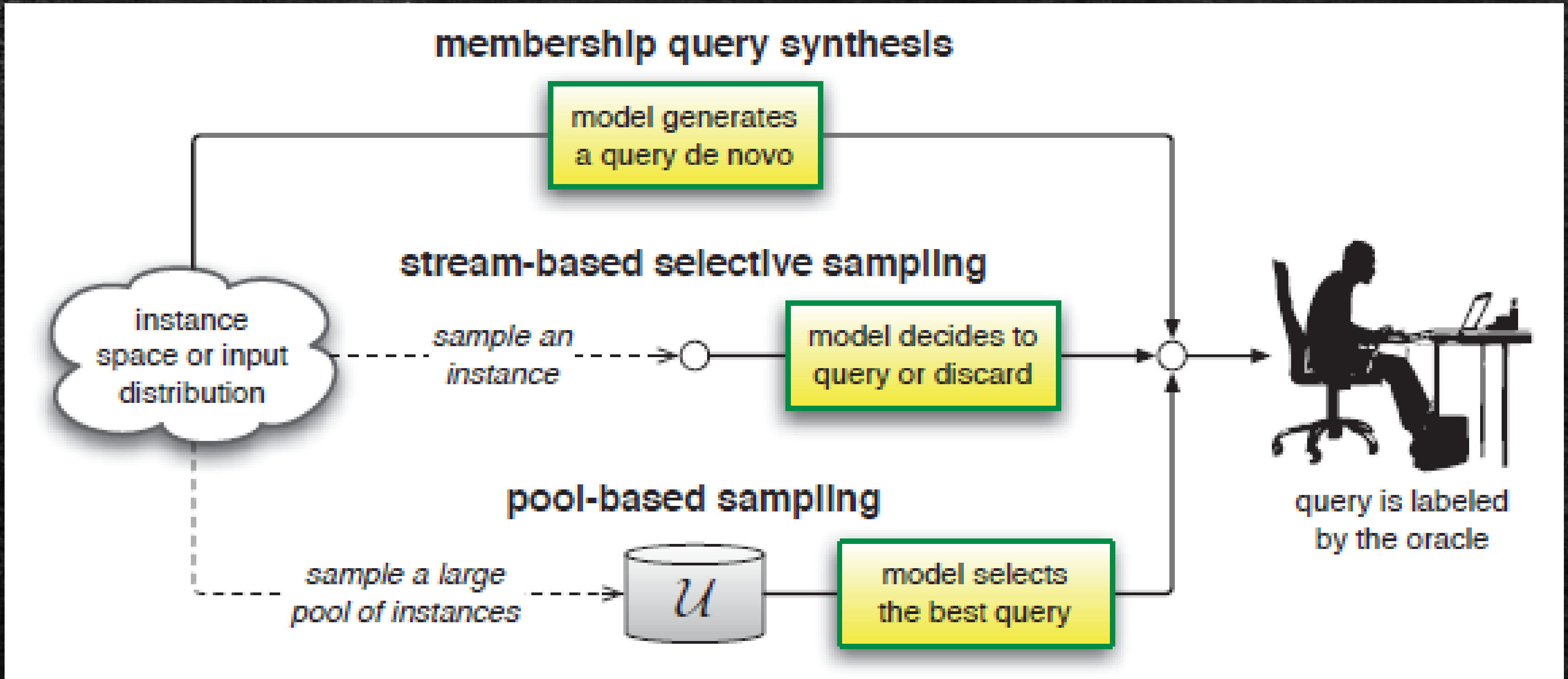
# Active Learning (VII)



Pick the BEST next point to label (e.g. closest to boundary)

# Scenarios (Approaches to Querying)

*Three Main Active Learning Scenarios (Settles, 2010)*


membership query synthesis
model generates a query de novo
stream-based selective sampling
instance space or input distribution
sample an instance
model decides to query or discard
pool-based sampling
sample a large pool of instances
$\mathcal{U}$
model selects the best query
query is labeled by the oracle

# Membership Query Synthesis

- The learner may request labels for *any unlabeled instance* in the input space, including queries that the learner generates de novo, rather than those sampled from some underlying natural distribution.
- Pros:
  - Computationally tractable (for finite domains)
  - Can be extended to regression tasks
  - Promising approach in automation of experiments that do not require a human annotator
- Cons:
  - Human annotator may have difficulty interpreting and labeling arbitrary instances

# Stream-Based Selective Sampling

- Generally used when obtaining an unlabeled instance is free or relatively cheap, the *query can first be sampled from the actual distribution*, and then the learner can decide whether or not to request its label.

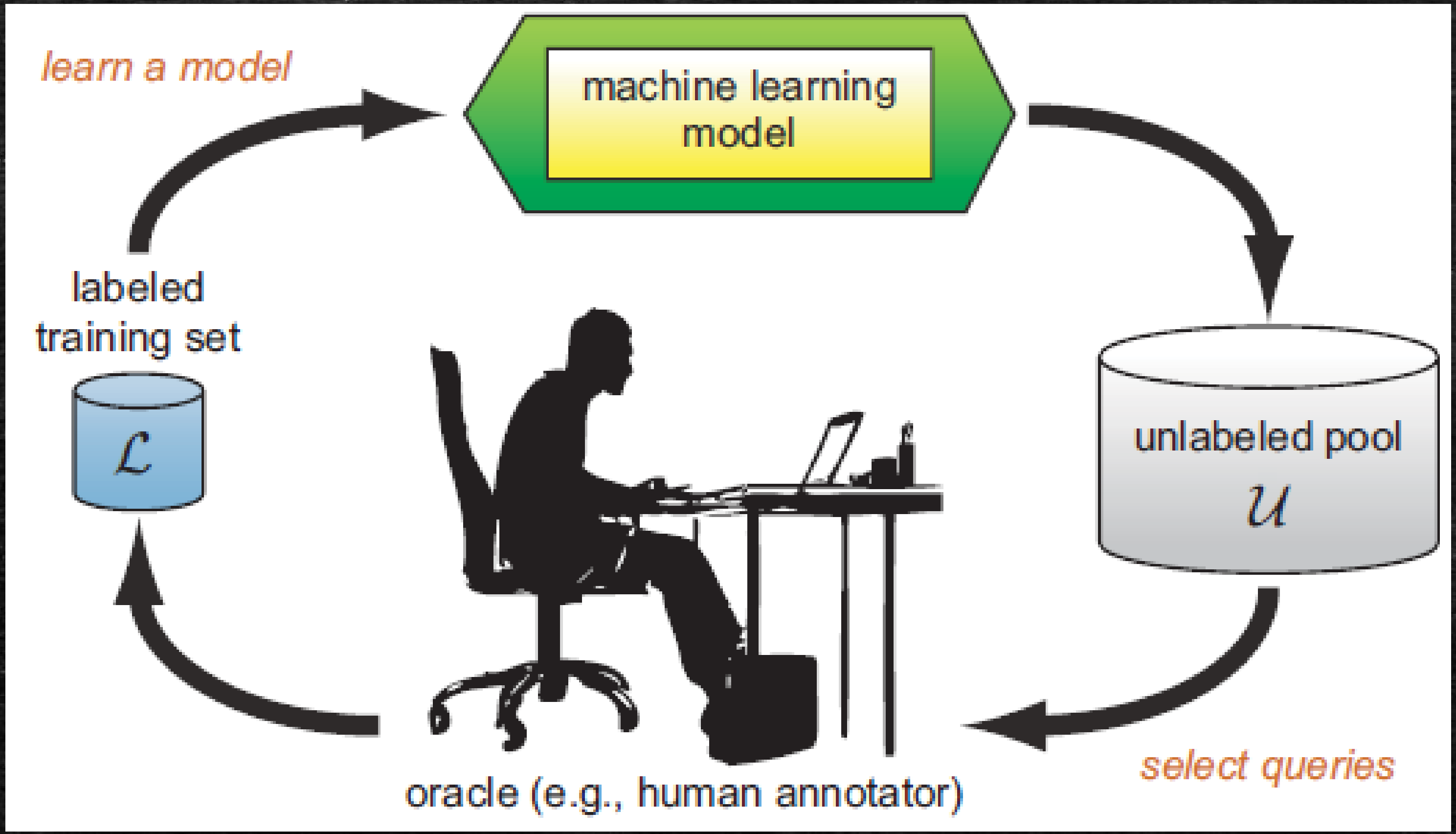
- This approach is called *stream / sequential* active learning, as each unlabeled instance is typically drawn one at a time from the data source, and the learner must decide whether to query or discard it.

- Pros:
  - Better to use when memory or processing power may be limited, as with mobile and embedded devices.

- Cons:
  - The assumption that the unlabeled data is available at low cost might not always hold.

# Pool-Based Sampling

*The pool-based active learning cycle (Settles, 2010)*



# Pool-Based Sampling

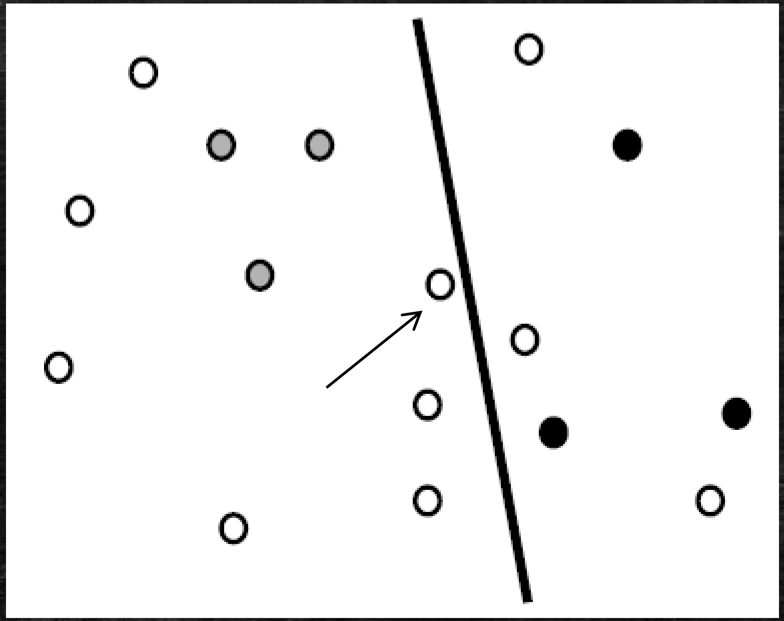
- It assumes that there is a small set of labeled data L and a large pool of unlabeled data U available
- The learner is supplied with a set of unlabeled examples from which it can selects queries.
- Pros:
  - Probably the most widely used sampling method (text classification, information extraction, image classification, video classification, speech recognition, cancer diagnosis)
  - Applied to many real-world tasks
- Cons:
  - Computationally intensive
  - Needs to evaluate entire set at each iteration

# Query Strategy Frameworks

1. Uncertainty Sampling
2. Query-By-Committee
3. Expected Model Change
4. Expected Error Reduction
5. Variance Reduction
6. Density-Weighted Methods

# Uncertainty Sampling (I)

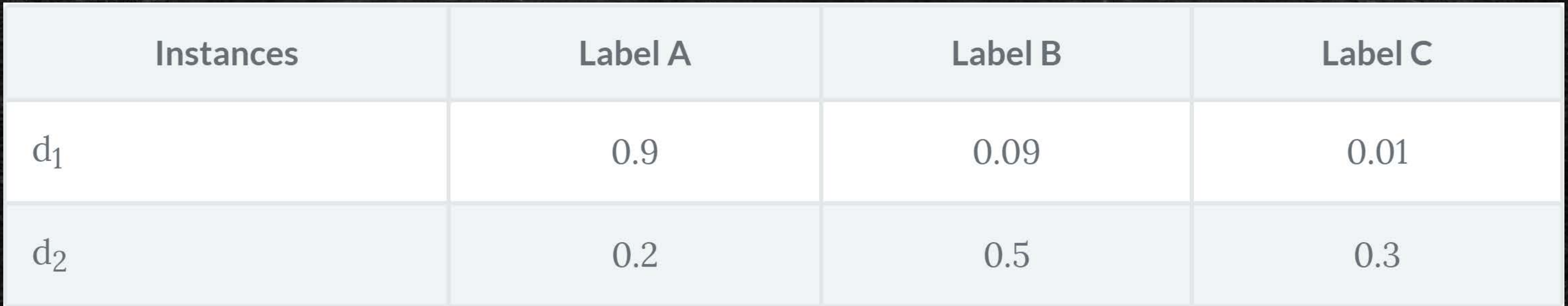
- An active learner queries the instances about which it is least certain how to label (e.g. closest to the decision boundary)



- Basically there are 3 strategies: least confident, margin sampling and entropy

# Example

- Two data points, three possible class labels



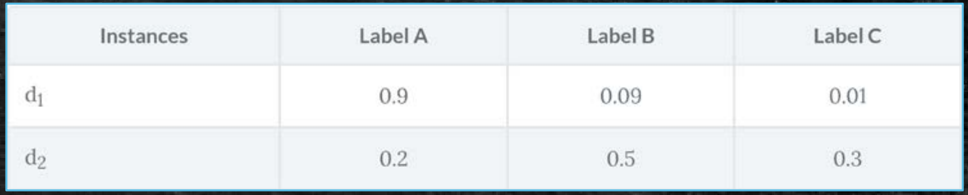
| Instances | Label A | Label B | Label C |
|---|---|---|---|
| $d_1$ | 0.9 | 0.09 | 0.01 |
| $d_2$ | 0.2 | 0.5 | 0.3 |

# Uncertainty Sampling (II)

- For problems with three or more class labels: it might query the instances whose prediction is the least confident
- The uncertainty measure:

$$x^*_{LC} = \underset{x}{\operatorname{argmax}}\ 1 - P_\theta(\hat{y}|x)$$

- Where $\hat{y} = \operatorname{argmax}_y P_\theta(y|x)$ is the class label with highest posterior probability under model θ
- Pros:
  - Appropriate if the objective function is to reduce classification error
- Cons:
  - This strategy only considers information about the most probable label



| Instances | Label A | Label B | Label C |
|---|---|---|---|
| $d_1$ | 0.9 | 0.09 | 0.01 |
| $d_2$ | 0.2 | 0.5 | 0.3 |

In the example, the learner is confident about the label for $d_1$, since it thinks it should be labelled A with probability 0.9, however, it is less sure about the label of $d_2$ since its probabilities are more spread and it thinks that it should be labelled B with a probability of only 0.5. Thus, using least confidence, the learner would select $d_2$

# Uncertainty Sampling (III)

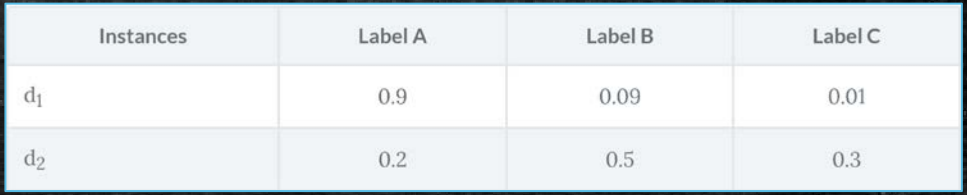
- Partial solution: margin sampling:

$$x^*_M = \underset{x}{\operatorname{argmin}}\, P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)$$

where $\hat{y}_1$ and $\hat{y}_2$ are the first and second most probable class labels under the model

- Pros:
  - Corrects the least confident strategy by incorporating the posterior of the second most likely label
  - Appropriate if the objective function is to reduce classification error

- Cons:
  - For problems with very large label sets, it still ignores the output distribution for the remaining classes



| Instances | Label A | Label B | Label C |
|---|---|---|---|
| $d_1$ | 0.9 | 0.09 | 0.01 |
| $d_2$ | 0.2 | 0.5 | 0.3 |

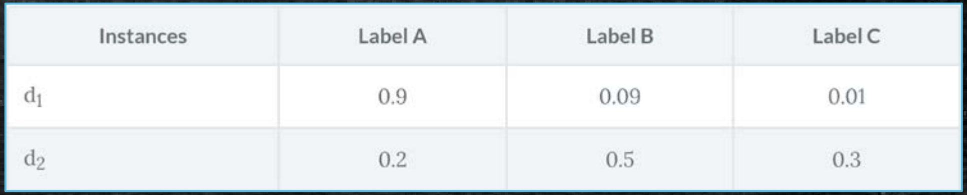
In the example, for d1, the difference between its first and second most probable labels is 0.81 (0.9 - 0.09) and for d2 it is 0.2 (0.5 - 0.3). Hence, the learner will select d2 again

# Uncertainty Sampling (IV)

- Solution: Entropy

$$x_H^* = \underset{x}{\operatorname{argmax}} - \sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)$$

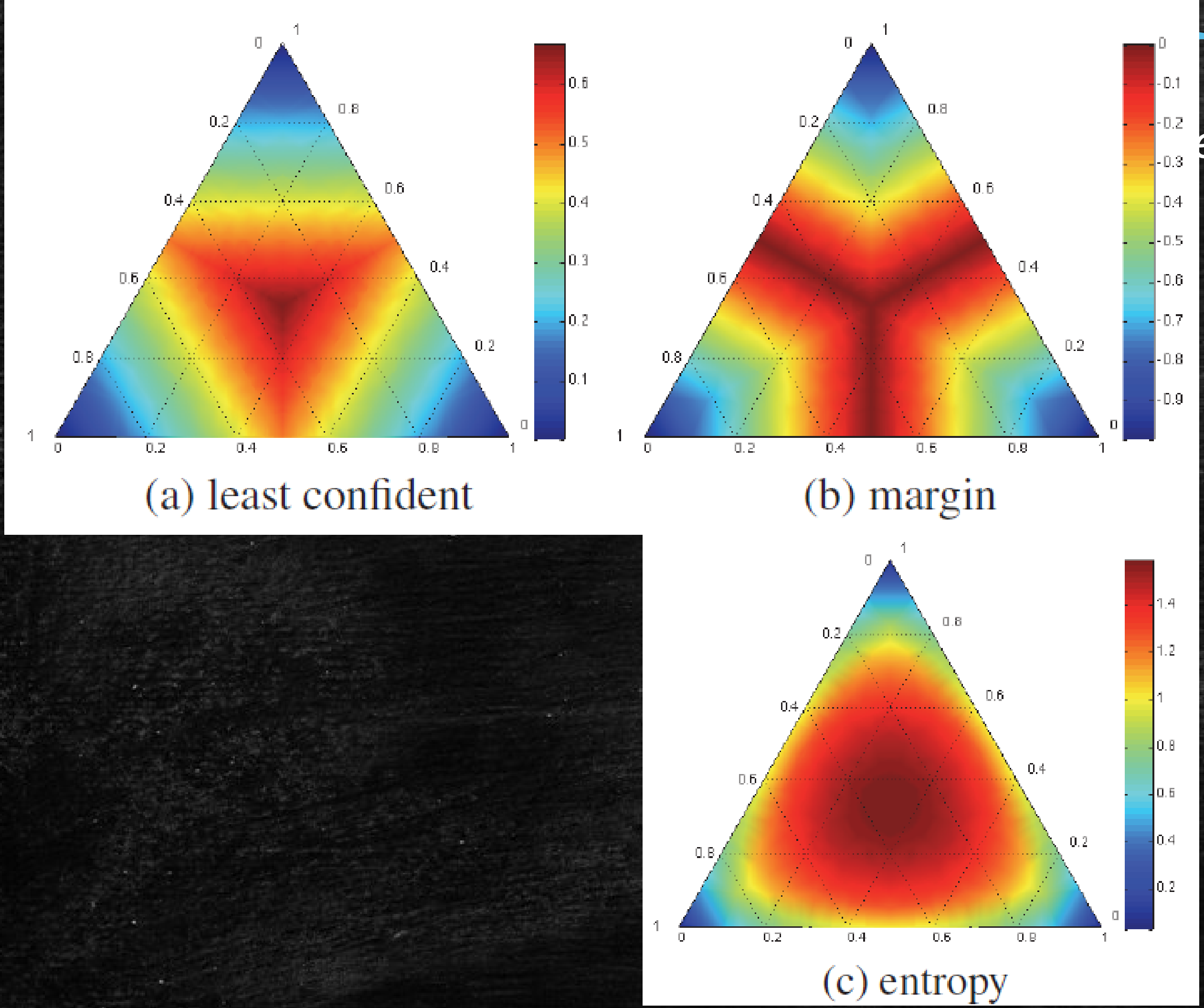
- Where $y_i$ ranges over all possible labeling
- Entropy is an information-theoretic measure that represents the amount of information needed to “encode” a distribution
- Pro:
  - Generalizes easily the strategy for probabilistic multi-label classifiers and for more complex structured instances (e.g. sequences, trees)
  - Appropriate if the objective function is to minimize log-loss (Log Loss is -1 * the log of the likelihood function)



| Instances | Label A | Label B | Label C |
|---|---|---|---|
| $d_1$ | 0.9 | 0.09 | 0.01 |
| $d_2$ | 0.2 | 0.5 | 0.3 |

The entropy formula is applied to each instance and the instance with the largest value is queried. Using our example, d1 has a value of 0.155 while d2's value is 0.447 and so the learner will select d2 once again.

# Uncertainty Sampling (V)



(a) least confident

(b) margin

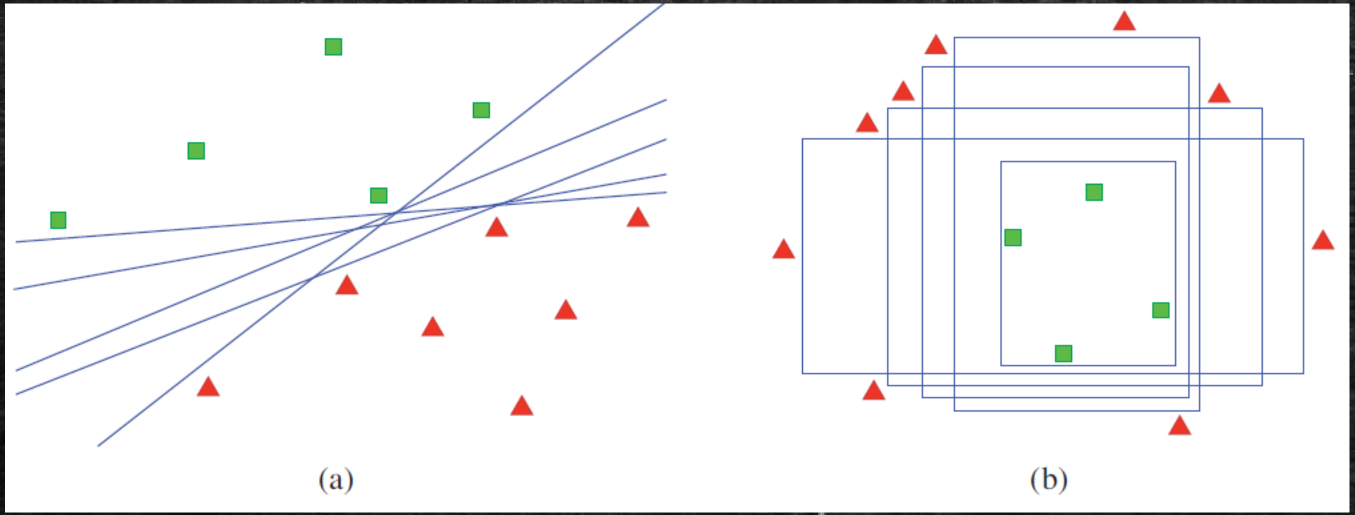
(c) entropy

- For all three measures, the most informative instance lies at the center of the triangular simplex, because this represents where the posterior label distribution is most uniform (most uncertain under the model).
- The least informative instances are at the three corners, where one of the classes has extremely high probability (little model uncertainty).
- The entropy measure does not favor instances for which only one of the labels is highly unlikely (i.e., along the outer side edges), because the model is fairly certain that it is not the true label.
- The least confident and margin measures, on the other hand, consider such instances to be useful if the model has trouble distinguishing between the remaining two classes (i.e., at the midpoint of an outer edge).

# Query-By-Committee (I)

- The QBC approach involves maintaining a committee C of models which are all trained on the current labeled set L, but represent competing hypotheses
- Each committee member votes on the labeling of query candidates
- Pick the instances generating the most disagreement among hypotheses
- Goal of QBC: minimize the version space (the set of hypotheses that are consistent with the current labeled trained data L), to constrain the size of this space as much as possible with a few labeled instances as possible

# Query-By-Committee (II)



*Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in L (as indicated by shaded polygons), but each represents a different model in the version space*

- To implement a QBC selection algorithm:
  - Construct a committee of models that represent different regions of the version space
  - Have some measure of disagreement among committee members

# Query-By-Committee (III)

- Two approaches for measuring the level of disagreement
- 1) Vote entropy:

$$x^*_{VE} = \underset{x}{\operatorname{argmax}} - \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}$$

  - Where $y_i$ ranges over all possible labeling,
  - $V(y_i)$ is the number of “votes”, *C* is the committee size
- 2) Kullback-Leibler (KL) divergence:

$$x^*_{KL} = \underset{x}{\operatorname{argmax}} \frac{1}{C} \sum_{c=1}^{C} D(P_{\theta^{(c)}} \| P_{\mathcal{C}})$$

  - Where: $$D(P_{\theta^{(c)}} \| P_{\mathcal{C}}) = \sum_i P_{\theta^{(c)}}(y_i|x) \log \frac{P_{\theta^{(c)}}(y_i|x)}{P_{\mathcal{C}}(y_i|x)}$$
  - $\theta^{(c)}$ represents a particular model, *C* is the committee as a whole

# Query-By-Committee (IV)

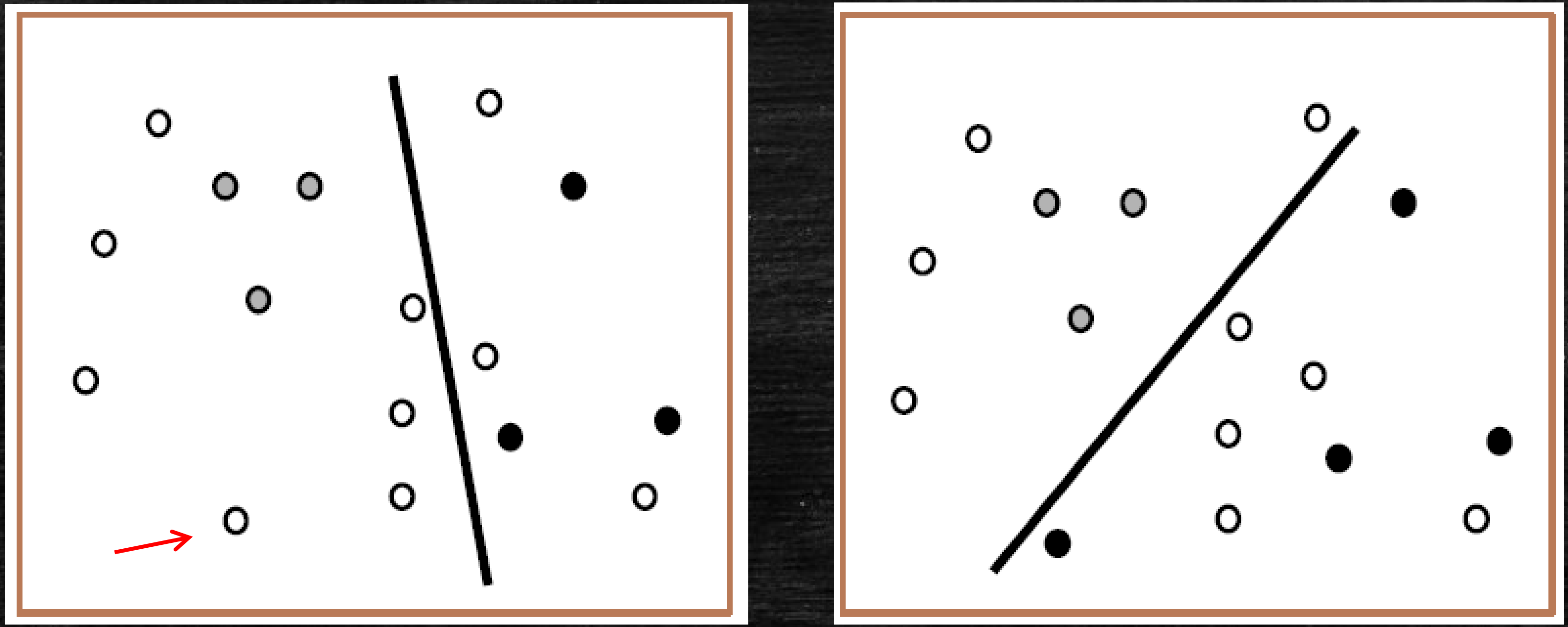
- So the “consensus” probability that $y_i$ is the correct label is given by:

$$P_{\mathcal{C}}(y_i|x) = \frac{1}{C}\sum_{c=1}^{C} P_{\theta^{(c)}}(y_i|x)$$

- KL divergence is an *information-theoretic* measure of the difference between two probability distributions
- Aside from QBC, other strategies attempt to minimize the version space:
  - A selective sampling algorithm that uses a committee of 2 NN, the “most specific” and the “most general” models (Cohn et al., 1994)
  - Pool-based margin strategy for SVMs (Tong and Koller, 2000)
  - The membership query algorithms (Angluin and King et al.)

# Expected Model Change (I)

- Is a decision-theoretic approach
- It selects the instance that would impart the greatest change to the current model if we knew its label



# Expected Model Change (II)

- Expected Gradient Length (EGL)
- It uses gradient-based training
- The learner should query the instance *x* which, if labeled and added to *L*, would result in the new training gradient of the largest magnitude
- It calculates the length as an expectation over the possible labeling:

$$x^*_{EGL} = \underset{x}{\operatorname{argmax}} \sum_i P_\theta(y_i|x) \left\| \nabla \ell_\theta(\mathcal{L} \cup \langle x, y_i \rangle) \right\|$$

- With $\nabla \ell_\theta(\mathcal{L})$ the gradient of the objective function ℓ with respect to the model parameters θ
- $\nabla \ell_\theta(\mathcal{L} \cup \langle x, y_i \rangle) \approx \nabla \ell_\theta(\langle x, y_i \rangle)$ the new gradient that would be obtained by adding the training tuple <*x*,*y*> to *L*

# Expected Model Change (III)

- It prefers instances that are likely to most influence the model, regardless of the resulting query label

- Cons:
  - Computationally expensive if both the feature space and set of labeling are very large
  - If features are not properly scaled, EGL approach can be led astray (Solution: parameter regularization)

# Density-Weighted Methods (I)

- Previous methods spend time querying possible outliers simple because they are controversial

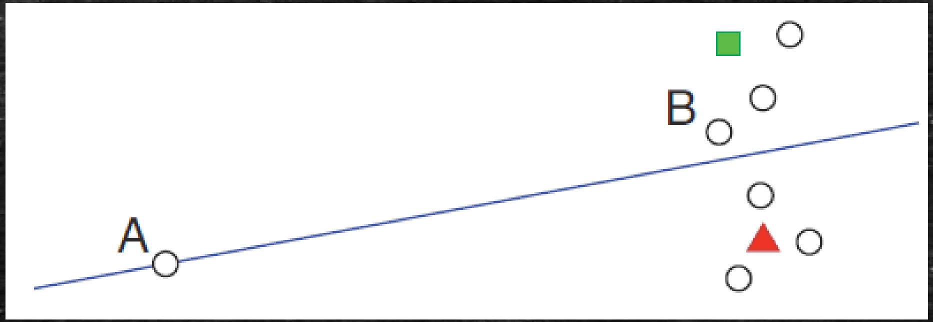
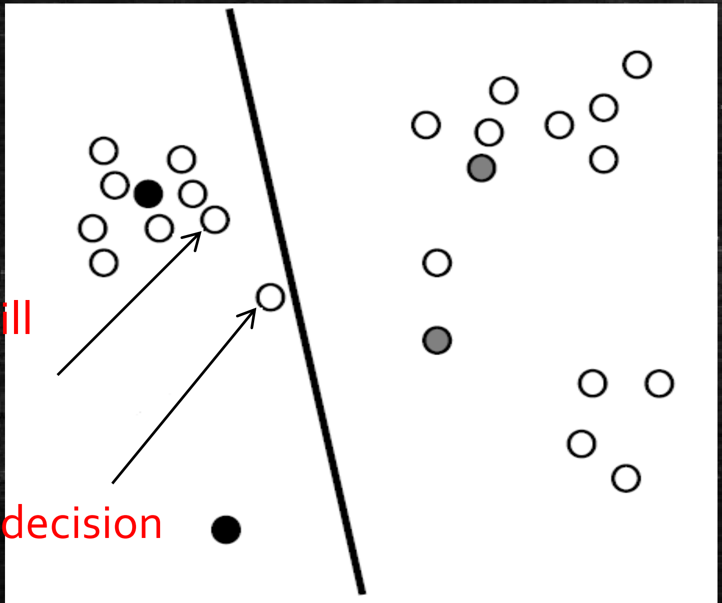


Figure: Poor strategy for classification when using uncertainty sampling can be a poor strategy for classification. Since *A* is on the decision boundary, it would be queried as the most uncertain. However, querying *B* is likely to result in more information about the data distribution as a whole

- The basic idea is to query the most “informative” and “representative” (inhabit dense regions of the input space)

# Density-Weighted Methods (II)

- 1) The information density framework (Settles and Craven, 2008)
- Main idea: informative instances should not only be those which are uncertain, but also those which are "representative" of the underlying distribution.

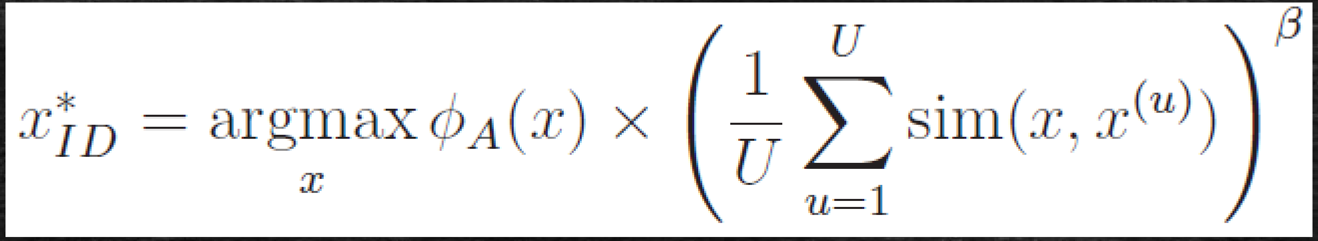


In densest region but still very close to decision boundary

Very close to decision boundary

# Density-Weighted Methods (III)

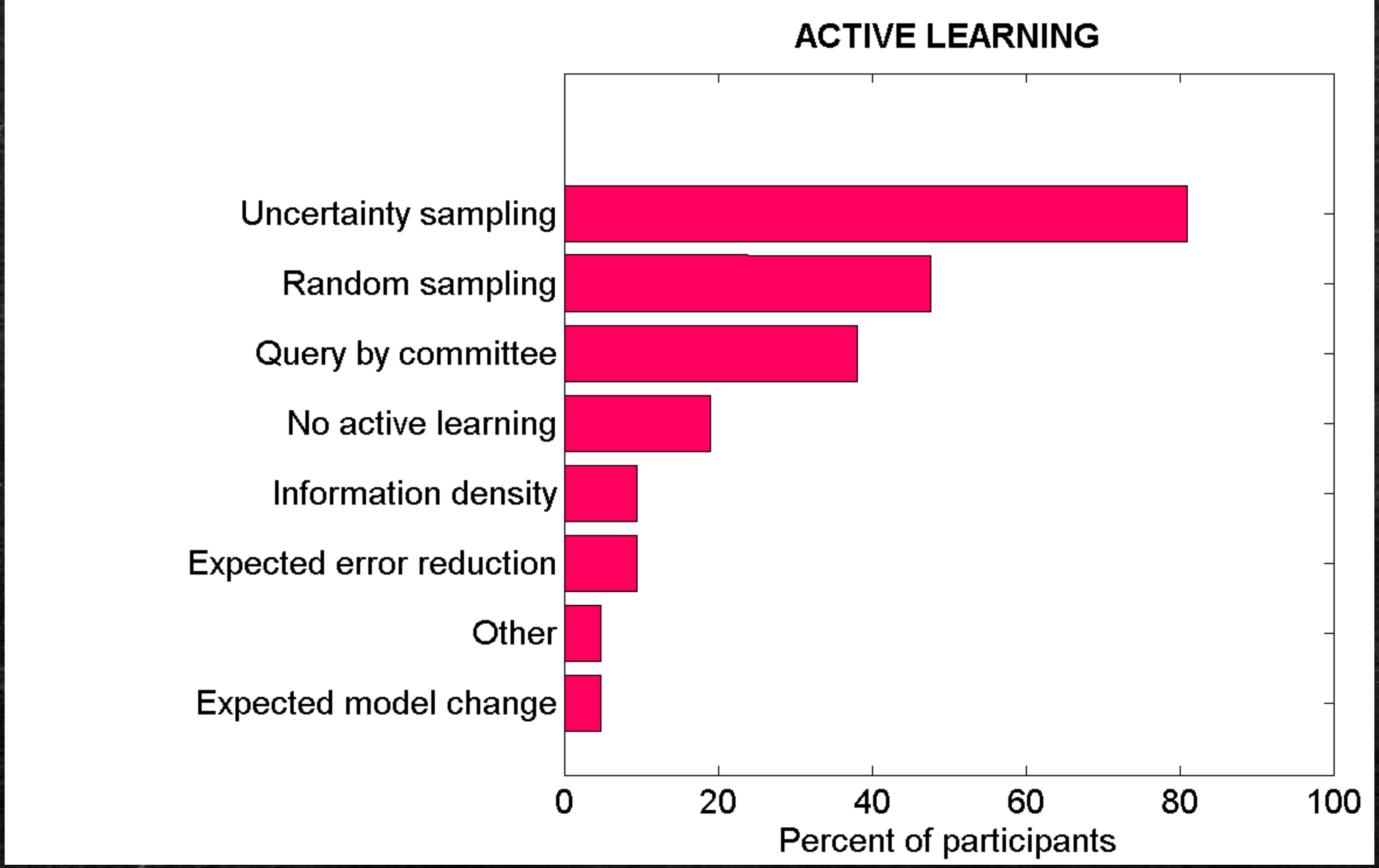
- The Settles and Craven approach:



$$x^*_{ID} = \underset{x}{\operatorname{argmax}}\, \phi_A(x) \times \left( \frac{1}{U} \sum_{u=1}^{U} \operatorname{sim}(x, x^{(u)}) \right)^{\beta}$$

- Where ϕ*A*(*x*) represents the informativeness of *x* according to some "base" query strategy *A*, such as an uncertainty sampling or QBC approach. The second term weights the informativeness of *x* by its average similarity to all other instances in the input distribution (as approximated by *U*), subject to a parameter β that controls the relative importance of the density term

# Usage Comparison of Different Active Learning Approach



[Guyon, Cawley , Dror , Lemaire 2009]
[Active Learning Challenge ]
http://www.causality.inf.ethz.ch/activelearning.php#cont

# Practical Considerations

- Until very recently the main question has been “Can machines learn with fewer training instances if they ask questions?”
  - Yes (subject to some assumptions e.g., single oracle, oracle is always right, cost of labeling instances is uniform)
- These assumptions don’t hold in most real world situations
- The research trend has now taken a turn to try and answer the question “Can machines learn **more economically** if they ask questions?”

# Stopping Criteria

- When is it good to stop?
  1. Cost of acquiring new training data is greater than the cost of the errors made by the current model
  2. Recognize when the accuracy plateau has been reached

- The real stopping criterion for practical applications is based on economic or other external factors, which likely come well before an intrinsic learner-decided threshold

# Analysis of Active Learning

- A number of research in the field and works done by large corporations like Google, IBM, CiteSeer suggest that it works
- It is however very closely related to the quality of the annotators
- It does reduce the number of instance labels required to achieve a given level of accuracy in most of the studied research
- However there are reported research in which the users opted out of active learning as they lacked the confidence in the approach

# Conclusions

- Active learning is a growing area of research in machine learning backed by the fact that the data available is ever growing and is available for less
- A lot of the research is involved in improving the accuracy of the learner with lesser number of queries
- Current day research is intended at implementing active learning in practical scenarios and problem solving