Clustering

Ciira Maina

Dedan Kimathi University of Technology

17th June 2015

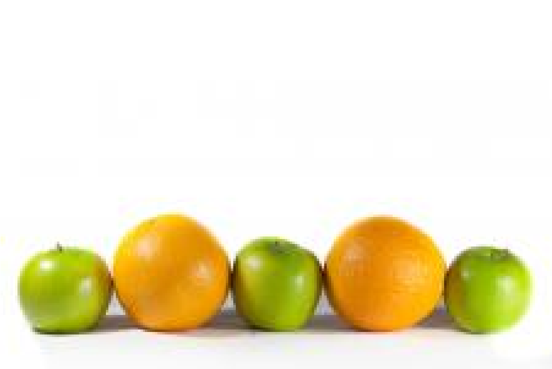
Introduction

- In most data science applications we are start off with a large collection of objects which form our data set.
- Clustering is often an initial exploratory operation applied to the data.
- The aim of clustering is the grouping of objects into subsets with closely related objects in the same group or cluster.

Introduction

Sheep vs. Goats [Source Wikipedia]

Introduction



Apples vs. Oranges [Source: `http://www.microassist.com/`]

Introduction

- Clustering has a number of applications such as:
  - Image segmentation for lossy image compression
  - Audio processing applications like diarization and voice activity detection
  - Clustering gene expression data
  - Wireless network base station cooperation

Introduction

- Here we will consider a number of clustering algorithms:
  - K-means clustering
  - Gaussian mixture modelling
  - Hierachical clustering

K-means

- Given a set of N data points, the goal of K-means clustering is to assign each data point to one of K groups
- Each cluster is characterised by a cluster mean μ_k $k = 1, \ldots, K$
- The data points are assigned to the clusters such that the average dissimilarity of data points in the cluster from the cluster mean is minimized.
- In K-means clustering the dissimilarity is measured using Euclidean distance

K-means, Example

- Consider 2D data from two distinct clusters. K-means does a good job of discovering these clusters.

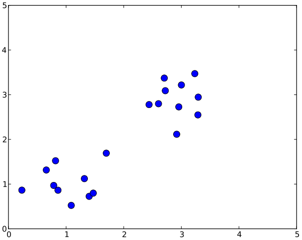


Figure: Data with two distinct clusters

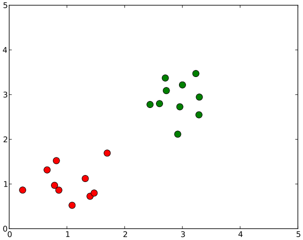


Figure: Result of K-means clustering

K-means, The Theory

- Consider the N data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ which we would like to partition into K clusters.
- We introduce K cluster centers μ_k $k = 1, \ldots, K$ and corresponding indicator variables $r_{n,k} \in \{0, 1\}$ where $r_{n,k} = 1$ if $\mathbf{x}_n$ belongs to cluster k.
- The objective function is the sum of square distances of the data points to assigned cluster centers. That is

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{n,k} ||\mathbf{x}_n - \mu_k||^2$$

K-means, The Theory

1. The K-means algorithm proceeds iteratively. Starting with an initial set of cluster centers, the variables $r_{n,k}$ are determined.

$$r_{n,k} = \begin{cases} 1 & \text{if } k = \text{argmin}_j ||\mathbf{x}_n - \mu_j||^2 \\ 0 & \text{otherwise} \end{cases}$$

2. In the next step, the cluster centers are updated based on the current assignment

$$\mu_k = \frac{\sum_n r_{n,k} \mathbf{x}_n}{\sum_n r_{n,k}}$$

3. Step 1 and 2 are repeated until the assignment remains unchanged or the relative change in J is small.

K-means, Example

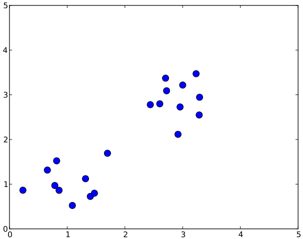


Figure: Data with two distinct clusters

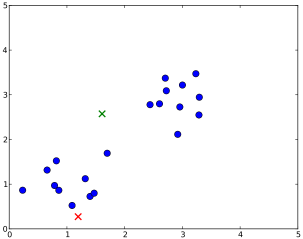


Figure: Randomly initialize the cluster centers

K-means, Example

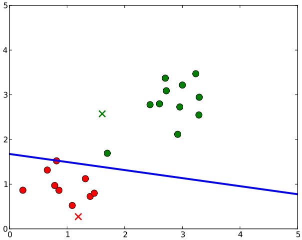


Figure: Assign data points to cluster centers

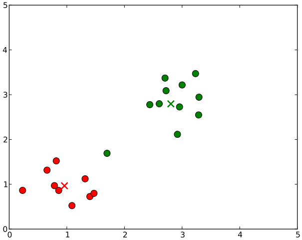


Figure: Recompute cluster centers

K-means, Example

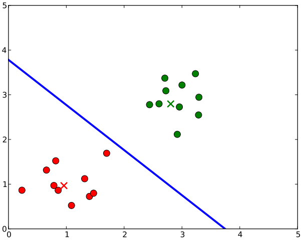


Figure: Assign data points to cluster centers

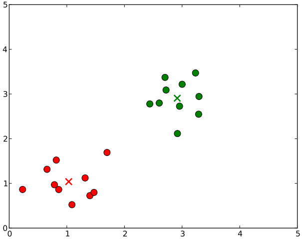
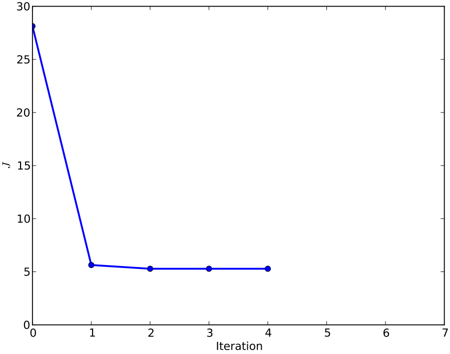


Figure: Recompute cluster centers

K-means, Example

- To determine when to stop K-means, we monitor the cost function J.
- In this case, 3 iterations are sufficient



K-means, Image compression Example

- K-means clustering can be used in image compression using vector quantization.
- This algorithm takes advantage of the fact that several nearby pixels of an image often appear the same.
- The image is divided into blocks which are then clustered using K-means.
- The blocks are then represented using the centroids of the clusters to which they belong.

K-means, Image compression Example

- In this example we start with a 196-by-196 pixel image of Mzee Jomo Kenyatta
- We divide the image into 2-by-2 blocks and treat these blocks as vectors in $\mathbf{R}^4$
- These vectors are clustered with $K = 100$ and $K = 10$
- The resulting image shows degradation but uses fewer bytes for storage

Figure: Original Image

Figure: VQ with 100 classes

Figure: VQ with 10 classes

K-means, Image compression Example

- The original image requires $196 \times 196 \times 8$ bits.
- To store the cluster to which each 2×2 block belongs to we require $\log_2(K)$ bits
- To store the cluster centers we need $K \times 4$ real numbers
- The total storage for the compressed image is $\log_2(K) \times \#\text{blocks} = \log_2(K) \times \frac{196^2}{4}$
- When $K = 10$, we can compress the image to $\frac{log_2(10)}{32} = 0.103$ of its original size

K-means, Practical Issues

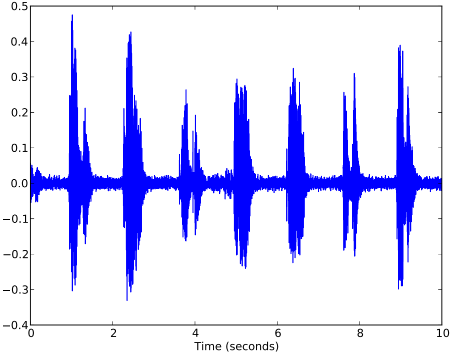
1. To avoid local minima we should have multiple random initializations.
2. Initial cluster centers chosen randomly from the data points.
3. Choosing K- Elbow method.

Gaussian Mixture Models

- So far we have considered situations where each data point is assigned to only one cluster.
- This is sometimes referred to as **hard clustering**
- In several cases it may be more approriate to consider assigning each data point a probability of membership to each cluster.
- This is **soft clustering**
- Gaussian Mixture Models are useful for soft clustering

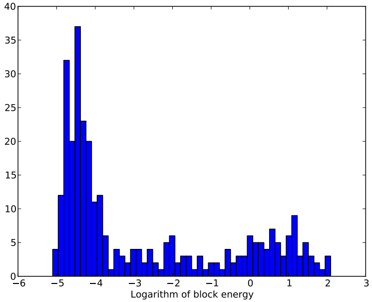
Gaussian Mixture Models

- GMMs are ideal for modelling continuous data that can be grouped into distinct clusters.
- For example consider a speech signal which contains regions with speech and other regions with silence
- We could use a GMM to decide which category a certain segment belongs to.



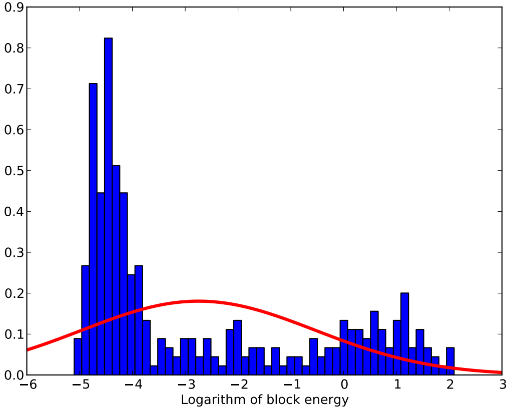
Gaussian Mixture Models, VAD Example

- Voice activity detection is a useful signal processing application
- It involves deciding whether a speech segment is speech or silence
- We divide the speech into short segments and compute the logarithm of the energy of each segment.
- We see that the log energy shows distinct clusters.



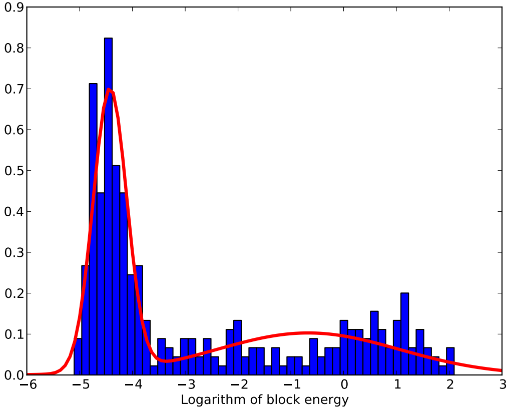
Gaussian Mixture Models, VAD Example

- A single Gaussian does not fit the data well



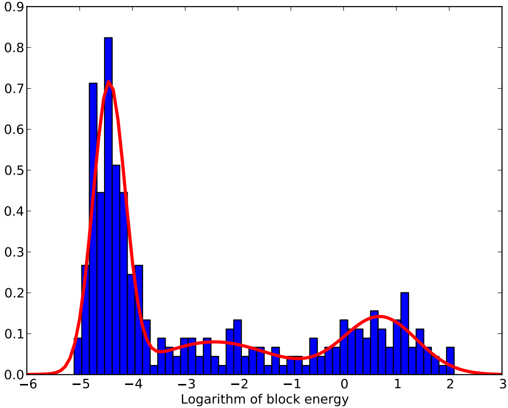
Gaussian Mixture Models, VAD Example

- Two Gaussians do a better job



Gaussian Mixture Models, VAD Example

- Are three Gaussians even better?



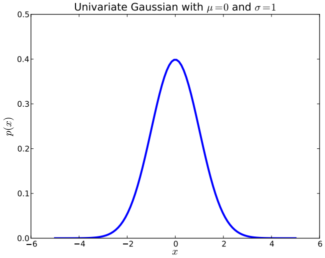
Gaussian Mixture Models, Theory

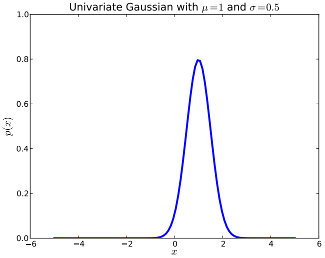
- The Gaussian distribution function for a 1D variable is given by

$$p(x) = \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp\left\{ -\frac{1}{2\sigma^2}(x-\mu)^2 \right\}$$

- The distribution is governed by two parameters
  - The mean μ
  - The variance σ^2
- The mean determines where the distribution is centered and the variance determines the spread of the distribution around this mean.

Gaussian Mixture Models, Theory





Gaussian Mixture Models, Theory

- The Gaussian density can not be used to model data with more than one distinct 'clump' like the log energy of the speech frames.
- Linear combinations of more than one Gaussian can capture this structure.
- These distributions are known as Gaussian Mixture Models (GMMs) or Mixture of Gaussians

Gaussian Mixture Models, Theory

- The GMM density takes the form

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k)$$

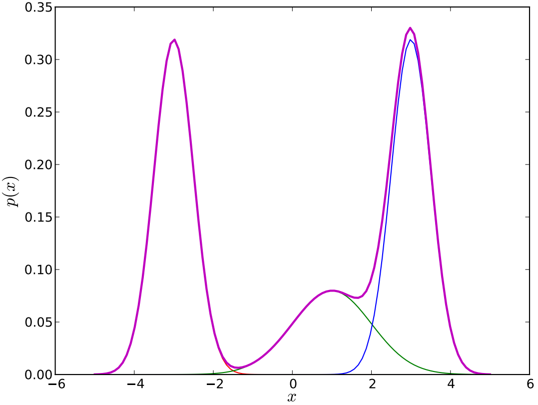
- π_k is known as a mixing coefficient. We have

$$\sum_{k=1}^{K} \pi_k = 1$$

and $0 \leq \pi_k \leq 1$

Gaussian Mixture Models, Theory

- A GMM with three mixture components



Gaussian Mixture Models, Theory

- The mixing coefficients can be viewed as the prior probability of the components of the mixture
- We can then use the sum and product rules and write

$$p(x) = \sum_{k=1}^{K} p(k)p(x|k)$$

- Where

$$p(k) = \pi_k$$

and

$$p(x|k) = \mathcal{N}(x|\mu_k, \sigma_k)$$

Gaussian Mixture Models, Theory

- Given an observation x, we will be interested to compute the posterior probability of each component that is $p(k|x)$
- We use Bayes' rule

$$
\begin{aligned}
p(k|x) &= \frac{p(x|k)p(k)}{p(x)} \\
&= \frac{p(x|k)p(k)}{\sum_i p(x|i)p(i)}
\end{aligned}
$$

- We can use this posterior to build a classifier

Gaussian Mixture Models, Learning the model

- Given a set of observations $\mathbf{X} = \{x_1, x_2, \dots, x_N\}$ where the observations are assumed to be drawn independently from a GMM, the log likelihood function is given by

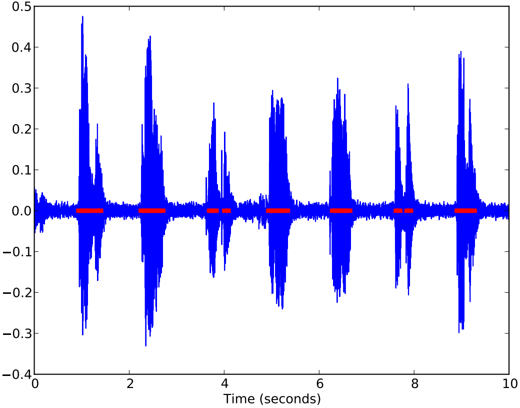
$$\ell(\theta; \mathbf{X}) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i | \mu_k, \sigma_k) \right\}$$

  where $\theta = \{\pi_1, \dots, \pi_K, \mu_1, \dots, \mu_K, \sigma_1^2, \dots, \sigma_K^2\}$ are the parameters of the GMM.
- To obtain a maximum likelihood estimate of the parameters, we use the expectation maximization (EM) algorithm

Gaussian Mixture Models, Returning to the VAD Example

- In the VAD example we use the implementation of EM in scikit-learn.
- We can then compute the posterior probability of all segments belonging to the component with the highest mean.
- Segments where this probability is greater than a threshold can be classified as speech.

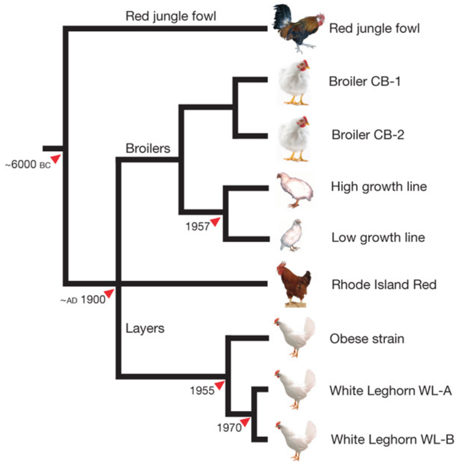
Gaussian Mixture Models, Returning to the VAD Example



Hierachical Clustering

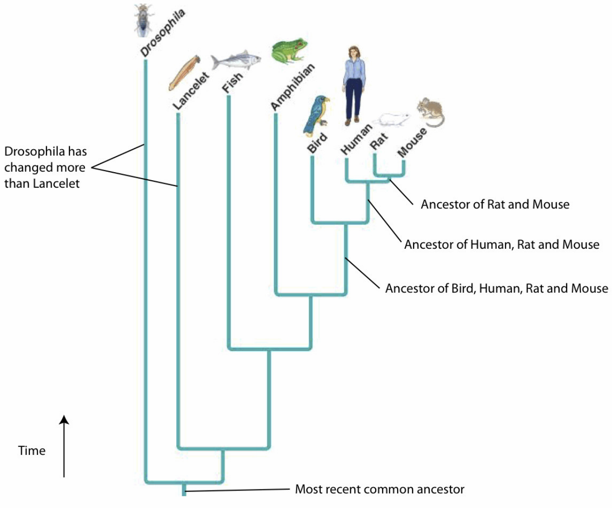
- An approach to clustering that yields a hierarchy of clusters.
- Clusters in one level of the hierarchy are formed by merging clusters in the lower level.
- At the lowest level of the hierarchy each datum is in its own cluster.

Hierachical Clustering



Source: `mikethechickenvet.wordpress.com`

Hierachical Clustering



Source: `http://guestblog.scientopia.org/`

Hierachical Clustering

- There are two main stategies:
  - Agglomerative (bottom-up): Start with each item as a cluster and succeccively merge clusters
  - Divisive (top-down): Start with all items in one cluster and recursively divide one of the exisiting clusters into two.

Agglomerative Clustering

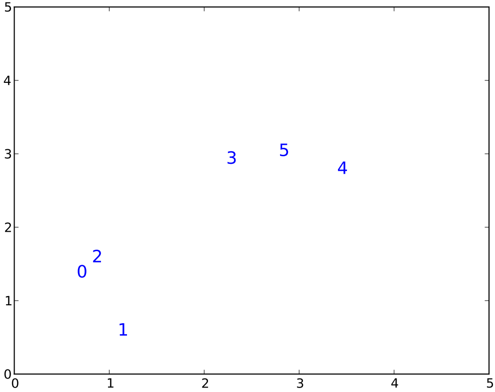
- In agglomerative we begin with each data point in a singleton cluster.
- At each step the two closest clusters are merged.
- We must specify a measure of dissimilarity between the clusters. This will be problem specific
- If there are N data points there will be $N - 1$ steps. At each step there is one less cluster.

Agglomerative Clustering-Measures of Dissimilarity

- If $\mathcal{C}_1$ and $\mathcal{C}_2$ are two clusters, the dissimilarity between them is denoted $d(\mathcal{C}_1, \mathcal{C}_2)$ and is based on the pairwise dissimilarity of items in each of the clusters.
- Let $d_{ii'}$ be the dissimilarity between $i \in \mathcal{C}_1$ and $i' \in \mathcal{C}_2$.
- We can define the dissimilarity between the clusters in different ways
    - Single linkage:
    $$d(\mathcal{C}_1, \mathcal{C}_2) = \min_{i \in \mathcal{C}_1, i' \in \mathcal{C}_2} d_{ii'}$$
    - Complete linkage:
    $$d(\mathcal{C}_1, \mathcal{C}_2) = \max_{i \in \mathcal{C}_1, i' \in \mathcal{C}_2} d_{ii'}$$
    - Average linkage:
    $$d(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{|\mathcal{C}_1||\mathcal{C}_2|} \sum_{i \in \mathcal{C}_1} \sum_{i' \in \mathcal{C}_2} d_{ii'}$$

Agglomerative Clustering-Example

- Consider the dataset in the figure below



Agglomerative Clustering-Example

- The first step is to compute pair-wise dissimilarity between the objects and find the closest pair of clusters. Here we use Euclidean distance

	0	1	2	3	4	5
0	-	0.902	**0.262**	2.21	3.085	2.696
1		-	1.035	2.605	3.192	2.977
2			-	1.951	2.85	2.443
3				-	1.176	0.563
4					-	0.662
5						-

- Merge $\{0\}$ and $\{2\}$ to form a new cluster $\{0, 2\}$

Agglomerative Clustering-Example

- We then compute the distance between this new cluster and the remaining clusters **using single linkage**

	$\{0,2\}$	1	3	4	5
$\{0,2\}$	-	0.902	1.951	2.85	2.696
1		-	2.605	3.192	2.977
3			-	1.176	**0.563**
4				-	0.662
5					-

- Merge $\{3\}$ and $\{5\}$ to form a new cluster $\{3,5\}$

Agglomerative Clustering-Example

- The process of finding the pair of clusters with least dissimilarity is repeated.

	$\{0,2\}$	$\{3,5\}$	1	4
$\{0,2\}$	-	1.951	0.902	2.85
$\{3,5\}$		-	2.605	**0.662**
1			-	3.192
4				-

- Merge $\{3,5\}$ and $\{4\}$ to form a new cluster $\{3,4,5\}$

Agglomerative Clustering-Example

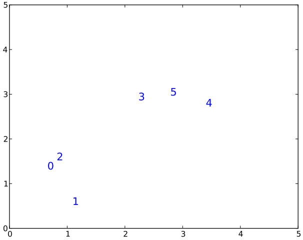
- Then...

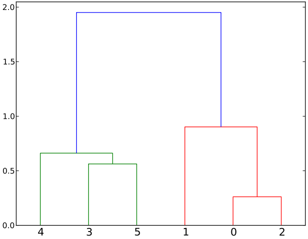
	$\{0,2\}$	$\{3,4,5\}$	1
$\{0,2\}$	-	1.951	**0.902**
$\{3,4,5\}$		-	2.605
1			-

- Merge $\{1\}$ and $\{0,2\}$ to form a new cluster $\{0,1,2\}$

Agglomerative Clustering-A dendogram

- We can use a dendogram to give a pictorial representation of the clustering.
- A node whose daughters are the merged clusters is formed at a height equal to the dissimilarity between the clusters.



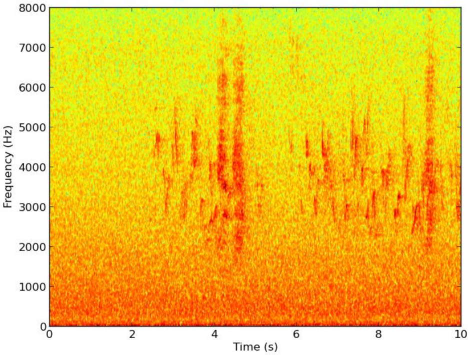


Agglomerative Clustering-Application to Audio Diarization

- We may want to cluster sections of audio according to ‘who spoke when’
- This is known as audio diarization.
- We begin by detecting change points in the audio to form initial clusters.
- We the perform agglomerative clustering on the initial clusters

Agglomerative Clustering-Application to Audio Diarization

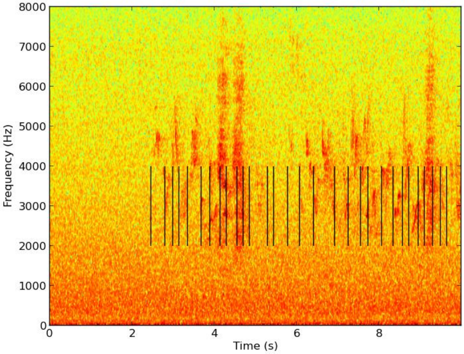
- This example shows a recording of bird sounds with vocalisation from two species
- The data set was used in the 2013 Machine Learning for Signal Processing (MLSP) competition and is freely available[1]



[1] https://www.kaggle.com/c/mlsp-2013-birds/data

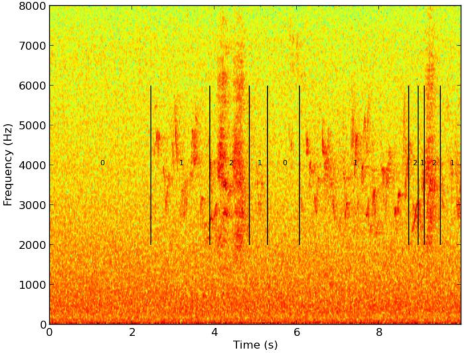
Agglomerative Clustering-Application to Audio Diarization

- We perform change point detection to discover initial clusters of sound segments.



Agglomerative Clustering-Application to Audio Diarization

- Perform agglomerative clustering on this initial set of clusters to discover segments of audio produced by the same species.
- Code to reproduce the results is available on Github (https://github.com/ciiram/BirdPy)



Conclusion

- We have covered three main methods of clustering
  - K-means clustering
  - Gaussian mixture modelling
  - Hierachical clustering
- We have demonstrate the use of clustering in
  - Image compression
  - Voice activity detection
  - Audio Diarization
- In the talks we will consider clustering of gene sequence data

Conclusion

- Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning (Vol. 1)*. Springer, Berlin: Springer series in statistics.

Thank You