Unsupervised Learning

Dr Dimitris C. Dracopoulos
email: d.dracopoulos@westminster.ac.uk

Course web page:
http://users.wmin.ac.uk/~dracopd/DOCUM/courses/2ait608/ait608.html

Unsupervised Learning vs Other Types of Learning

- *Supervised learning*: Given input vectors $x_i, i = 1, 2, \ldots N$ and desired (correct) outputs $y_i, i = 1, 2, \ldots N$ build an appropriate model.
- *Unsupervised learning*: Given input vectors $x_i, i = 1, 2, \ldots N$ build an appropriate model for some specific task, e.g. clustering.
- *Reinforcement learning*: Given input vectors $x_i, i = 1, 2, \ldots N$ and a measure U_i of how good (or bad) is each input, build an appropriate model.

Clustering

Partition some n-dimensional points in space.

Application Example:

Assume a 24-bit image. Each pixel has up to 16 million colours. Given a colour screen with 8bits/pixel (256 colours per pixel), find the best 256 colours among all 16 million colours such that if the image uses these 256 colours it looks as close as possible to the original image.

- *Colour quantisation* problem mapping a high resolution to a lower resolution.
- General problem is *vector quantisation*: mapping from a continuous space to a discrete space.

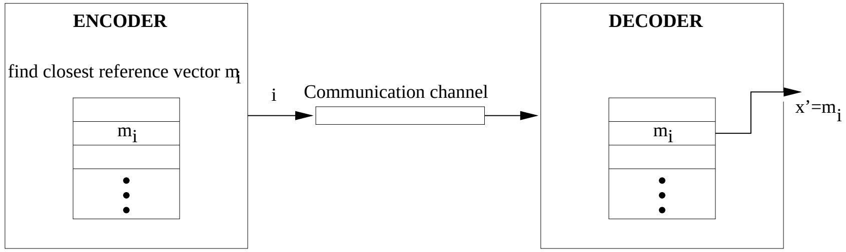
Possible solutions to the colour quantisation problem

- Quantise uniformly: divide 16 million colours by 256. Waste of colourmap as some colours do not exist in image. Does not take advantage of the fact that some colours appear more frequently in the image.
- The distribution of the colour mapping should be close to the original density.

The problem is to find $k = 256$ reference vectors $\mathbf{m}_j, j = 1, \ldots, k$.

Vector quantisation vs encoding/decoding

The vector (colour) quantisation problem can be seen as an encoding/decoding process.



In the colour quantisation problem maximum compression achieved is 3: (24bits/8bits). The reference vectors $\mathbf{m}_i$ have to be transmitted to the decoder as well.

Assuming an image contains N pixels, instead of transmitting $24N$ bits we only need to transmit $N \log_2 k + 24K$ bits ($\log_2 k$ bits per pixel are required to transmit the identity of a pixel of the reference

vector $\mathbf{m}_i$ and $24K$ bits are needed to transmit all the reference vectors).

k-means Clustering

Use nearest (most similar) reference, for example Euclidean distance between two n-dimensional points:

$$||\mathbf{x}^t - \mathbf{m}_i|| = \min_j ||\mathbf{x}^t - \mathbf{m}_j|| \tag{1}$$

where $t = 1, \ldots, N$.

Total reconstruction error (e.g. decoding problem):

$$E(\{\mathbf{m}_i\}_{i=1}^k|\mathcal{X}) = \sum_t \sum_i b_i^t ||\mathbf{x}^t - \mathbf{m}_i||^2 \tag{2}$$

where

$$b_i^t = \left\{ \begin{array}{ll} 1 & \text{if } ||\mathbf{x}^t - \mathbf{m}_i|| = \min_j ||\mathbf{x}^t - \mathbf{m}_j|| \\ 0 & \text{otherwise} \end{array} \right\} \tag{3}$$

The k-means clustering algorithm is an iterative algorithm. At each iteration, the estimated labels are calculated, by minimising the constructive error:

$$\frac{\partial E}{\partial \mathbf{m}_i} = 0 \Rightarrow \mathbf{m}_i = \frac{\sum_t b_i^t \mathbf{x}^t}{\sum_t b_i^t} \tag{4}$$

The iterative steps continue until $\mathbf{m}_i$ stabilise (does not change any more).

Pseudocode for k-means clustering

Initialise $\mathbf{m}_i, i = 1, \ldots, k$ for example randomly.
Repeat
    for all $\mathbf{x}^t \in \mathcal{X}$

$$b_i^t = \left\{ \begin{array}{ll} 1 & \text{if } ||\mathbf{x}^t - \mathbf{m}_i|| = \min_j ||\mathbf{x}^t - \mathbf{m}_j|| \\ 0 & \text{otherwise} \end{array} \right\}$$

    for all $\mathbf{m}_i, i = 1, \ldots, k$

$$\mathbf{m}_i = \frac{\sum_t b_i^t \mathbf{x}^t}{\sum_t b_i^t}$$

until $\mathbf{m}_i$ converge

Figure 1: Pseudocode of the k-means clustering algorithm

Example:

Suppose we have four types of medicines and each medicine have two attributes as shown in table below. Our goal is to group these medicines into $K = 2$ groups of medicines based on the two features (pH and weight index).

Object	weight index	pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

In a 2-dimensional space, each medicine represents a point as in Figure 2:

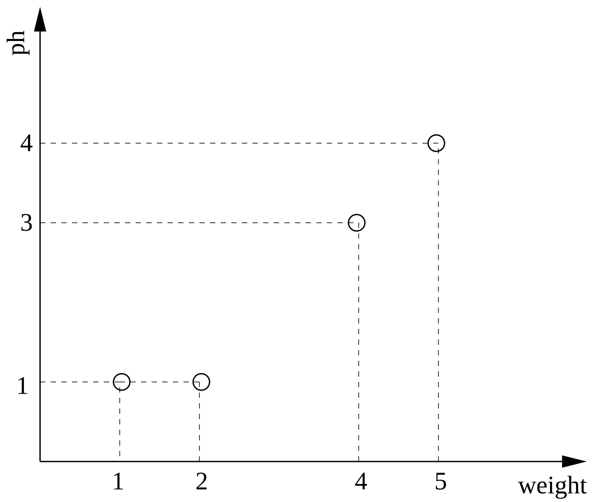


Figure 2: Categorising medicines based on k-means clustering.

Iteration 1:

Randomly select $\mathbf{m}_i$ to correspond to `MedicineA` and `MedicineB` respectively. Then $\mathbf{m}_1 = (1, 1)$ and $\mathbf{m}_2 = (2, 1)$

Calculating the Euclidean distances from all the medicines (points) to $\mathbf{m}_i$ we find:

	distance_to_m1	distance_to_m2	categorise
Medicine A	0	1	group1
Medicine B	1	0	group2
Medicine C	3.61	2.83	group2
Medicine D	5	4.24	group2

Calculating new $\mathbf{m}_i$: $\mathbf{m}_1 = (1, 1)$ and $\mathbf{m}_2 = (\frac{11}{3}, \frac{8}{3})$

Iteration 2:

Calculating the Euclidean distances from all the medicines (points) to $\mathbf{m}_i$ we find:

	distance_to_m1	distance_to_m2	categorise
Medicine A	0	3.14	group1
Medicine B	1	2.36	group1
Medicine C	3.61	0.47	group2
Medicine D	5	1.89	group2

Calculating new $\mathbf{m}_i$: $\mathbf{m}_1 = (\frac{3}{2}, 1)$ and $\mathbf{m}_2 = (\frac{9}{2}, \frac{7}{2})$

Iteration 3:

Calculating the Euclidean distances from all the medicines (points) to $\mathbf{m}_i$ we find:

	distance_to_m1	distance_to_m2	categorise
Medicine A	0.5	4.30	group1
Medicine B	0.5	3.54	group1
Medicine C	3.2	0.71	group2
Medicine D	4.61	0.71	group2

Calculating new $\mathbf{m}_i$: $\mathbf{m}_1 = (\frac{3}{2}, 1)$ and $\mathbf{m}_2 = (\frac{9}{2}, \frac{7}{2})$.

Since $\mathbf{m}_1$, $\mathbf{m}_2$ converged the algorithm terminates and categorises the medicines as in the last table above.

Fuzzy k-means Clustering

A variation of k-means clustering based on fuzzy logic. Each sample $\mathbf{x}^j$ has some "fuzzy" membership in a cluster. The "memberships" are equivalent to probabilities $P(\omega_i|\mathbf{x}_j)$.

Minimise the following heuristic cost function:

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} P(\omega_i|\mathbf{x}_j)^b ||\mathbf{x}_j - \mathbf{m}_i||^2 \qquad (5)$$

where c the number of clusters and n the total number of $\mathbf{x}_j$ vectors.

b is a free parameter which defines the blending of different clusters. For $b > 1$ each pattern $\mathbf{x}_j$ belongs to multiple clusters.

The probabilities of cluster membership for each point are normalised:

$$\sum_{i=1}^{c} P(\omega_i|\mathbf{x}_j) = 1 \qquad j = 1, \ldots, n \qquad (6)$$

To minimise cost:

$$\frac{\partial J}{\partial \mathbf{m}_i} = 0 \Rightarrow \mathbf{m}_i = \frac{\sum_{j=1}^{n} (P(\omega_i|\mathbf{x}_j))^b \mathbf{x}_j}{\sum_{j=1}^{n} (P(\omega_i|\mathbf{x}_j))^b} \qquad (7)$$

and

$$\frac{\partial J}{\partial \mathbf{P}_j} = 0 \Rightarrow P(\omega_i|\mathbf{x}_j) = \frac{(1/d_{ij})^{1/(b-1)}}{\sum_{r=1}^{c} (1/d_{rj})^{1/(b-1)}} \qquad (8)$$

where $b > 1$ and $d_{ij} = ||\mathbf{x}_j - \mathbf{m}_i||^2$.

Notice that the k-means clustering algorithm is a special case of the fuzzy k-means clustering algorithm when the memberships for all points:

$$P(\omega_i|\mathbf{x}_j)) = \left\{ \begin{array}{ll} 1 & \text{if } ||\mathbf{x}_j - \mathbf{m}_i|| = \min_i ||\mathbf{x}_j - \mathbf{m}_i|| \\ 0 & \text{otherwise} \end{array} \right\} \quad (9)$$

Pseudocode for fuzzy k-means clustering

```
Initialise b, m_i, P(ω_i|x_j)),  i = 1,...,c,  j = 1,...,n
Normalise P(ω_i|x_j)) by Σ_{i=1}^{c} P(ω_i|x_j) = 1    j = 1,...,n
Repeat
        calculate m_i by equation (7);
        calculate P(ω_i|x_j)) by equation (8);
until changes in both m_i and P(ω_i|x_j)) less than ε

Return m_1, m_2, ..., m_c;
```

Figure 3: Pseudocode of the fuzzy k-means clustering algorithm