1. When do we use semi-supervised learning? What is self-training?

When:

Have a small number of labelled instances & a large number of unlabelled instances.

=) don't have enough data to train a reliable classifier

=) potentially leverage the labelled data to build a better classifier than a purely unsupervised method

(or the classifier with labelled data only)

Self-training:

1. Train the learner (model) on currently-labelled instances
2. Use the learner to predict the label of the unlabelled instances
3. Where the learner is very confident (e.g. high probability), add newly-labelled (predicted) to the training set.
4. Repeat (from step 1) until all instances are labelled or no instances can be labelled confidently.

2. What is the logic behind active learning, and what are some methods to choose instances for the oracle?

Allow the learner to choose a small number of instances to be labelled (by a human judge)

Idea:

① Many instances are easy to classify

② A small number of instances are difficult to classify

Select instances for query (2 cases)

① Measure uncertainty (uncertainty sampling)

② Select those raised highest disagreements (query by committe)

3. One of the strategies for Query sampling was query-by-committee (QBC). Using the equation below, which captures vote entropy, determine the instance that our active learner would select first.

$$x^*_{VE} = \underset{x}{\text{argmax}}(-\sum_{y_i} \frac{V(y_i)}{C} log_2 \frac{V(y_i)}{C})$$

Respectively y_i, $V(y_i)$, *and* C are the possible labels, the number of "votes" that a label receives from the classifiers, and the total number of classifiers.

classifier	Instance 1			Instance 2			Instance 3		
	y_1	y_2	y_3	y_1	y_2	y_3	y_1	y_2	y_3
C_1	0.2	0.7	0.1	0.2	0.7	0.1	0.6	0.1	0.3
C_2	0.1	0.3	0.6	0.2	0.6	0.2	0.21	0.21	0.58
C_3	0.8	0.1	0.1	0.05	0.9	0.05	0.75	0.01	0.24
C_4	0.3	0.5	0.2	0.1	0.8	0.1	0.1	0.28	0.62

↓ Actual prediction

classifier	Instance 1 Votes			Instance 2			Instance 3		
	y_1	y_2	y_3	y_1	y_2	y_3	y_1	y_2	y_3
C_1	0	1	0	0	1	0	1	0	0
C_2	0	0	1	0	1	0	0	0	1
C_3	1	0	0	0	1	0	1	0	0
C_4	0	1	0	0	1	0	0	0	1
	V(1)=1	V(2)=2	V(3)=1	V(1)=0	V(2)=4	V(3)=0	V(1)=2	V(2)=0	V(3)=2

Entropy : Instance 1 : $H = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{2}{4}\log_2\frac{2}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) = 1.5$

Instance 2 : $H = -\left(1\log_2 1\right) = 0$

Instance 3 : $H = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$

#1 highest vote entropy (disagreement) ⇒ Choose #1.

(Most difficult to classify ⇒ learn more about it by querying this instance)

4. Given the following univariate dataset, calculate a statistical model based on the assumption that your data is coming from a normal distribution. Determine whether the instance x=1.2 is anomalous or not if we use the boxplot test?

$$X = \{2, 2.5, 2.6, 3, 3.1, 3.2, 3.4, 3.7, 4, 4.1, 4.8\}$$

Anomaly detection: learn a model → fit the dataset & identify (statistical) the objs in low prob regions. (as anomalies)

Normal dist: $X \sim N(\mu, \sigma^2)$

$\hat{\mu} = \frac{1}{n}\sum_i x_i = \frac{1}{11}(2 + 2.5 + \cdots + 4.8) = 3.3$

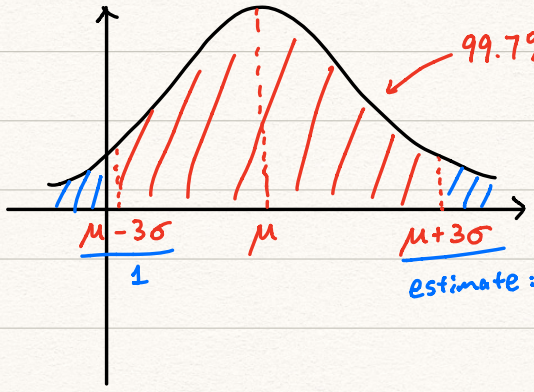
$\hat{\sigma}^2 = \frac{1}{n}\sum_i (x_i - \hat{\mu})^2 = \cdots = 0.59$

$\Rightarrow \hat{\sigma} = 0.77$

Boxplot (z-test):

If $X \sim N(\mu, \sigma^2)$: ~95% data lies in $\mu \pm 2\sigma$

~99.7% data lies in $\mu \pm 3\sigma$



$\approx 0.03\%$

estimate: $\hat{\mu} + 3\hat{\sigma} = 5.62$

$1 < 1.2 < 5.62 \Rightarrow$ within $3\sigma \Rightarrow$ not outlier
(x)

5. Given the following univariate dataset, determine the outlier score for instances (x=0.5) and (x=4) using the following strategies:

Dataset = {1,1.05, 1.1, 1.15, 1.2, 1.21, 1.3, 1.4, 1.45, 1.5, 4.55, 5.6, 6.8, 7.58, 8.6, 9.7, 10.3, 11.4, 12.3,13.5}

(a) Inverse Relative density using 2-NN (Manhattan distance)

Density based ⇒ outliers are in low density regions.

Relative density: "compactness" of each cluster of objects

$$\text{relative density}(x,k) = \frac{\text{density}(x,k)}{\frac{1}{k}\sum_{y \in N(x,k)} \text{density}(y,k)}$$ ← avg density of neighbors

$$\text{density}(x,k) = \left(\frac{1}{k}\sum_{y \in N(x,k)} \text{distance}(x,y)\right)^{-1}$$

inv of (average dist from k neighbors)

⇒ penalize instance if its nearest neighbors in high density region

① $x = 0.5$ (neighbours: 1, 1.05)

$$\text{density}(x=0.5, k=2) = \left(\frac{1}{2}(\underbrace{|0.5-1|}_{\text{dist}} + \underbrace{|0.5-1.05|}_{\text{dist}})\right)^{-1} = 1.9$$

neighbours: density$(x=1, k=2) = 13.3$ (neighbors: 1.05, 1.1)

density$(x=1.05, k=2) = 20$ (neighbors: 1, 1.1)

⇒ relative density $= \frac{1.9}{\frac{1}{2}(13.3+20)} = 0.11$

$IRD = \frac{1}{0.11} = 9.1$ (high outlier score)

② $x = 4$ (neighbours : 4.55, 5.6)

$$density(x = 4, k = 2) = \left(\frac{1}{2}(|4 - 4.55| + |4 - 5.6|)\right)^{-1} = 0.93$$

$$density(x = 4.55, k = 2) = \left(\frac{1}{2}(|4.55 - 5.6| + |4.55 - 6.8|)\right)^{-1} = 0.61$$

$$density(x = 5.6, k = 2) = \left(\frac{1}{2}(|5.6 - 4.55| + |5.6 - 6.8|)\right)^{-1} = 0.89$$

$$RD = \frac{0.93}{\frac{1}{2}(0.61 + 0.89)} = 1.24$$

$$IRD = \frac{1}{1.24} = 0.81 \quad (\text{low outlier score})$$

$x = 0.5$ penalise more : nearest cluster is very compact

$x = 4$: close to low density clusters.

(b) Distance to 2nd nearest neighbor (Manhattan distance)

Proximity based ⇒ anomaly if nearest neighbors are far away.

(⇒ it's far away from most objs in the dataset)

① $x = 0.5$

$dist(x = 0.5, x = 1.05) = 0.55$

② $x = 4$

$dist(x = 4, x = 5.6) = 1.6$

⇒ this can't capture the variability in cluster size

⇒ $x = 4$ has higher score than $x = 0.5$ (although it's close to a very sparse cluster)

6. In Assignment 1 we worked with the 'animals' dataset. Suggest a suitable method to detect anomalies among animal instances. Would you use a supervised, semi-supervised or unsupervised approach? Can you think of a way to make anomaly detection more reliable?

Animal dataset : 101 instances of animals (16 features)

↑ 7 groups (classes) : bird, reptile ...

Anomalies : instances that don't match the characteristic of any of the groups very well.

E.g. Platypus → anomaly (or outlier)

↳ some features of birds & amphibians & mammal

Assume
- we have a model trained to classify the animals into 7 groups
- platypus was not part of the training set

① Supervised :

Use it when we have access to label for "normal" data & "anomalies".

In our case, we've labels for animal categories but no labels for "normal" & "not normal" instances

⇒ Can't use supervised anomaly detection method.

② Semi-Supervised

Train a model on "normal" instances → use it to indirectly detect outliers

1. Assume that the labelled dataset includes only "normal" instances.
2. Train a supervised model on "normal" data (e.g. NB, LR, ...)

NB : outlier could be one having posterior probs with high entropy (evenly distributed among several classes)

3. Platypus features → expect the classifier to assign similar probs for bird & mammal & amphibian.

Problem: this approach can lead to high FP rate

(there could be high entropy distribution for noisy or just slightly atypical instances)

To be more confident about the outcome ⇒ ensemble (several different classifiers)

③ Unsupervised

Useful when we are not quite confident that the given dataset is good representative of "normal" instances (OR have no access to labels y)

⇒ treat dataset as unsupervised set

E.g. Cluster-based outlier detection (& ensemble method could lead to more reliable predictions)

k-means with different seeds or "k"...

- We can decide on a threshold → outlier or not

  e.g. outlier as top-n instances with furthest dist from any centroid (relative dist)

- Clustering → make binary decision (outlier or not) ⇒ voting

  Alternatively, clustering algorithm → return relative distance values directly

  ⇒ ensemble { Average dist; "max" dist; "min" dist }

  ⇒ threshold the value