

1.

✓ When 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:

model

1. Train the learner 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 committee)

3. One of the strategies for Query sampling was **query-by-committee (QBC)**, where a suite of classifiers is trained over a fixed training set, and the instance that results in the **highest disagreement** amongst the classifiers, is selected for querying. Using the equation below, which captures vote entropy, determine the **instance that our active learner would select first**.

$$x_{VE}^* = \underset{x}{\operatorname{argmax}} \left(- \sum_{y_i} \frac{V(y_i)}{C} \log_2 \frac{V(y_i)}{C} \right)$$

entropy
 (y_1, y_2, y_3)

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 Votes			Instance 3 Votes		
	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 : $H = - (1 \log_2 1) = 0$

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) \Rightarrow Choose #1.

(Most difficult to classify \Rightarrow 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?

$$X = \{2, 2.5, 2.6, 3, 3.1, 3.2, 3.4, 3.7, 4, 4.1, 4.8\} \quad (11)$$

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

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

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

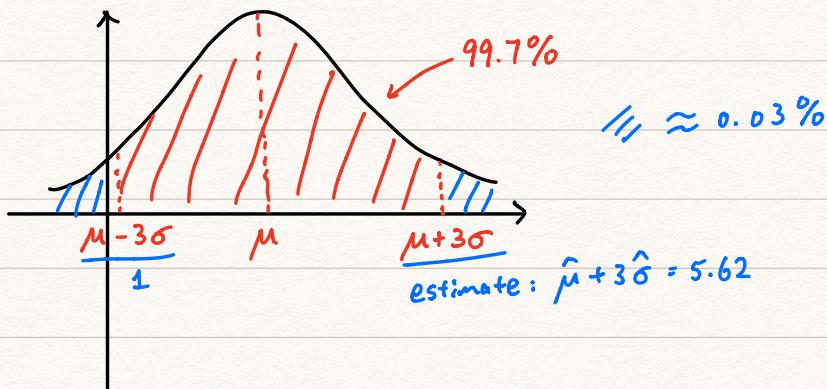
$$\hat{\sigma}^2 = \frac{1}{n} \sum_i (x_i - \hat{\mu})^2 = \dots = 0.59$$

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

Boxplot (z-test):

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

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



$1 < 1.2 < 5.62 \Rightarrow$ 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 \Rightarrow outliers are in low density regions.

Relative density: "compactness" of each cluster of objects

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

$$\text{density}(x, k) = \left(\underbrace{\frac{1}{k} \sum_{y \in N(x, k)} \text{distance}(x, y)}_{\text{inv of (average dist from } k \text{ neighbors)}} \right)^{-1}$$

\Rightarrow penalize instance if its nearest neighbors in high density region

① $x = 0.5$ (neighbours: 1, 1.05)

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

neighbours:

$$\text{density}(x = 1, k = 2) = \left(\frac{1}{2} (|1 - 1.05| + |1 - 1.1|) \right)^{-1} = 13.3$$

$$\text{density}(x = 1.05, k = 2) = \left(\frac{1}{2} (|1.05 - 1| + |1.1 - 1.05|) \right)^{-1} = 20$$

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

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

② $\kappa = 4$ (neighbours : 4.55, 5.6)

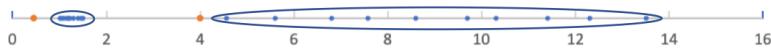
$$\text{density}(x = 4, k = 2) = \left(\frac{1}{2} (|4 - 4.55| + |4 - 5.6|) \right)^{-1} = 0.93$$

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

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

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

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



$\kappa = 0.5$ penalise more: nearest cluster is very compact

$\kappa = 4$: close to low density clusters.

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

Proximity based \Rightarrow anomaly if nearest neighbors are far away.

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

① $\kappa = 0.5$

$$\text{dist}(\kappa = 0.5, \kappa = 1.05) = 0.55$$

② $\kappa = 4$

$$\text{dist}(\kappa = 4, \kappa = 5.6) = 1.6$$

\Rightarrow this can't capture the variability in cluster size

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