Lecture 18: Semi-Supervised and Active Learning

COMP90049

Semester 2, 2020

Lida Rashidi, CIS

Acknowledgement: Tim Baldwin, Karin Verspoor & Kris Ehinger

Semi-supervised Learning

Where we're at so far

- To date, we have talked a lot about **supervised learning** where we have assumed (fully) labelled training data
- We also talked about **unsupervised learning** where we have (fully) unlabelled training data
- What if we had a small amount of labelled training data, and lots of unlabelled training data?
- What if we had a small amount of labelled training data and a limited budget to label more training data?

Why bother?

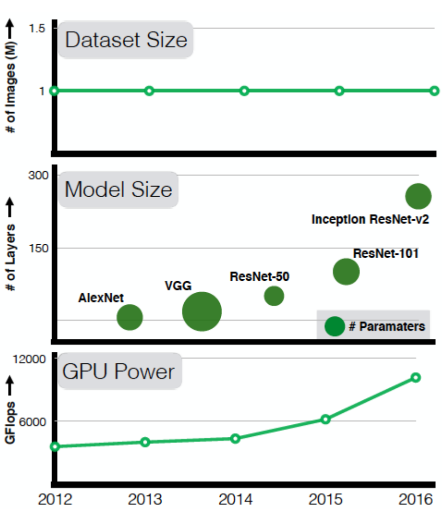
- “Simple models and a lot of data trump more elaborate models based on less data!”[1]
- (Labelled) data is a bottleneck for machine learning
  - labels may require human experts
  - labels may require special devices
- unlabelled data is often cheap and in large quantity

[1] Halevy, Norvig, & Pereira (2009) “The Unreasonable Effectiveness of Data”

Example I

In image classification task - Sun, Shrivastava, Singh, & Gupta (2017)

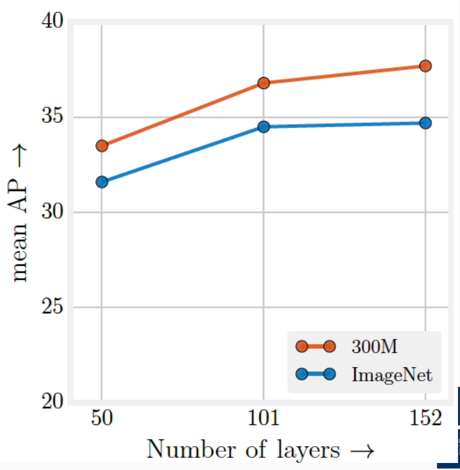
- model depth has increased dramatically
- AlexNet $\approx$ 10 layers $\rightarrow$ ResNet $>$ 150 layers
- the size of “large scale” datasets has not kept pace
- 1 Million labelled images



Example II

In image classification task - Sun, Shrivastava, Singh, & Gupta (2017)

- Adding data is nearly as effective as adding layers
- There is a limit to what a network can learn on a smaller dataset more parameters are not helpful unless you have more data



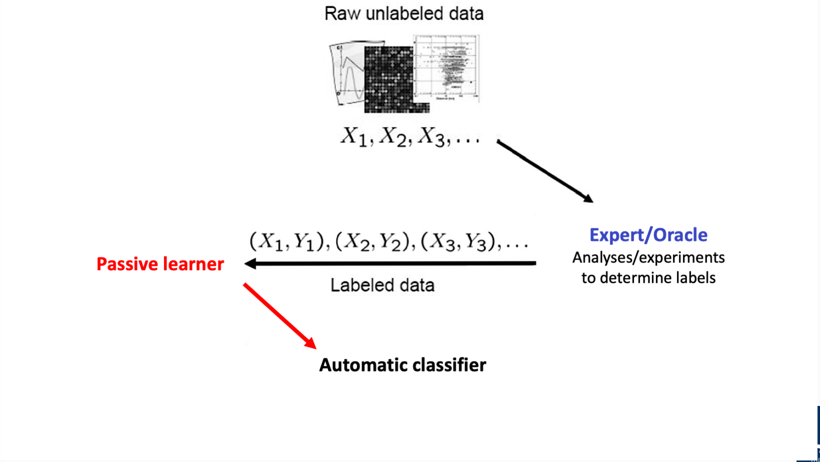
Semi-supervised learning

Semi-supervised learning I

- Semi-supervised learning is learning from both labelled and unlabelled data
- **Semi-supervised classification**:
  - L is the set of labelled training instances $\{x_i, y_i\}_{i=1}^{l}$
  - U is the set of unlabelled training instances $\{x_i\}_{i=l+1}^{l+u}$
  - Often $u \gg l$
  - Goal: learn a better classifier from $L \cup U$ than is possible from L alone

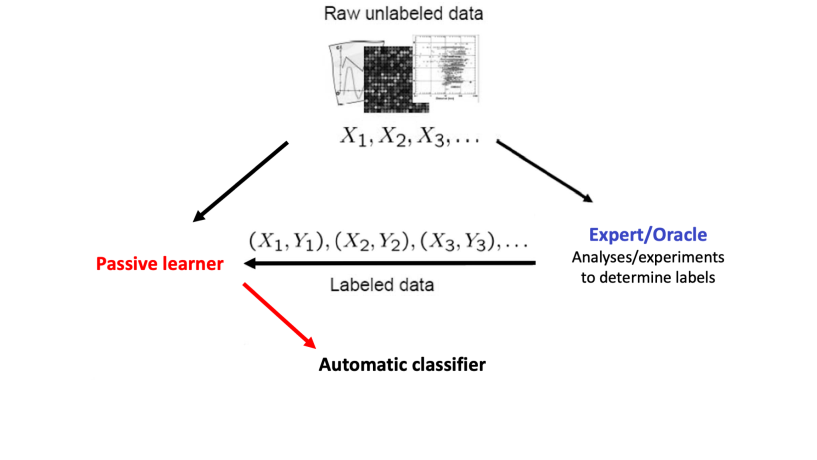
Supervised vs. Semi-supervised learning II

"Supervised" Learning:



Supervised vs. Semi-supervised learning II

"Semi-supervised" Learning:



Motivation of Semi-supervised learning

Cognitive science

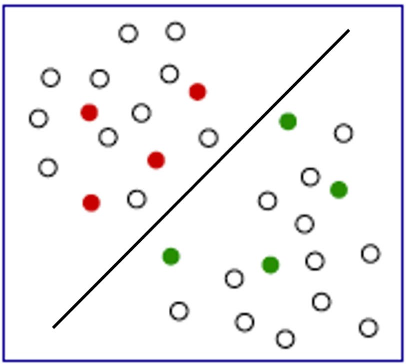
- model of how humans learn from labelled and unlabelled data
- concept learning in children: $x =$animal, $y =$concept (e.g., dog)
- You point to a brown animal and say “dog!”
- Children also observe animals by themselves

Hard-to-get Labels

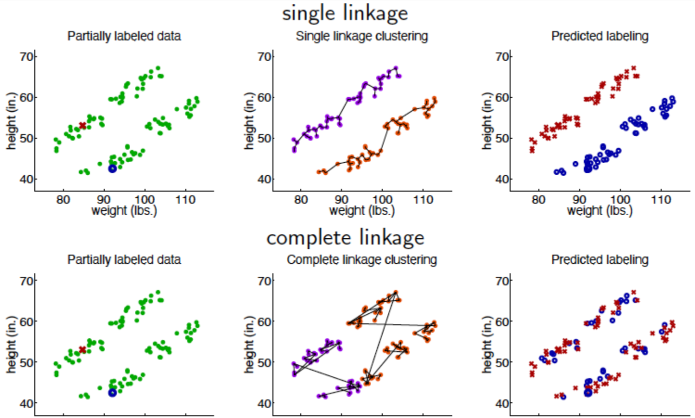
- Task: speech analysis
- Switchboard dataset and telephone conversation transcription
- 400 hours annotation time for each hour of speech

Semi-Supervised Learning Approach I

- A simple approach: combine a supervised and unsupervised model
- For example, Find clusters, choose a label for each (most common label?) and apply it to the unlabelled cluster members



Semi-Supervised Learning Approach I



Semi-Supervised Learning Approach II

Self-Training (Also known as "Bootstrapping")

- Assume you have $L = \{x_i, y_i\}_{i=1}^{l}$ labelled and $U = \{x_i\}_{i=l+1}^{l+u}$ unlabelled training instances
- Repeat
  - Train a model f on L using any supervised learning method
  - Apply f to predict the labels on each instance in U
  - Identify a subset U' of U with "high confidence" labels
  - Remove U' from U and add it to L with the classifier predictions as the "ground-truth" labels ($U \leftarrow U \backslash U'$ and $L \leftarrow L \cup U$)
  - Until L does not change

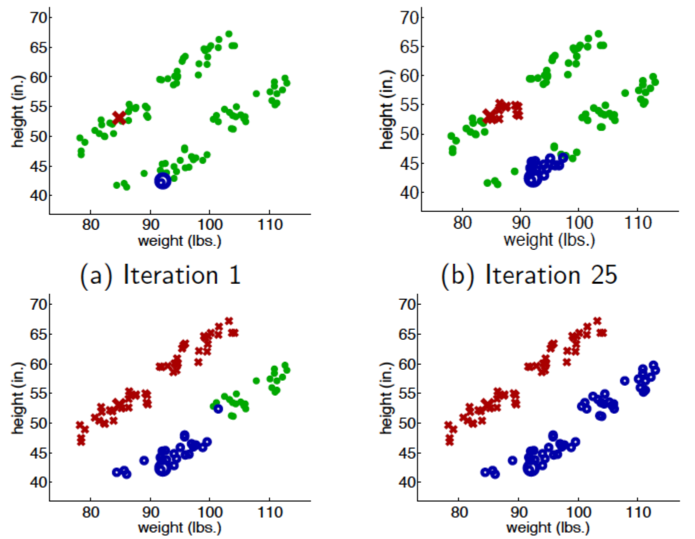
Self-Training Assumptions

- Propagating labels requires some assumptions about the distribution of labels over instances:
  - Points that are nearby are likely to have the same label
- Classification errors are propagated
  - One option is to move points back to the "unlabelled" pool if the classification confidence falls below a threshold

Self-Training Example: 1-NN

- 1-nearest neighbour with $L = \{x_i, y_i\}_{i=1}^{l}$ labelled and $U = \{x_i\}_{i=l+1}^{l+u}$ unlabelled training instances
- Repeat
    - Find neighbours for unlabelled instances in U
    - For instances x, whose nearest neighbour is in L, take the labels y' from 1-NN
    - $U \leftarrow U \backslash \{x\}$
    - $L \leftarrow L \cup \{x, y'\}$
    - Until L does not change

Self-Training Example: 1-NN



Self-Training Example: Naive Bayes

- Naive Bayes with $L = \{x_i, y_i\}_{i=1}^{l}$ labelled and $U = \{x_i\}_{i=l+1}^{l+u}$ unlabelled training instances
- Initialization: Train on L to learn $P(X|Y)$ and $P(Y)$ for all features X and all classes Y
- Repeat (EM algorithm)
  - **Expectation**: For each unlabelled instance, compute a probability distribution over classes
  - **Maximization**: Recompute $P(X|Y)$ and $P(Y)$ with all data, weighting the unlabelled instances by their probability of being in each class

Self-Training Example: Naive Bayes

- **Problem**: if the unlabelled dataset is much larger than the labelled dataset, probability estimates will be based almost entirely on unlabelled data
- **Solution**: add only a small amount of unlabelled data initially and gradually add more in later EM iterations

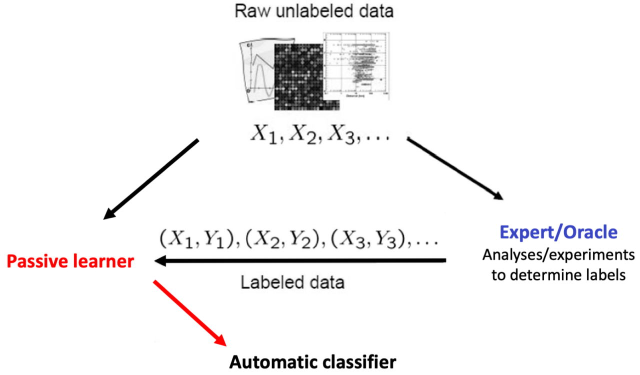
Active Learning

Active Learning I

- Active learning builds off the hypothesis that a classifier can achieve higher accuracy with fewer training instances if it is allowed to have some say in the selection of the training instances
- The underlying assumption is that labelling is a finite resource, which should be expended in a way which optimises machine learning effectiveness
- Active learners pose **queries** (unlabelled instances) for labelling by an **oracle** (e.g. a human annotator)

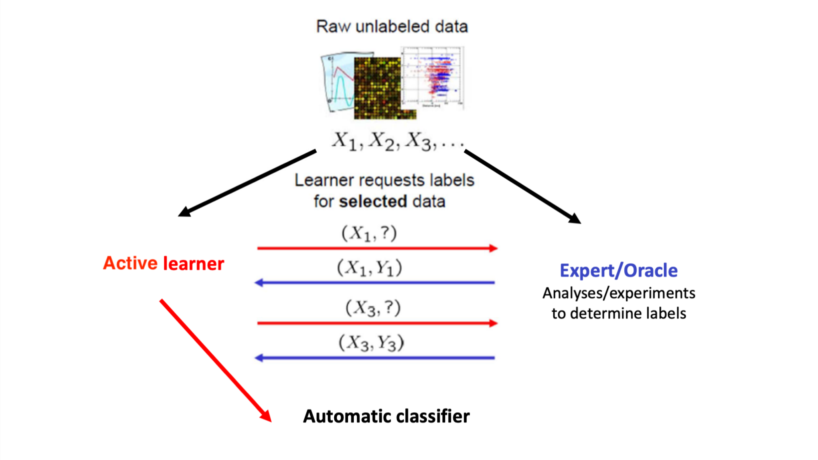
Semi-supervised learning vs. Active Learning

"Semi-supervised" Learning:



Semi-supervised learning vs. Active Learning

"Active" Learning:



Active Learning II

- Ideally, we'd want the instances that are most effective for distinguishing between competing models
  - To do this most efficiently, we should have some sense of the likelihood of different models, or knowledge of how labels are distributed over instances, which usually isn't the case
- In machine learning, querying generally focuses on instances with high uncertainty, e.g.:
  - Instances near the boundaries between classes
  - Instances in regions with few labels

Query Strategies I

- Which unlabelled instances will be most useful for learning?
- One simple strategy: query instances where the classifier is least confident of the classification

$$x = \underset{x}{argmax}(1 - P_\theta(\hat{y}|x))$$

where $$\hat{y} = \underset{y}{argmax}(P_\theta(y|x))$$

Query Strategies II

- Which unlabelled instances will be most useful for learning?
- Alternatively, margin sampling selects queries where the classifier is least able to distinguish between two categories, e.g.:

$$x = \underset{x}{argmin}(P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x))$$

- where $\hat{y}_1$ and $\hat{y}_2$ are the first- and second-most-probable labels for x

Query Strategies III

- Which unlabelled instances will be most useful for learning?
- Use entropy as an uncertainty measure to utilize all the possible class probabilities:

$$x = \underset{x}{argmax} - \sum_i P_\theta(\hat{y}_i|x) log_2 P_\theta(\hat{y}_i|x)$$

Query Strategies VI

- A more complex strategy, if you have multiple classifiers: **query by committee (QBC)**
- Train multiple classifiers on a labelled dataset, use each to predict on unlabelled data, and select instances with the highest disagreement between classifiers
- Assumes that all the classifiers learn something different, so can provide different information
- Disagreement can be measured by entropy

THE UNIVERSITY OF MELBOURNE

Active Learning Practicalities

Active learning is used increasingly widely, but must be handled with some care:

- empirically shown to be a robust strategy, but a theoretical justification has proven elusive
- querying is inherently biased towards a particular class set and learning approach(es), which may limit the general utility of the resulting dataset
- results to suggest that active learning is more highly reliant on “clean” labelling

Data Augmentation

Data Augmentation

- There are various ways to expand a labelled training dataset
- General: re-sampling methods
- Dataset-specific: add artificial variation to each instance, without changing ground truth label

Bootstrap sampling

- Bootstrap sampling: create new datasets by resampling existing data, with or without replacement
- Common in perceptron and neural network training (“mini-batch”,“batch size”), methods that involve stochastic gradient descent
- Each “batch” has a slightly different distribution of instances, forces model to use different features and not get stuck in local minima

Data Manipulation

- Another option: add a small amount of noise to each instance to create multiple variations:
    - Images: adjust brightness, flip left-right, shift image up /down / left / right, resize, rotate
    - Audio: adjust volume, shift in time, adjust frequencies
    - Text: synonym substitution
- These perturbations should not change the instance's label
- Generally, they should be the same kind of variations you expect in real-world data

Data Augmentation Pros and Cons

Advantages:

- More data nearly always improves learning
- Most learning algorithms have some robustness to noise (e.g., from machine-translation errors)

Disadvantages

- Biased training data
- May introduce features that dont exist in the real world
- May propagate errors
- Increases problems with interpretability and transparency

Summary

Summary

- What is semi-supervised learning?
- What is self-training, and how does it operate?
- What is active learning?
- What are the main sampling strategies in active learning?
- Outline a selection of query strategies in active learning.

References

- Burr Settles. Active learning literature survey. Technical report, Department of Computer Sciences, University of Wisconsin, Madison, 2010.
- Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison, 2005.
- Xiaojin Zhu. Tutorial on semi-supervised learning.