

Clustering

David Barber

Learning Objective

Lectures

- Understanding of Unsupervised Learning and Clustering¹.
 - Familiarity and ability to apply K-means to cluster data. Understanding of the limitations of K-means.
 - Understanding of Gaussian Mixture Models
 - Understanding of more general mixture models, and how to deal with categorical data.
 - How to deal with missing data in clustering.
-

Practicals

- Clustering product data
- Clustering of questionnaire data, including missing data.

¹Some material inspired by Piyush Rai, CS Utah

Clustering

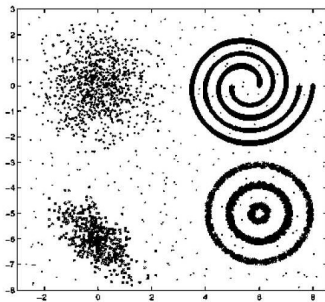
What is it?

- Data is unlabelled – a form of unsupervised learning.
 - Want to find compact descriptions of data.
-

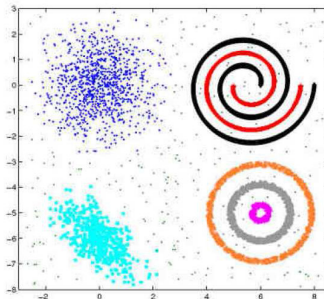
Applications

- My clients can be considered to belong to 5 different groups, with a typical client from each group looking like this.
- Clustering webpages based on their content.
- Clustering web-search results.
- Clustering people in social networks based on user properties/preferences
- More generally, we want to reduce the apparent complexity to gain insight into the structure of the data.

Clustering



(a) Input data





















(b) Desired clustering

- Want clusters such that datapoints within the same cluster have high similarity²
- Want clusters such that datapoints from different clusters have low similarity.
- Different definitions of 'dissimilarity' - most common is the squared distance between points.

²Data Clustering: 50 Years Beyond K-Means, A.K. Jain (2008)

Basket-Item Matrix

		Basket					
		1	2	3	4	5	6
Item	eggs						
	bacon						
	milk						
	beer						
	tea						

- Data may come in list form (a list of items in each basket).
- Often useful to represent this in matrix form.
- Typically a very sparse matrix.

A priori algorithm for frequent item sets

		Basket					
		1	2	3	4	5	6
Item	eggs	●		●	●		
	bacon		●	●			
	milk	●	●	●		●	●
	beer	●	●	●		●	
	tea	●		●	●		●

- Start by finding frequent single items, then pairs of frequent items, triplets, *etc.*
- For a threshold of 3, {Eggs}, {Milk}, {Beer}, {Tea}, {Eggs,Tea}, {Milk, Beer}, {Milk,Tea} are frequent.
- This is a classical topic in Data Mining (finding association rules).
- In 'clustering' we want something a bit different – we want to find baskets that look similar and group them together typically into a single 'cluster basket'. Each basket will belong to only one 'cluster basket'.

Clustering

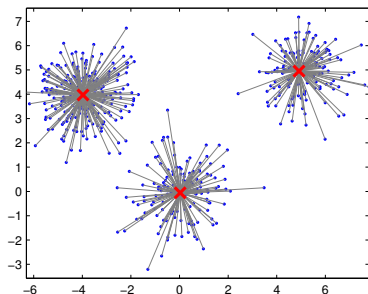
Example problems

- We have a basket-item matrix in which each basket contains a list of items purchased on a visit. Are there typical groupings of baskets?
 - We have a questionnaire and each customer answers each of the 100 questions in the questionnaire. Are there certain kinds of customers that we can find?
-

Solutions

- K-means (classical approach): Fast to run but not always appropriate.
- Gaussian Mixture Models: Popular and most appropriate when the data in each cluster is Gaussian distributed.
- More general mixture models: can naturally handle discrete observations and missing data.
- Mixed Membership style models: useful when it is not appropriate to describe a customer in terms of membership of a single group (customer might be a little bit this, a little bit that).

K-means



- We have a dataset $\mathbf{x}^1, \dots, \mathbf{x}^N$ of data vectors (N customers).
- Want to segment these customers by assigning each datapoint to one of K groups (or 'centres'), $\mathbf{m}^1, \dots, \mathbf{m}^K$.
- In this example $K = 3$; blue dots are datapoints, red crosses are centres.

- We will assign each datapoint to a cluster, $\mathbf{x}^n \rightarrow c(n)$ where $c(n) \in \{1, \dots, K\}$ is the cluster index of datapoint number n .
- Want to find the assignments and centres that minimise the squared loss

$$\sum_{n=1}^N \left(\mathbf{x}^n - \mathbf{m}^{c(n)} \right)^2$$

- The squared loss is the sum of the squared lengths of the grey lines.

K-means algorithm

- 1: Initialise the centres $\mathbf{m}_i, i = 1, \dots, K$.
- 2: **while** not converged **do**
- 3: For each centre i , find all the \mathbf{x}^n for which i is the nearest centre.
- 4: Call this set of points \mathcal{N}_i . Let N_i be the number of datapoints in set \mathcal{N}_i .
- 5: Update the centre by taking the mean of those datapoints assigned to this centre:

$$\mathbf{m}_i^{new} = \frac{1}{N_i} \sum_{n \in \mathcal{N}_i} \mathbf{x}^n$$

- 6: **end while**

Initialisation

- We can get different clusterings depending on the initialisation of the centres.
- Typically we run the algorithm several times with different initialisations.
- The 'best' clustering is that which has the lowest squared loss.

K-means in action

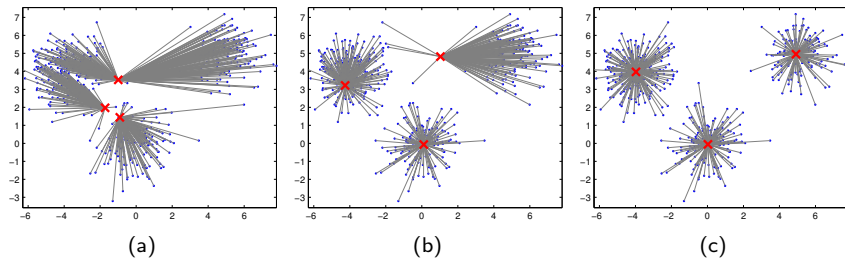
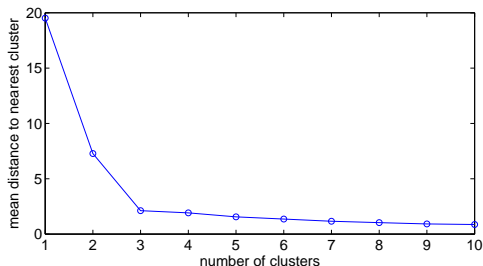


Figure : 550 datapoints clustered using K-means with 3 components. The means are given by the red crosses. **(a):** First iteration. **(b):** Second iteration. **(c):** Third iteration.

How we order the cluster centres is irrelevant – we might call the top right cluster number '2', but we could equally call it number '3'.

`demoKmeansMNIST.m`

Determining the number of clusters



- As we increase the number of clusters, the mean distance to the nearest cluster centre will decrease.
- Look for a 'knee' (or 'elbow') – there is one here at $K = 3$.
- This is the point in which the distance stops decreasing significantly.

K-means limitations

Hard assignment

- A datapoint is assigned to only a single cluster.
 - It might be preferable to make a soft assignment – probably belongs to cluster 1, but could belong to cluster 3 with a certain probability.
 - Mixture models can address this.
-

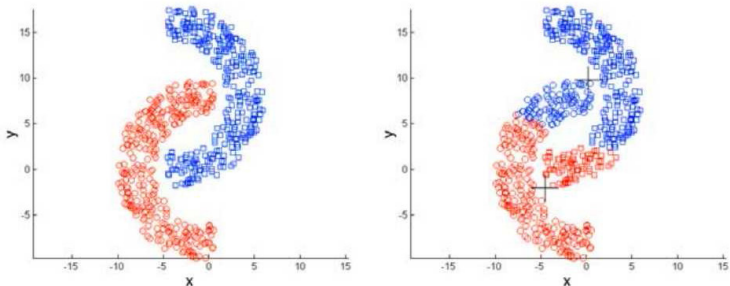
Outlier sensitivity

- If there is an outlier (datapoint far from the rest) this can throw off the K-means algorithm.
 - The K-medians algorithm can be less sensitive to outliers.
-

Shape/density issues

- Works best when the clusters are roughly spherical and of equal number of points.
- Can use spectral clustering in case the data is not spherical.

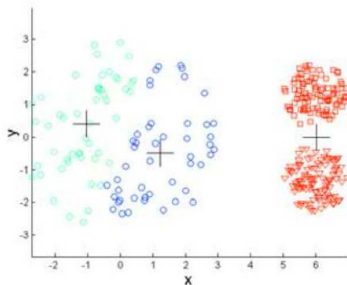
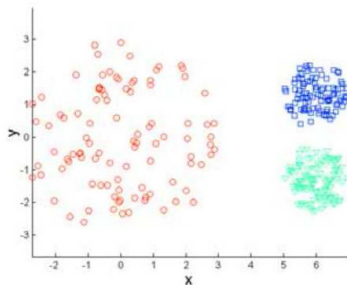
K-means limitations: non spherical data



- Left: clustering we would like. Right: clustering K-means produces³.
- Problem is that the clusters are not spherical and they are too close.
- Can address using spectral clustering (see later).

³Images from Christof Monz (Queen Mary, Univ. of London)

K-means limitations: non equal density



- Left: clustering we would like. Right: clustering K-means produces.
- Hard to address this with K-means.

K-means: product purchase clustering

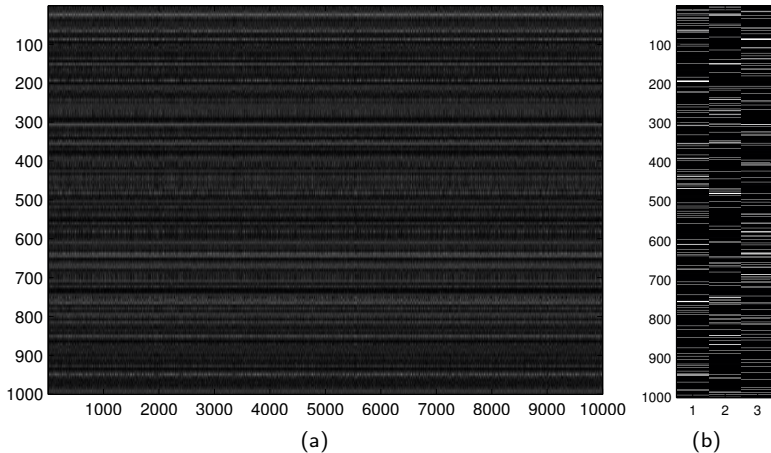


Figure : (a): 10,000 datapoints of 1000 dimensions (90% of entries are zero). (b): 3 product purchase clusters.

Practical Issues

Computational cost

- K-means is a fast method.
 - However, if each datapoint is very high dimensional, finding the nearest centre can be expensive.
 - There are speed-up techniques available (see fastNN material).
-

Representation Sensitivity

- Let's say that each data-vector contains two attributes x_1 and x_2 . x_1 represents the temperature in centigrade, and x_2 the distance.
- If we represent the distance in millimetres, then the Euclidean distance will be dominated by the difference in distance.
- If we represent the distance in kilometers, the distance will be most likely dominated by the difference in temperature.
- Rescaling (see data intro) is one way around this.

Practical Issues

Categorical Data

- If the data are for example questionnaire responses, 'a','b','c', how are we to numerically encode these? We could use 1,2,3, but this would bias the clusters found since 1 is closer to 2 than to 3.

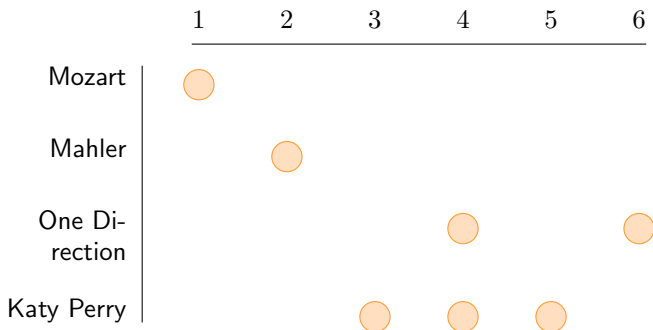
- 1-of- M encoding: $a \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $b \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, $c \rightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$. So, for example

a data vector with two attributes might be replaced by $\begin{pmatrix} a \\ 0.2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0.2 \end{pmatrix}$

Missing Data

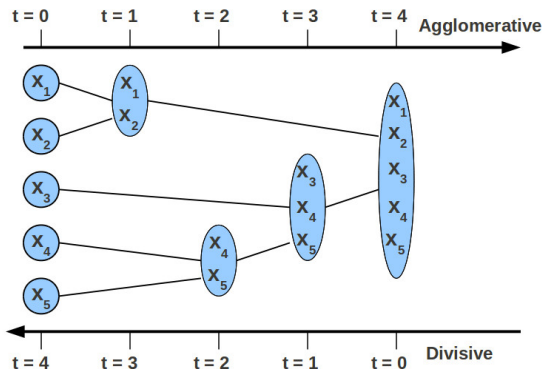
- How can we deal with missing data? Encoding missing data with a '0' will bias the clusters found.
- There is no simple and justifiable way to deal with missing data in K-means.

Hierarchical K-means



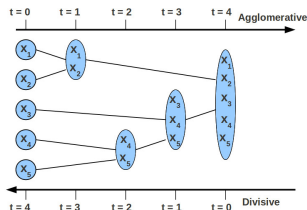
- Here there are two groups – classical music buyers and ‘pop’ music buyers.
- Classical music data can be further split into a Mozart and Mahler group.
- Pop music data splits into One Direction and Katy Perry groups.
- Can do this Top Down (clustering all data into a small number of groups, and then clustering the data in each group) or Bottom Up (recursively merging datapoints or clusters that are close together).

Hierarchical K-means



- Hierarchical clustering forms a tree in which the children are contained within their parent cluster.
- This is a binary tree example, but could have say ternary tree, *etc.*

Hierarchical K-means



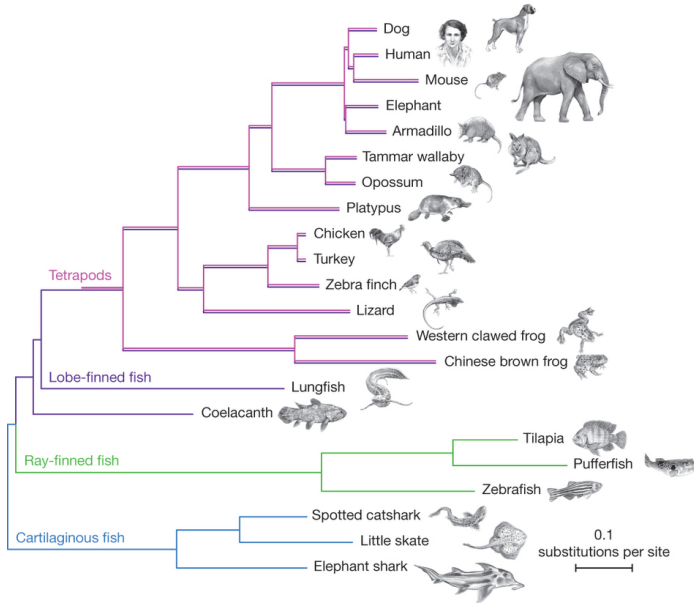
Agglomerative (bottom-up) Clustering

- Start with each example in its own singleton cluster
- At each time-step, greedily merge 2 most similar clusters

Divisive (top-down) Clustering

- Start with all examples in the same cluster
- At each time-step, remove the 'outsiders' from the least cohesive cluster

Hierarchical Clustering using gene similarity⁴



Spectral Clustering

- Define a symmetric matrix of the similarity of datapoints, for example

$$A_{i,j} = e^{-\lambda ||\mathbf{x}^i - \mathbf{x}^j||^2}$$

for some chosen λ .

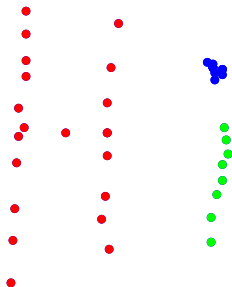
- Define Markov chain transition matrix with elements

$$p(i|j) = \frac{A_{i,j}}{\sum_i A_{i,j}}$$

- This defines a model which specifies the probability that we jump to datapoint i given that we are currently at datapoint j .
- The equilibrium distribution of p is the probability that we visit each datapoint in the long run by transitioning randomly according to p .
- Key insight is that the closely connected points will form a single high probability cluster.

Spectral Clustering

- The equilibrium distribution is given by finding the eigenvector e of p with largest eigenvalue.
- Datapoints i with e_i larger than some threshold are placed then in a single cluster and the remaining datapoints in a separate cluster. Alternatively, compute the largest k eigenvectors and cluster these using k-means.
- I prefer to take each datapoint and find the equilibrium distribution, starting from that datapoint, and then cluster these distributions using k-means.

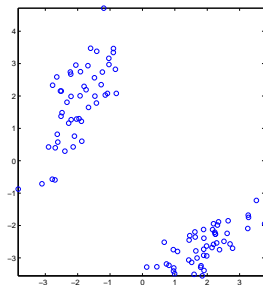


Spectral Clustering

- Spectral Clustering can be very sensitive to the choice of similarity measure ●
- If we apply K-means to the equilibrium distributions, we still have a potentially difficult optimisation problem, requiring multiple runs to get the best solution ●
- We have to set parameters such as λ (in the A matrix) by hand ●
- Spectral Clustering is a very powerful method that can deal with data distributions where standard k-means does not produce a satisfactory result ●

`demoSpectralClustering.m`

Clustering



Samples from a Gaussian mixture model $1/2\mathcal{N}(\mathbf{x}|\mathbf{m}_1, \mathbf{C}_1) + 1/2\mathcal{N}(\mathbf{x}|\mathbf{m}_2, \mathbf{C}_2)$ with means $\mathbf{m}_1, \mathbf{m}_2$ and covariances $\mathbf{C}_1, \mathbf{C}_2$.

- Mixture models have a natural application in clustering data, where h indexes the cluster.
- This interpretation can be gained from considering how to generate a sample datapoint v from the model.
- First we sample a cluster h from $p(h)$, and then draw a visible state v from $p(v|h)$.
- Mixture Models are important since they enable the clustering of diverse kinds of data and can readily deal with missing data and hierarchical constraints.

Clustering as Data generation: an example

- We have four products: apples, broccoli, crisps, doughnut.
- Represent the products a customer bought on a visit by a 4-dimensional vector \mathbf{x} ($x_i = 0$ zero represents non-purchase):

$x_1 = 1$ represents that the customer bought apples

$x_2 = 1$ represents that the customer bought broccoli

$x_3 = 1$ represents that the customer bought crisps

$x_4 = 1$ represents that the customer bought doughnuts

Cluster 1

- These people tend to buy health food products and avoid 'junk' food.
- Let's assume that they will buy any of the products independently.

Cluster 2

- These people tend to buy 'junk' food products and avoid 'healthy' food.
- Let's assume that they will buy any of the products independently.

Clustering as data generation: an example

- Let's say that 25% of customers are in the healthy group and 75% in the junk food purchasing group.
- Let's try to generate a set of data. To do this we also need to specify the probabilities that each group will buy certain products:

Cluster 1:

$$p(x_1 = 1 | \text{cluster 1}) = 0.5, \quad p(x_1 = 0 | \text{cluster 1}) = 0.5$$

$$p(x_2 = 1 | \text{cluster 1}) = 0.7, \quad p(x_2 = 0 | \text{cluster 1}) = 0.3$$

$$p(x_3 = 1 | \text{cluster 1}) = 0.1, \quad p(x_3 = 0 | \text{cluster 1}) = 0.9$$

$$p(x_4 = 1 | \text{cluster 1}) = 0.1, \quad p(x_4 = 0 | \text{cluster 1}) = 0.9$$

Cluster 2:

$$p(x_1 = 1 | \text{cluster 2}) = 0.1, \quad p(x_1 = 0 | \text{cluster 2}) = 0.9$$

$$p(x_2 = 1 | \text{cluster 2}) = 0.1, \quad p(x_2 = 0 | \text{cluster 2}) = 0.9$$

$$p(x_3 = 1 | \text{cluster 2}) = 0.5, \quad p(x_3 = 0 | \text{cluster 2}) = 0.5$$

$$p(x_4 = 1 | \text{cluster 2}) = 0.4, \quad p(x_4 = 0 | \text{cluster 2}) = 0.6$$

Clustering as data generation: an example

- For each customer we first decide if they are going to be in the healthy group or the junk food group.
- We do this by drawing a cluster index $h \in \{1, 2\}$ from the distribution $p(h = 1) = 0.25$, $p(h = 2) = 0.75$.
- Given the cluster we now draw a set of product purchases, independently for each product according to the given probabilities.
- For example, we might have drawn $h = 1$, and then drawn a purchase vector

$$\mathbf{x} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

- We repeat this process for 10 people and have the data matrix:

$$\mathbf{X} = \begin{pmatrix} 0, 1, 0, 1, 1, 0, 0, 0, 0, 1 \\ 0, 0, 1, 1, 1, 0, 0, 0, 0, 0 \\ 0, 0, 1, 0, 1, 0, 0, 1, 1, 1 \\ 1, 0, 0, 0, 1, 1, 1, 0, 1, 0 \end{pmatrix}$$

Clustering as data generation: an example

- Given the data matrix:

$$\mathbf{X} = \begin{pmatrix} 0, 1, 0, 1, 1, 0, 0, 0, 0, 1 \\ 0, 0, 1, 1, 1, 0, 0, 0, 0, 0 \\ 0, 0, 1, 0, 1, 0, 0, 1, 1, 1 \\ 1, 0, 0, 0, 1, 1, 1, 0, 1, 0 \end{pmatrix}$$

- Can we figure out which are the healthy customers and the junk food customers?

Bayes Rule

$$p(h = 1|\mathbf{x}) = \frac{p(\mathbf{x}|h = 1)p(h = 1)}{p(\mathbf{x}|h = 1)p(h = 1) + p(\mathbf{x}|h = 2)p(h = 2)}$$

$$p(\mathbf{x}|h = 1) = \prod_{i=1}^4 p(x_i|h = 1)$$

Clustering as data generation: an example

Using Bayes rule to calculate cluster assignment

$$\begin{aligned} p(h = 1|(0, 0, 0, 1)) &= \frac{0.5 \times 0.3 \times 0.9 \times 0.1 \times 0.25}{0.5 \times 0.3 \times 0.9 \times 0.1 \times 0.25 + 0.9 \times 0.9 \times 0.5 \times 0.4 \times 0.75} \\ &= 0.027 \end{aligned}$$

- Hence $p(h = 2|(0, 0, 0, 1)) = 1 - 0.027 = 0.973$.
- This person is very likely to be from the 'junk food' group.
- We do this for each person in the data to get a 'soft' assignment to the clusters.

Exercise:

What is the probability that $\mathbf{x} = (1, 0, 1, 0)$ belongs to the healthy group?

Clustering as data generation: an example

Joint distribution

$$p(\mathbf{x}, h) = p(\mathbf{x}|h)p(h) = p(h) \prod_i p(x_i|h)$$

Conditional

Using the joint distribution we can compute the conditional

$$p(h|\mathbf{x}) \propto p(\mathbf{x}, h)$$

Marginal

$$p(\mathbf{x}) = \sum_h p(\mathbf{x}, h) = \sum_h p(h) \prod_i p(x_i|h)$$

Likelihood

Since each customer is generated independently in the same way, the distribution over all customers is

$$p(\mathbf{x}^1, \dots, \mathbf{x}^{10}) = \prod_{n=1}^{10} p(\mathbf{x}^n)$$

Clustering as data generation: an example

- In this example, we assumed that we know the probabilities $p(h)$ and $p(x_i|h)$.
- However, in general we are interested in the situation that we have an observed dataset \mathbf{X} and we want to learn these parameters.
- We can do this by setting the parameters to those for which the data we observe is the most likely to have been generated by the model.
- This is called the maximum likelihood approach.

Training philosophy

- Define the model $p(\mathbf{x}, h|\theta)$.
- For the given dataset \mathbf{X} , search over all parameters of the model θ such that

$$\theta_{opt} = \arg \max_{\theta} p(\mathbf{X}|\theta) = \arg \max_{\theta} \prod_n p(\mathbf{x}^n|\theta)$$

- Given the optimal parameters, form the soft-clustering

$$p(h^n|\mathbf{x}^n)$$

Mixture Models

- Mixture models are a powerful technique for unsupervised learning and clustering in particular.
 - Can deal with missing data, categorical data (without the biases of K-means)
-

A mixture model is one in which a set of component models is combined to produce a richer model:

$$p(v) = \sum_{h=1}^H p(v|h)p(h)$$

The variable v is 'visible' or 'observable' and $h = 1, \dots, H$ indexes each component model $p(v|h)$, along with its weight $p(h)$.

Clustering

Mixture models have a natural interpretation in terms of clustering with each state of h corresponding to a cluster model $p(v|h)$.

Learning the Parameters of a Mixture Model

- For a dataset $\mathbf{v}^1, \dots, \mathbf{v}^N$, the model has a probability of generating this data

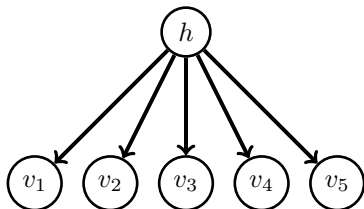
$$L(\theta) \equiv \prod_{n=1}^N p(\mathbf{v}^n | \theta)$$

- The EM (Expectation Maximisation) algorithm is an iterative procedure for maximising $L(\theta)$ with respect to the model parameters θ .
- Typically the EM algorithm (and other parameter learning algorithms) are highly sensitive to the initial guess for the parameters. In practice we therefore run the algorithm several times, using different random initial parameters.
- The best parameters correspond to the highest likelihood value.
- Once trained, we compute the posterior cluster assignment for each datapoint

$$p(h | \mathbf{v}^n) \propto p(\mathbf{v}^n | h) p(h)$$

We can make a hard assignment by finding which h has the highest posterior probability.

Mixture of Independent Bernoulli



A model that can be used to cluster a set of binary vectors, $\mathbf{v}^n = (v_1^n, \dots, v_D^n)^\top$, $v_i \in \{0, 1\}$, $n = 1, \dots, N$ is

$$p(\mathbf{v}) = \sum_{h=1}^H p(h) \prod_{i=1}^D p(v_i|h)$$

where each term $p(v_i|h)$ is a Bernoulli distribution.

Parameters

We need to learn $p(v_i = 1|h = h)$ and $p(h = h)$.

An example: clustering binary digits

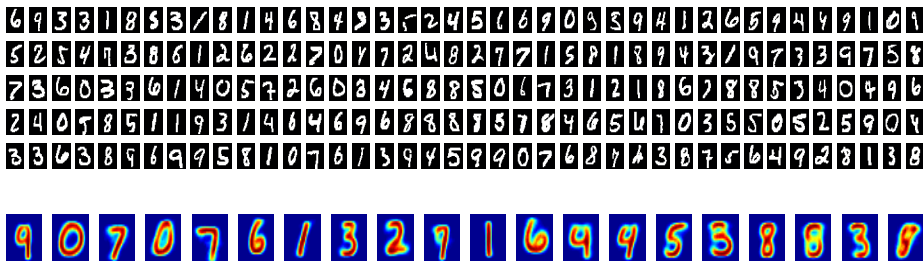


Figure : Top: a selection of 200 of the 5000 handwritten digits in the training set. Bottom: the trained cluster outputs $p(v_i = 1|h)$ for $h = 1, \dots, 20$ mixtures. See [demoMixBernoulliDigits.m](#).

An example: clustering binary digits with missing data

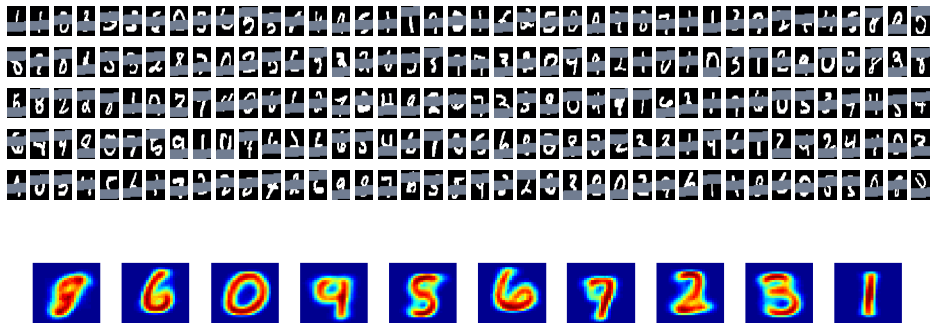


Figure : Top: a selection of the 5000 handwritten digits in the training set with missing data. Bottom: the trained cluster outputs $p(v_i = 1|h)$ for $h = 1, \dots, 10$ mixtures. See [demoMixBernoulliDigitsMissing.m](#).

Clustering Questionnaires using Bernoulli Mixture

If an attribute i is missing for this model one simply removes the corresponding factor $p(v_i^n|h)$ from the algorithm.

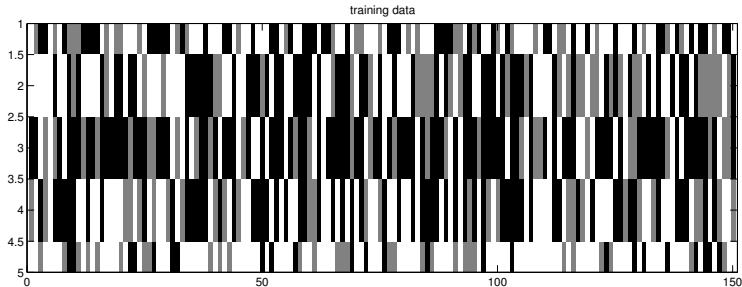


Figure : Data from questionnaire responses. 150 people were each asked 5 questions, with 'yes' (white) and 'no' (gray) answers. Black denotes the absence of a response (missing data). This training data was generated by two component Binomial mixture. Missing data was simulated by randomly removing values from the dataset. See [demoMixBernoulli.m](#)

Clustering Questionnaires using Bernoulli Mixture

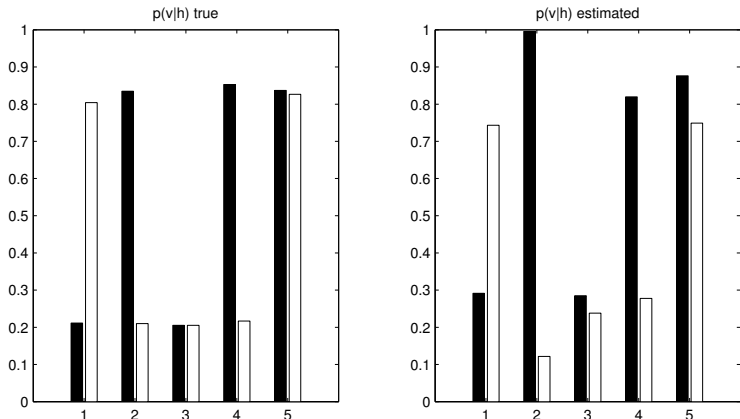


Figure : The two components found. On the left is the true data generating probability $p(v_i|h)$, $i = 1, \dots, 5$ for each of the 5 questions and each of the two clusters (black and white). On the right are the corresponding probabilities learned by maximum likelihood from the dataset alone.

ABC Questionnaire



















Q1: Is Tesco's expensive? (yes,no,somewhat)

Q2: Is the ambience of the supermarket important? (yes,no,somewhat)

Q3: Do you like music to be played in the store? (yes,no,somewhat)

Q4: Do you use the cafe in the store? (often,never,sometimes)

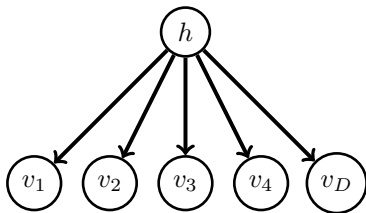
Q5: Do you get annoyed by frequent product reshelfing? (yes,no,a bit)

	1	2	3	4	5	6
Q1						
Q2						
Q3						
Q4						
Q5						

Questionnaire data is often categorical and has missing entries.

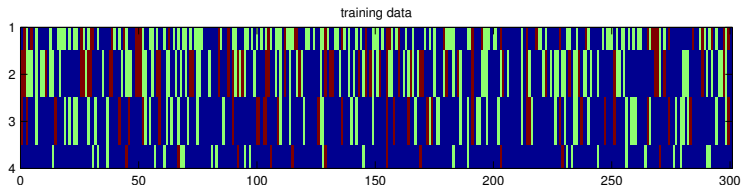
ABC questionnaire

- We have a questionnaire with D questions. Each response is 'a', 'b' or 'c' (or missing).
- Want to cluster the responders into K clusters.



- This time $p(v_i = c|h)$ is a categorical distribution. For each attribute i of the datavector v we have the probability that it belongs to category c . Each cluster h has a different set of such distributions.
- We can run the EM algorithm again on this model.

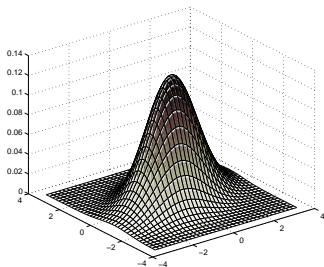
ABC questionnaire



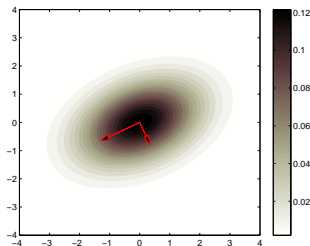
- There are 4 questions, each with possible red, blue, green responses. 300 people returned fully completed questionnaires.
- Let's assume there are two clusters, $h^n \in \{1, 2\}$.
- We look at $p(h^n | \mathbf{v}^n)$ and find the most likely state of h^n . This is the cluster label we give to datapoint n .
- Running the demo below, the K-means (1-of-M) and mixture model cluster labels can differ significantly in which datapoints are assigned together into a cluster.
- Straightforward to generalise to any number of states for the categorical variables (or have different numbers of categories for different variables).

`demoMixCategorical.m`

The multivariate Gaussian distribution



(a)



(b)

Figure : (a): Bivariate Gaussian. (b): Probability density contours.

$$p(\mathbf{x}) = \frac{1}{\sqrt{\det(2\pi\mathbf{S})}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m})^\top \mathbf{S}^{-1} (\mathbf{x} - \mathbf{m}) \right\}$$

where \mathbf{m} is the mean and \mathbf{S} is the covariance matrix.

The Gaussian Mixture Model

- A mixture of Gaussians is

$$p(\mathbf{x}) = \sum_{i=1}^H p(\mathbf{x}|\mathbf{m}_i, \mathbf{S}_i)p(i)$$

where $p(i)$ is the mixture weight for component i .

- This means that we can describe the data density by composing 'blobs' of localised Gaussian densities.
- The GMM is very widely used, typically with the simplifying constraint that the covariance matrices are diagonal.

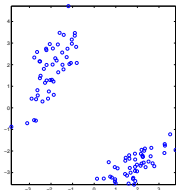


Figure : Two dimensional data which displays clusters. In this case a Gaussian mixture model $1/2\mathcal{N}(\mathbf{x}|\mathbf{m}_1, \mathbf{C}_1) + 1/2\mathcal{N}(\mathbf{x}|\mathbf{m}_2, \mathbf{C}_2)$ would fit the data well for suitable means $\mathbf{m}_1, \mathbf{m}_2$ and covariances $\mathbf{C}_1, \mathbf{C}_2$.

Maximum Likelihood

For a set of data $\mathcal{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ the log likelihood is

$$\sum_{n=1}^N \log \sum_{i=1}^H p(i) \frac{1}{\sqrt{\det(2\pi \mathbf{S}_i)}} \exp \left\{ -\frac{1}{2} (\mathbf{x}^n - \mathbf{m}_i)^\top \mathbf{S}_i^{-1} (\mathbf{x}^n - \mathbf{m}_i) \right\}$$

Parameter Constraints

- The \mathbf{S}_i must be symmetric positive definite matrices, in addition to $0 \leq p(i) \leq 1$, $\sum_i p(i) = 1$.
- The EM approach which in this case is particularly convenient since it automatically provides parameter updates that ensure these constraints.
- Constraints on the covariance matrices (such as being diagonal) are straightforward to incorporate.

Infinite troubles

Consider placing a component $p(\mathbf{x}|\mathbf{m}_i, \mathbf{S}_i)$ with mean \mathbf{m}_i set to one of the datapoints $\mathbf{m}_i = \mathbf{x}^n$. The contribution from that Gaussian will be

$$p(\mathbf{x}^n|\mathbf{m}_i, \mathbf{S}_i) = \frac{1}{\sqrt{\det(2\pi\mathbf{S}_i)}} e^{-\frac{1}{2}(\mathbf{x}^n - \mathbf{x}^n)^\top \mathbf{S}_i^{-1}(\mathbf{x}^n - \mathbf{x}^n)} = \frac{1}{\sqrt{\det(2\pi\mathbf{S}_i)}}$$

As the covariance goes to zero, this probability density becomes infinite. This means that one can obtain a Maximum Likelihood solution by placing zero-width Gaussians on a selection of the datapoints, resulting in an infinite likelihood.

Remedy

Include an additional constraint on the width of the Gaussians, ensuring that they cannot become too small.

Symmetry Breaking

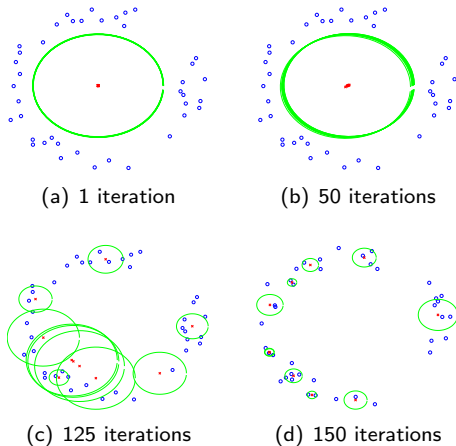


Figure : Training a mixture of 10 isotropic Gaussians **(a)**: If we start with large variances for the Gaussians, even after one iteration, the Gaussians are centred close to the mean of the data. **(b)**: The Gaussians begin to separate **(c)**: One by one, the Gaussians move towards appropriate parts of the data **(d)**: The final converged solution. The Gaussians are constrained to have variances greater than a set amount.

GMM and K-Means

Consider a mixture of K Gaussians in which each covariance is constrained to be equal to $\sigma^2 \mathbf{I}$,

$$p(\mathbf{x}) = \sum_{i=1}^K p_i \mathcal{N}(\mathbf{x} | \mathbf{m}_i, \sigma^2 \mathbf{I})$$

- As $\sigma^2 \rightarrow 0$, one can show that the parameter updates in the EM algorithm become equivalent to the K-means updates.
- K-means can therefore be seen as a special case of the GMM.
- For non-zero σ^2 the GMM gives a 'softer' clustering than K-means.

Extensions

- We can use any distribution $p(x_i|h)$ that we wish.
 - Could e.g. use a Gaussian for say x_1 and a categorical distribution for x_2 .
 - These models are straightforward to train using the EM algorithm.
 - The Gaussian Mixture Model is not always very robust to outliers.
 - Using a student t distribution is a common way to make the clustering more robust to outliers.
-

Finding the number of mixture components

- Most common approach is to use some hold-out data (not part of the training set).
- After training the parameters of the model on the training data, we compute the likelihood of the hold-out data using these parameters.
- That model which has the highest hold-out likelihood is deemed the best model (number of mixture components).
- There are other approaches involving Bayesian statistics (very complex) and simpler techniques (such as the Bayesian Information Criterion).

Summary

Clustering is a natural way to simplify and give insight into data.

K-means

- Very popular approach and easy to implement ●
 - Easy to perform hierarchical clustering ●
 - Sensitive to initialisation of cluster centres ●
 - Not clear how to cluster data which doesn't have a simple interpretation in terms of a vector ●
-

Mixture Models

- Natural way to perform clustering in a probabilistic setting ●
- Significantly generalises K-means ●
- Some models are very straightforward to implement – can be even faster than K-means ●
- Can deal with missing data and data that doesn't have a natural vector representation ●
- Also sensitive to initialisation ●