# School of Computing and Information Systems
# The University of Melbourne
# COMP90049 Knowledge Technologies (Semester 1, 2017)
## Workshop exercises: Week 9

1. Revise the definition of **data mining**. Name the main modes of data mining, and give an example of a task for each one.

   - **Classification**, for example, choosing the best label which represents the topic of a text document
   - **Clustering**, for example, finding groups of network attackers which have something(s) in common
   - **Regression**, for example, predicting the numeric quantity representing the sea level rise in 2020
   - **Association Rule Mining**, for example, finding which purchases are strongly predictive of other purchases in a credit card dataset
   - **Sequential Discovery**, for example, finding weather patterns in a given city, or how a catalyst will change the rate of a chemical reaction
   - **Outlier Detection**, for example, finding one or more erroneous values in an otherwise regular collection of data

2. What is the **Naive Bayes** classifier? How does it classify data? What assumptions do we need to make about the data?

   - Naive Bayes is a probabilistic model, which classifies instances according to the class which has the greatest probability, given the instance as defined by its attribute values.
   - The assumption most central to the formulation of the Naive Bayes classifier that we have discussed is the **conditional independence assumption**, namely, that each attribute is independent of all of the other attributes, given the class under consideration. This assumption (while false) is necessary to make the problem tractable, where finding reliable estimates of the joint distribution of features requires more data than we are likely to have.
   - Another important assumption is in the way we find probabilities from the training data. This is most important for the **maximum likelihood estimation** we perform over the class priors (in contrast, we hedge our bets on the posterior probabilities of the terms using **smoothing**).
   - The fact that these assumptions are demonstrably untrue makes it seem like the classifier should not be effective at predicting the classes of unseen data. However, Naive Bayes is a pretty solid performer! It turns out that the methodology is robust enough to produce decent predictions, despite small (and predictable) discrepancies in the individual probabilities under consideration.
   - There are also a number of more minor assumptions:
     - We assume that the distribution of classes in the test set is (roughly) the same as the distribution of classes in the training set, and also that all of the classes in the test data are attested in the training data. This is often phrased as "the training and test instances were sampled from the same underlying distribution." (In fact, this is an assumption of most supervised machine learning algorithms.)
     - We typically require some kind of assumption for our smoothing method, depending on which smoothing method we actually use.
     - We need to make some assumptions about the distribution of continuous attributes (typically Gaussian) if they exist in our data set.

3. For the following dataset:

   Classify the test instances according to the method of Naive Bayes, as discussed in this subject.

| *apple* | *ibm* | *lemon* | *sun* | CLASS |
|---|---|---|---|---|
| TRAINING INSTANCES | | | | |
| Y | N | Y | Y | FRUIT |
| Y | N | Y | Y | FRUIT |
| Y | Y | N | N | COMPUTER |
| Y | Y | Y | Y | COMPUTER |
| TEST INSTANCES | | | | |
| Y | N | Y | Y | ? |
| Y | N | Y | N | ? |

- Naive Bayes selects a class $c$ from a set of classes $C$ for a test instance $T = \langle t_1, t_2, \ldots, t_n \rangle$ using a set of training instances $\mathcal{D}$ according to:

$$c = \text{argmax}_{c_j \in C} P(c_j) P(T \mid c_j) \tag{1}$$

- We will expand $P(T \mid c_j)$ based on our conditional independence assumption (according the attribute values $t_i$ seen in our test instance):

$$P(T \mid c_j) = \prod P(t_i \mid c_j) \tag{2}$$

- We can calculate the prior probabilities of the classes straight from the training data, using maximum likelihood estimation. For FRUIT, 2 of the 4 instances are FRUIT:

$$P(f) = \frac{2}{4} = \frac{1}{2}$$

- We find the conditional probabilities $P(t_i \mid c_j)$ based on maximum likelihood estimation from the data set, smoothing by replacing 0 values with a small, positive, non-zero value $\epsilon$. (In the real world, we would probably use Laplacian smoothing, or some other more serious smoothing method.)
- To do this, we will observe what proportion of the instances for the given class contain the term that we're looking for. For example, for $P(a = \text{T} \mid f)$: of the two FRUIT instances, both of them contain *apple*.

$$\begin{aligned}
P(a = \text{T} \mid f) &= \frac{2}{2} = 1 \\
P(i = \text{T} \mid f) &= \frac{0}{2} = 0 \\
P(l = \text{T} \mid f) &= \frac{2}{2} = 1 \\
P(s = \text{T} \mid f) &= \frac{2}{2} = 1 \\
P(a = \text{T} \mid c) &= \frac{2}{2} = 1 \\
P(i = \text{T} \mid c) &= \frac{2}{2} = 1 \\
P(l = \text{T} \mid c) &= \frac{1}{2} \\
P(s = \text{T} \mid c) &= \frac{1}{2}
\end{aligned}$$

- When we substitute these values into (2) above, we will replace 0 values with $\epsilon$.
- We will also need the conjugate probabilities, for example $P(t = \text{F} \mid c)$. To find these, we will observe that $P(a \mid b) + P(\bar{a} \mid b) = 1$. Consequently, $P(t = \text{F} \mid c) = 1 - P(t = \text{T} \mid c)$.

- Going all the way back to (1), and substituting our simplification from (2), we will now consider the values for FRUIT and COMPUTER (they aren't really probabilities any more, but it isn't important now).

$$\begin{aligned}
\text{FRUIT} \quad : \quad & P(a = \mathrm{T} \mid f)P(i = \mathrm{F} \mid f)P(l = \mathrm{T} \mid f)P(s = \mathrm{T} \mid f)P(f) \\
= \quad & P(a = \mathrm{T} \mid f)(1 - P(i = \mathrm{T} \mid f))P(l = \mathrm{T} \mid f)P(s = \mathrm{T} \mid f)P(f) \\
= \quad & 1 \times (1 - 0) \times 1 \times 1 \times \frac{1}{2} \\
= \quad & \frac{1}{2} \\
\text{COMP} \quad : \quad & P(a = \mathrm{T} \mid c)P(i = \mathrm{F} \mid c)P(l = \mathrm{T} \mid c)P(s = \mathrm{T} \mid c)P(c) \\
= \quad & P(a = \mathrm{T} \mid c)(1 - P(i = \mathrm{T} \mid c))P(l = \mathrm{T} \mid c)P(s = \mathrm{T} \mid c)P(c) \\
= \quad & 1 \times (1 - 1) \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \\
\approx \quad & 1 \times \epsilon \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \\
\approx \quad & \frac{\epsilon}{8}
\end{aligned}$$

- For this test instance, $T_1$, we choose FRUIT, because $\frac{1}{2} > \frac{\epsilon}{8}$ (because $\epsilon$ is small).
- For $T_2$, the calculations are similar, except for the fact that $s = \mathrm{F}$:

$$\begin{aligned}
\text{FRUIT} \quad : \quad & P(a = \mathrm{T} \mid f)P(i = \mathrm{F} \mid f)P(l = \mathrm{T} \mid f)P(s = \mathrm{F} \mid f)P(f) \\
= \quad & P(a = \mathrm{T} \mid f)(1 - P(i = \mathrm{T} \mid f))P(l = \mathrm{T} \mid f)(1 - P(s = \mathrm{T} \mid f))P(f) \\
= \quad & 1 \times (1 - 0) \times 1 \times (1 - 1) \times \frac{1}{2} \\
\approx \quad & \frac{\epsilon}{2} \\
\text{COMP} \quad : \quad & P(a = \mathrm{T} \mid c)P(i = \mathrm{F} \mid c)P(l = \mathrm{T} \mid c)P(s = \mathrm{F} \mid c)P(c) \\
= \quad & P(a = \mathrm{T} \mid c)(1 - P(i = \mathrm{T} \mid c))P(l = \mathrm{T} \mid c)(1 - P(s = \mathrm{T} \mid c))P(c) \\
= \quad & 1 \times (1 - 1) \times \frac{1}{2} \times (1 - \frac{1}{2}) \times \frac{1}{2} \\
\approx \quad & \frac{\epsilon}{8}
\end{aligned}$$

- And again, this is classified as FRUIT.