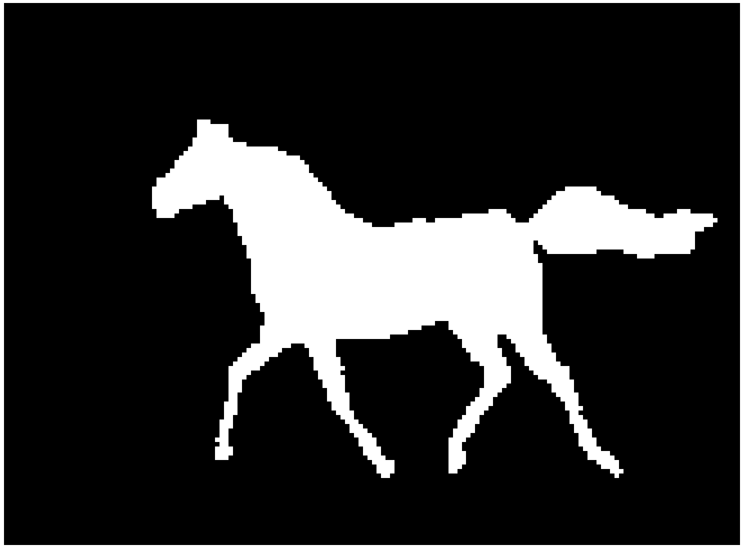
Learning With Less Data: Active, Semi-Supervised, and Self-Supervised Learning

Rishabh Iyer
University of Texas at Dallas

Supervised Learning

- We’re given lots and lots of labelled examples
  - Goal is to predict the label of unseen examples
  - Observations:
    - We don’t necessarily need that many data points to construct a good classifier (think SVMs)
    - In certain applications, labels are ***expensive***
      - They can cost time, money, or other resources

Image Segmentation



Someone (probably a graduate student) had to produce these labels by hand!

Expensive Data

- In general, data is easy to come by but labels are expensive
  - Labelled speech
  - Labelled images and video
  - Large corpora of texts
- These tasks are mind numbing and boring
  - Can pay people to do them! (Amazon Mechanical Turk)
  - Can get expensive fast and we need some way to ensure that they are accurately solving the problem or else we are wasting money!

Active Learning

- Given lots of unlabeled examples
  - Learn to predict the label of unseen data points
  - The added feature: we have the ability to ask for the label of any one of the unlabeled inputs (e.g., a labelling oracle/expert)
    - Treat asking the oracle for a label as an expensive operation
    - The performance of the algorithm will be judged by how few queries it can make to learn a good classifier

Related to Experimental Design

- Suppose that we want to determine what disease a patient has
  - We can run a series of (possibly expensive) tests in order to determine the correct diagnosis
  - How should we choose the tests so as to minimize cost (dollars and life) while still guaranteeing that we come up with the correct diagnosis?

A First Attempt

- Could just randomly pick an unlabeled data point
  - Request its label
  - Add it to the training data
  - Retrain the model
  - Repeat
- If labels are really expensive, can be a terrible idea
  - Many unlabeled data points may have very little impact on the predicted labels
  - This is effectively the supervised setting

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

Ideal case: number of hypotheses consistent with the labeling is approximately halved at each step

Types of Active Learning

- Pool based
  - We're given all of the unlabeled data upfront
- Streaming
  - Unlabeled examples come in one at a time and we have to decide whether or not we want to label them as they arrive
  - Also applies to applications in which storing all the data is not possible

Basic Strategy

- Iteratively build a model
- Use the current model to find “informative” unlabeled examples
- Select the most informative example(s)
  - Label them and add them to the training data
- Retrain the model using the new training data
- Repeat

Basic Strategy

- Iteratively build a model
- Use the current model to find “informative” unlabeled examples
- Select the most informative example(s)
    - Label them and add them to the training data
- Retrain the model using the new training data
- Repeat

Note: this procedure will result in a biased sampling of the underlying distribution in general (the actively labeled dataset is not reflective of the underlying data generating process)

Informative Examples

- For learning algorithms that model the data generating process...
  - A data point is informative if the current model is not confident in its prediction for this example
  - Least confident labeling (binary label case):

$$\arg \max_{x \text{ unlabeled}} 1 - \max_{y} p(y|x, \theta)$$

- For learning algorithms, like SVMs, that are simply selecting among a collection of hypotheses...
  - Unlabeled data points that are far from the current decision boundary are unlikely to provide useful information

Query-by-Committee

- Select a committee of T consistent classifiers using the labeled data
- Find examples for which the committee has the largest disagreement
  - For example, in a binary labeling problem, find the examples for which the committee's votes are split as close to 50/50 as possible between +1 and -1
- Request the label for these examples

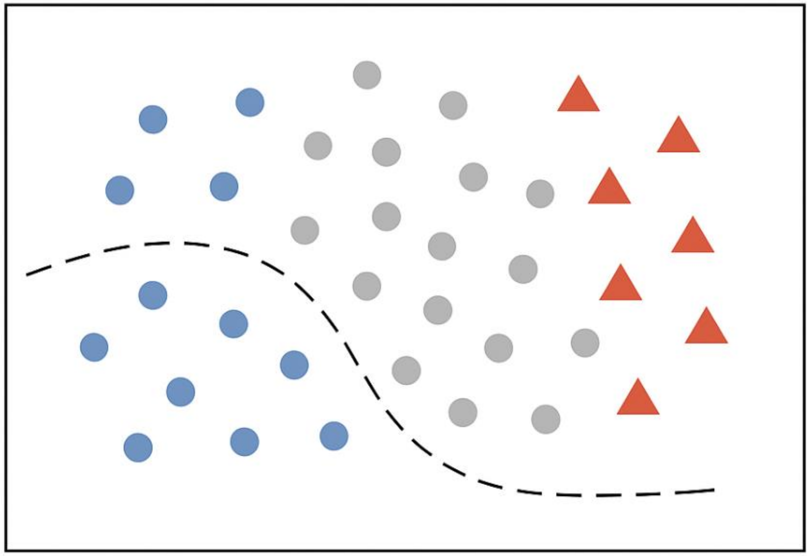
Goal: reduce the version space as much as possible by selecting points whose label will eliminate the most hypotheses

Query-by-Committee

- Key Idea:
  - Maintain a **committee of diverse models** (e.g., via different initializations, subsets of data, or architectures).
  - For each unlabeled example, evaluate how much the models **disagree** in their predictions.
  - **Select examples with the highest disagreement** to query for labels from the oracle (e.g., a human annotator).
- Why it Works?
  - High disagreement implies **high model uncertainty**.
  - Labeling such samples **reduces hypothesis space** more efficiently.
  - Leads to faster learning with **fewer labeled samples**.

Semi-supervised Learning

- Given a collection of labeled and unlabeled data, use it to build a model to predict the labels of unseen data points
  - We never get to see the labels of the unlabeled data
  - However, if we assume something about the data generating process, the unlabeled data can still be useful...



Semi-Supervised Learning

💡 **Core Idea:**

- Semi-supervised learning sits between **supervised** and **unsupervised learning**, leveraging **a small set of labeled data** + **a large set of unlabeled data** to improve learning.

📚 **Why It Matters:**

- Labeling data is **expensive and slow**
- Unlabeled data is **abundant and cheap**
- SSL bridges the gap by exploiting structure in the data distribution

Key Assumptions in SSL

- **Smoothness Assumption**: Close points likely share the same label
- **Cluster Assumption**: Data forms clusters; points in the same cluster likely share a label
- **Manifold Assumption**: Data lies on a lower-dimensional manifold

SSL Techniques – How It's Done

1. **Pseudo-Labeling**:

- Use the model to assign "pseudo-labels" to unlabeled data
- Retrain model using both true + confident pseudo-labels
- 🔁 Repeat iteratively

2. **Consistency Regularization**:

- Add a loss that encourages predictions to be **stable** under small input perturbations
- Ex: If an image is flipped or augmented, the model should still predict the same label

SSL Techniques – How It's Done

3. **Graph-Based SSL**:

- Represent data as a graph (nodes = samples, edges = similarity)
- Propagate labels from labeled to unlabeled nodes

4. **Entropy Minimization & Confidence-Based Filtering**:

- Prefer confident predictions; penalize uncertain outputs
- Can be combined with pseudo-labeling or consistency

Please evaluate the course!

eval.utdallas.edu