Lecture 7: More on Learning Theory. Introduction to Active Learning

- VC dimension
- Definition of PAC learning
- Motivation and examples for active learning
- Active learning scenarios
- Query heuristics

With thanks to Burr Settles, Sanjoy Dasgupta, John Langford for active learning part

The Vapnik-Chervonenkis (VC) Dimension

- The *Vapnik-Chervonenkis dimension*, $VC(\mathcal{H})$, of hypothesis space $\mathcal{H}$ defined over input space $\mathcal{X}$ is the size of the largest finite subset of $\mathcal{X}$ shattered by $\mathcal{H}$. If arbitrarily large finite sets of $\mathcal{X}$ can be shattered by $\mathcal{H}$, then $VC(\mathcal{H}) \equiv \infty$.
- In other words, the VC dimension is the maximum number of points for which $\mathcal{H}$ has no approximation error (is capable of making no mistakes, regardless of the actual target)
- VC dimension measures how many distinctions the hypotheses from $\mathcal{H}$ are able to make
- This is, in some sense, the number of "effective degrees of freedom"

Establishing the VC dimension

- Play the following game with the enemy:
  - You are allowed to *choose k points*. This actually gives you a lot of freedom!
  - The enemy then labels these points any way it wants
  - You now have to produce a hypothesis, out of your hypothesis class, which correctly matches these labels.

  If you are able to succeed at this game, the *VC dimension is at least k*.
- To show that it is *no greater than k*, you have to show that for any set of $k+1$ points, the enemy can find a labeling that you cannot correctly reproduce with any of your hypotheses.

Example revisited: VC dimension of two-sided intervals

- Suppose we have a hypothesis set that labels all points inside an interval $[a, b]$ as class 1. What is its VC dimension?
- Can we shatter 2 points on a line with a two-sided interval?

  Yes!
- Can we shatter 3 points on a line with one interval?

  No! The enemy can label the most distant points 1 and the middle one 0
- What is the VC dimension of intervals?

  VC dimension is 2
- Note that if we allow the class inside the interval to be 1 or 0, we could do 3 points too, but in this case, we have an extra "degree of freedom" (the class inside the interval, in addition to its boundaries)

VC dimension of linear decision surfaces

- Consider a linear threshold unit in the plane.
- First, show there exists a set of 3 points that can be shattered by a line $\Longrightarrow$ VC dimension of lines in the plane is at least 3.
- We do this by picking 3 non-colinear points, labelling them all possible ways, and picking lines that correctly separate them
- To show it is at most 3, show that NO set of 4 points can be shattered.
- For this we have to consider all qualitative layouts of the points (all in a line, 3 on a line and one off it, 3 points forming a convex hull with the 4th inside, and 4 points forming a convex hull)
- For an n-dimensional space, one can generalize this reasoning to show that the VC dimension of linear estimators is $n + 1$.

Error bounds using VC dimension

- Recall our error bound in the finite case:

$$e(h_{emp}) \leq \left(\min_{h \in \mathcal{H}} e(h)\right) + 2\sqrt{\frac{1}{2m} \log \frac{2|\mathcal{H}|}{\delta}}$$

- Vapnik showed a similar result, but using VC dimension instead of the size of the hypothesis space:
- For a hypothesis class $\mathcal{H}$ with VC dimension $VC(\mathcal{H})$, given m examples, with probability at least $1 - \delta$, we have:

$$e(h_{emp}) \leq \left(\min_{h \in \mathcal{H}} e(h)\right) + O\left(\sqrt{\frac{VC(\mathcal{H})}{m} \log \frac{m}{VC(\mathcal{H})} + \frac{1}{m} \log \frac{1}{\delta}}\right)$$

Remarks on VC dimension

- The previous bound is tight up to log factors. In other words, for hypotheses classes with large VC dimension, we can show that there exists some data distribution which require a number of examples matching the upper bound.
- For many reasonable hypothesis classes (e.g. linear approximators) the VC dimension is linear in the number of "parameters" of the hypothesis.
- This shows that to learn "well", we need a number of examples that is linear in the VC dimension (so linear in the number of parameters, in this case).
- However, in other cases (e.g. neural nets) the VC dimension may depend on other factors (eg. the magnitude allowed for the parameters)
- An important property: if $\mathcal{H}_1 \subseteq \mathcal{H}_2$ then $VC(\mathcal{H}_1) \leq VC(\mathcal{H}_2)$.

Structural risk minimization

$$e(h_{emp}) \leq \left(\min_{h\in\mathcal{H}} e(h)\right) + O\left(\sqrt{\frac{VC(\mathcal{H})}{m}\log\frac{m}{VC(\mathcal{H})} + \frac{1}{m}\log\frac{1}{\delta}}\right)$$

- As before we can use this bound to pick the hypothesis class that minimizes the upper bound (so, to do model selection)
- In other words, we can use the VC dimension for *structural risk minimization*

Probably Approximately Correct (PAC) Learning

Let $\mathcal{F}$ be a concept (target function) class defined over a set of instances $\mathcal{X}$ in which each instance has n attributes. An algorithm L, using hypothesis class $\mathcal{H}$ is a *PAC learning algorithm* for $\mathcal{F}$ if:

- for any concept $f \in \mathcal{F}$
- for any probability distribution P over $\mathcal{X}$
- for any parameters $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$

the learner L will, with probability at least $(1 - \delta)$, output a hypothesis with true error at most ϵ.

A class of concepts $\mathcal{F}$ is *PAC-learnable* if there exists a PAC learning algorithm for $\mathcal{F}$.

Computational vs Sample Complexity

- A class of concepts is *polynomial-sample PAC-learnable* if it is PAC learnable using a number of examples at most polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$ and n.
- A class of concepts is *polynomial-time PAC-learnable* if it is PAC learnable in time at most polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$ and n.
- Sample complexity is often easier to bound than time complexity!
- Sometimes there is a trade-off between the two (if there are more samples, less work is required to process each one and vice versa)

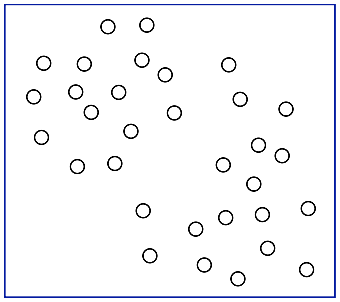
Summary

- The complexity results for binary classification show trade-offs between the desired degree of precision ϵ, the number of samples m and the complexity of the hypothesis space $\mathcal{H}$
- The complexity of $\mathcal{H}$ can be measured by the VC dimension
- For a fixed hypothesis space, minimizing the training set error is well justified (empirical risk minimization)
- We have not talked about the relationship between margin and VC dimension (better bounds than the results discussed)

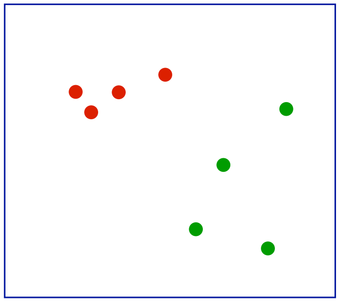
Passive supervised learning

- The environment provides labelled data in the form of pairs $(\mathbf{x}, y)$
- We can process the examples either as a batch or one at a time, with the goal of producing a predictor of y as a function of $\mathbf{x}$
- We assume that there is an underlying distribution P generating the examples
- Each example is drawn i.i.d. from P
- What if instead we are allowed to *ask for particular examples*?
- Intuitively, if we are allowed to ask questions, and if we are smart about what we want to know, fewer examples may be necessary

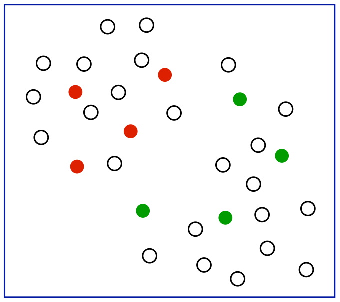
Semi-Supervised and Active Learning



Unlabeled points



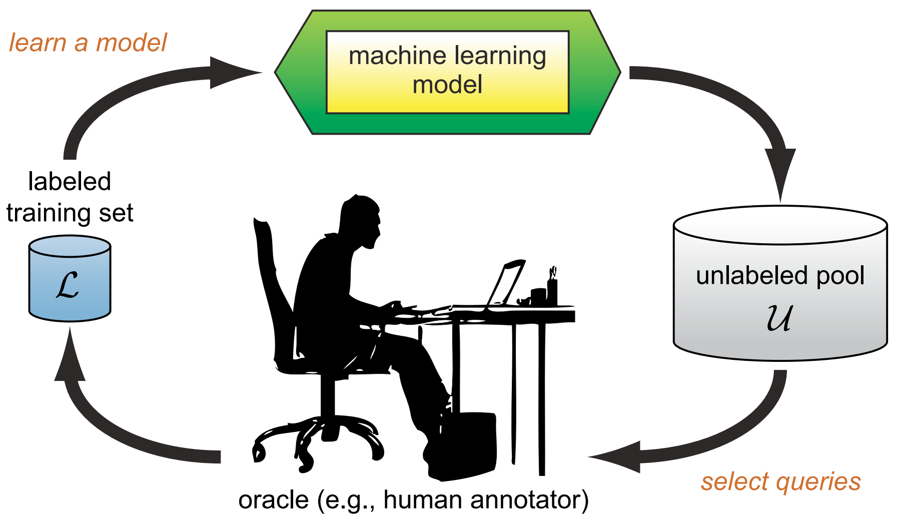
Supervised learning



Semisupervised and active learning

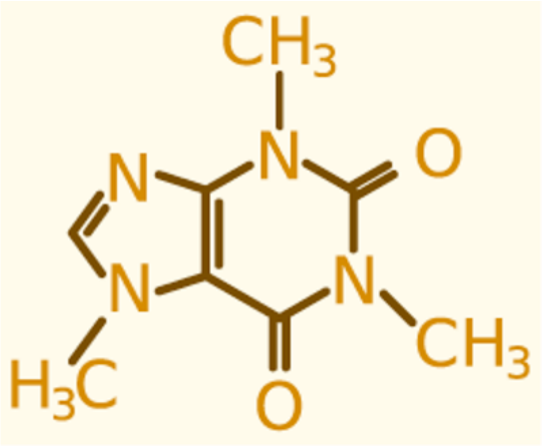
- Suppose you had access to a lot of unlabeled data
  E.g. all the documents on the web
  E.g. all the pictures on Instagram
- You can also get some labelled data, but not much
- How can we take advantage of the unlabeled data to improve supervised learning performance?

Active Learning



- The learner can query an “expert” for a label on any example
- The expert could be a person or a fancy automated program
- Queries are usually expensive or slow
- What examples should we ask for next?

Example: Drug Discovery (Warmuth et al., 2003)

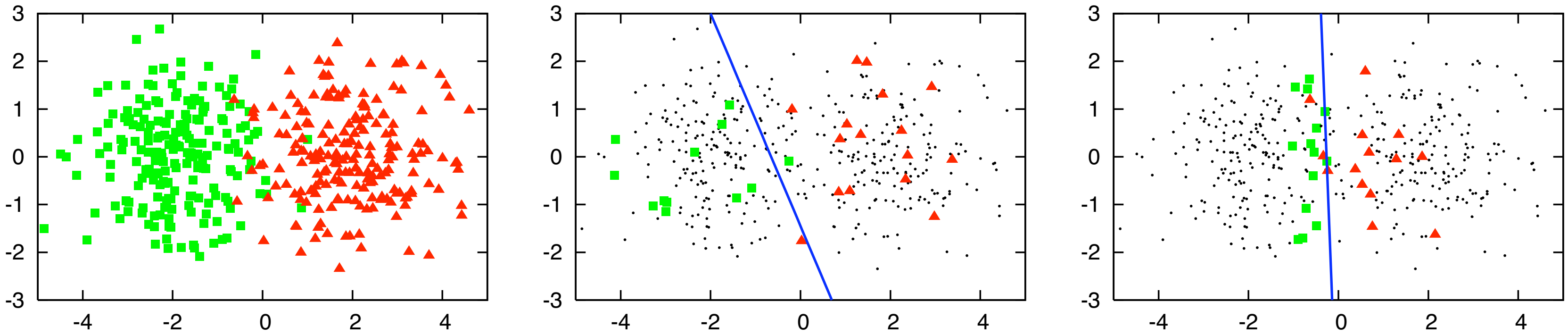


- We have access to many libraries of chemicals from different companies (millions of substances)
- Each chemical is described in a standard vector form (bonds, bond angles, groups...)
- Goal: establish if the chemical binds or not with a target
- Getting a label means physically performing a chemical reaction!

Applications

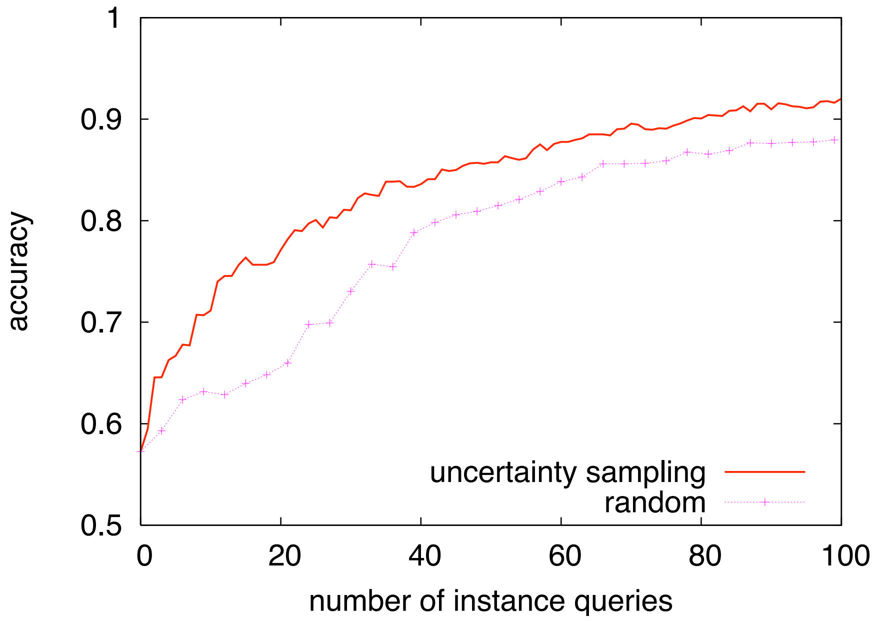
- Document classification
- Document tagging (e.g. determining parts-of-speech, semantic objects like places, names, ..)
- Image classification
- Image tagging (e.g. tag all people in a picture)
- Chemistry
- Biomedical applications (labels are obtained by asking a doctor)
- Robotics: what is the true position and velocity of the robot?

The active learning (potential) advantage



- Typically better accuracy, at the same number of instances, than can be obtained by random selection
- Queries that are selected may indicate problematic examples

Typical active learning curve

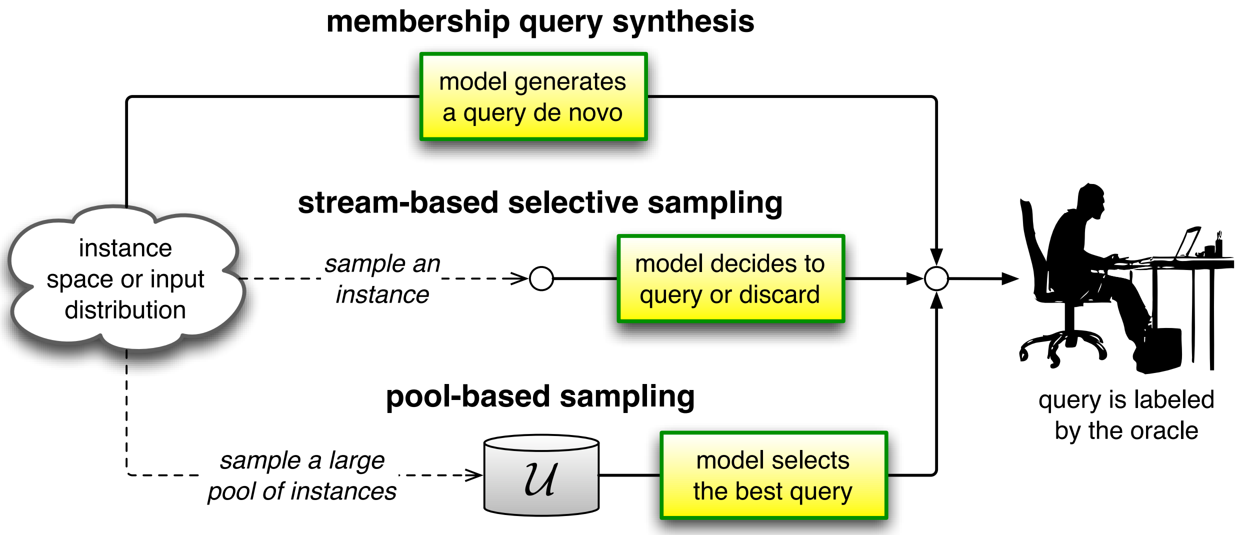


Informed sampling strategy is uniformly better, at all data set sizes

Relationship to supervised learning

- Active learning is a “wrapper” around a supervised learning algorithm
- Once a supervised data set has been obtained, we can used the usual algorithms (logistic regression, naive Bayes, decision or regression trees, SVMs, neural nets, Adaboost...) to get a hypothesis
- In principle, any query generation and sampling strategy can work with any supervised learner (though for theoretical guarantees we may need particular learners)
- In practice, certain combinations are better, e.g. due to the cost of re-fitting the classifier.

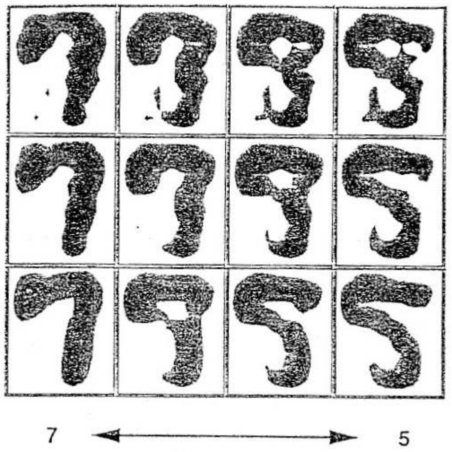
Generating queries


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

- Generate new examples (synthesizing all inputs)
- As each data point comes in, make a decision whether to query or not
- Consider a larger set of examples and pick the "best" one to query

Generating new examples (cf. Angluin)

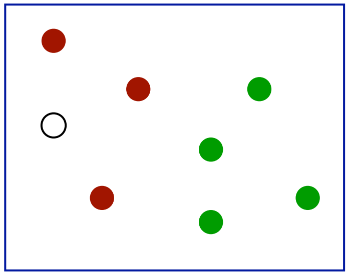
- Learner thinks of an input that would be confusing according to the current hypothesis and asks about it
- Nice theoretical guarantees: PAC-style bounds on the number of examples that need to be asked, in the noise-free case, before the target hypothesis can be correctly identified
- But the examples can be very tough for people to label!



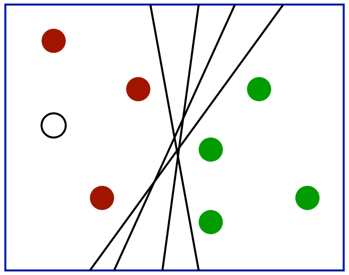
- The inputs are *not drawn form the true data distribution*

Stream-based sampling

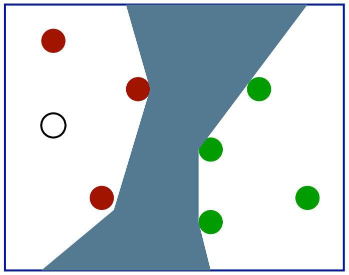
- Each instance has to be considered in isolation, and a binary decision is made whether to query or not
- Natural for problems in which data comes on-line and it would be hard to store
- Strategies:
  1. Trade off cost of query and “informativeness”
  2. Query if the instance is within the current region of uncertainty



Is a label needed?



H_t = current candidate hypotheses



Region of uncertainty

Problem: maintaining the region of uncertainty in the general case is hard, so it needs approximations

Pool-based sampling

- A pool of instances (possibly big!) is considered
- The “best” instance is picked (according to some criterion)
- Decisions are more informed than in stream-based sampling, but the memory and computation cost can be much higher

Query strategies

- Intuitively, the learner should ask about instances about which it is uncertain
- Several heuristics to implement this idea:
  - Uncertainty sampling
  - Query-by-committee
  - Expected impact of the instance on the decision boundary
- Relationship to other instances may also be important

Uncertainty sampling strategies

- Classification:
  1. Ask about the instance for which the most likely class is very uncertain E.g., in a probabilistic classifier, the best input $\mathbf{x}$ is given by:

     $$\mathbf{x}^* = \arg\max_{\mathbf{x}}(1 - \max_{y_i} P(y_i|\mathbf{x}))$$

  2. Ask about the instance where the class label has the highest entropy

     $$\mathbf{x}^* = \arg\max_{\mathbf{x}} \left(-\sum_{y_i} P(y_i|\mathbf{x}) \log P(y_i|\mathbf{x}) \right)$$

  3. Ask about the instance for which the top two classes have close probability
- Regression: ask about the instance with highest variance.

Query-by-committee

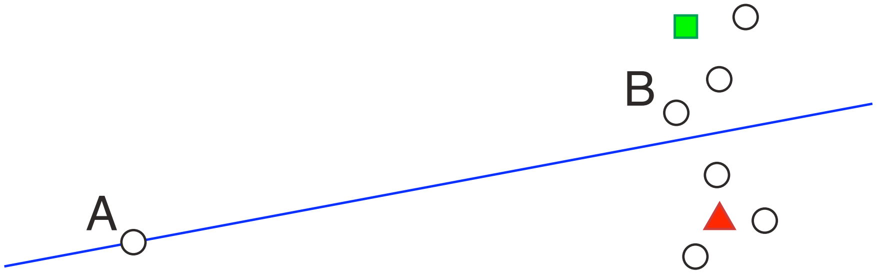
- You have a set of hypotheses that get to vote on the example
- Examples on which there is a lot of disagreement make good queries

  E.g., for which the entropy of the distribution generated is high, or the KL-divergence between the distributions predicted by each hypothesis is high
- Hypotheses may be trained on different subsets of attributes

Expected error reduction/Maximum information gain

- Consider the impact that the instance would have on the rest of the set U
- Goal: reduce the entropy in the U labels after the instance is used for training
- Setup:
  - Consider an input $\mathbf{x} \in U$ and pretend you will label it *in all possible ways*
  - Each label y_i has some probability
  - Consider adding $(\mathbf{x}, y_i)$ to the set of labelled data
  - Re-train the predictors on the new labelled data, and measure impact on the other unsupervised examples
- Ideally, this will lead to a more consistent labeling of the remaining unlabeled examples
- Can be very expensive

Density-based sampling



- Queries that are far away from the major concentration of the data are less useful
- Weigh the “informativeness” of the query (obtained according to one of the previous criteria) by its average similarity to the rest of the unlabeled set U
- Requires a distance measure between inputs.