UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Unsupervised Learning

Alta de Waal

Department of Statistics
University of Pretoria, South Africa

13 September 2017

Overview

- Bayesian Concept Learning
- Dimensionality Reduction
- Clustering
- Evaluation
- Resources

How does a child learn a word?

How does a child learn a word?

- Positive examples

How does a child learn a word?

- Positive examples
- Active learning involves negative examples

How does a child learn a word?

- Positive examples
- Active learning involves negative examples
- Phsychological research has shown that people can learn concepts from positive examples alone

Concept Learning

Learning the meaning of a word is equivalent to concept learning, which in turn is equivalent to binary classification.

Definition

Define $f(x) = 1$ if x is an example of the concept C and $f(x) = 0$ otherwise. The goal is to learn the indicator function f, which defines which elements are in the set C

Concept Learning

Learning the meaning of a word is equivalent to concept learning, which in turn is equivalent to binary classification.

Definition

Define $f(x) = 1$ if x is an example of the concept C and $f(x) = 0$ otherwise. The goal is to learn the indicator function f, which defines which elements are in the set C

Learn from positive examples

Note that standard binary classification techniques require positive and negative examples. By contrast, we will devise a way to learn from **positive examples alone**.

Number Game (Tennenbaum, 1999)

The concept C

Number Game (Tennenbaum, 1999)

The concept C

- Integers between 1 and 100

Number Game (Tennenbaum, 1999)

The concept C

- Integers between 1 and 100
- Suppose I tell you $\mathcal{D} = \{16\}$ is a positive example of the concept.

Number Game (Tennenbaum, 1999)

The concept C

- Integers between 1 and 100
- Suppose I tell you $\mathcal{D} = \{16\}$ is a positive example of the concept.
- What other numbers do you think are positive?

Number Game (Tennenbaum, 1999)

The concept C

- Integers between 1 and 100
- Suppose I tell you $\mathcal{D} = \{16\}$ is a positive example of the concept.
- What other numbers do you think are positive?
- Presumably numbers that are similar in some sense to 16 are more likely.

Number Game (Tennenbaum, 1999)

The concept C

- Integers between 1 and 100
- Suppose I tell you $\mathcal{D} = \{16\}$ is a positive example of the concept.
- What other numbers do you think are positive?
- Presumably numbers that are similar in some sense to 16 are more likely.
- But similar in what why?

Number Game (Tennenbaum, 1999)

The concept C

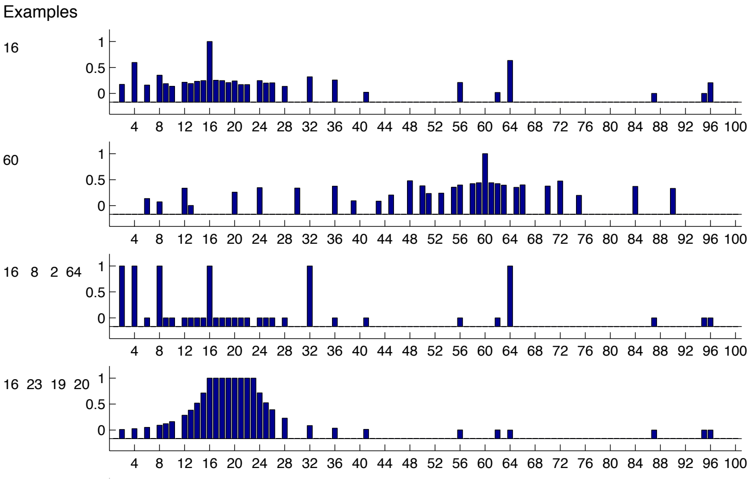
- Integers between 1 and 100
- Suppose I tell you $\mathcal{D} = \{16\}$ is a positive example of the concept.
- What other numbers do you think are positive?
- Presumably numbers that are similar in some sense to 16 are more likely.
- But similar in what why?
- Suppose I update the positive examples to $\mathcal{D} = \{2, 8, 16, 64\}$

Number Game (Tennenbaum, 1999)

The concept C

- Integers between 1 and 100
- Suppose I tell you $\mathcal{D} = \{16\}$ is a positive example of the concept.
- What other numbers do you think are positive?
- Presumably numbers that are similar in some sense to 16 are more likely.
- But similar in what why?
- Suppose I update the positive examples to $\mathcal{D} = \{2, 8, 16, 64\}$
- What other numbers do you think are positive?

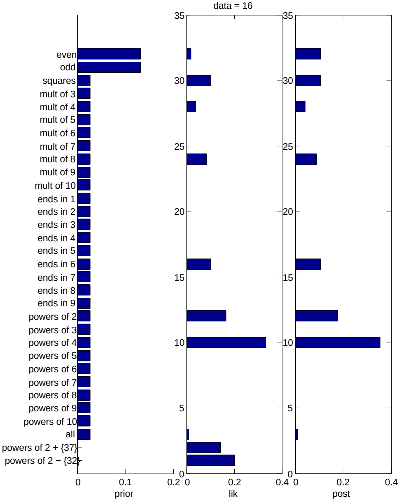
Human Experiment



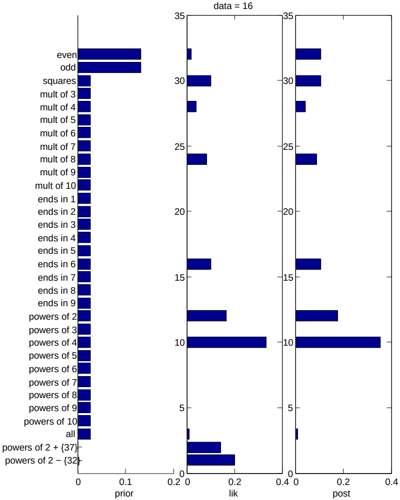
Plausible concepts:

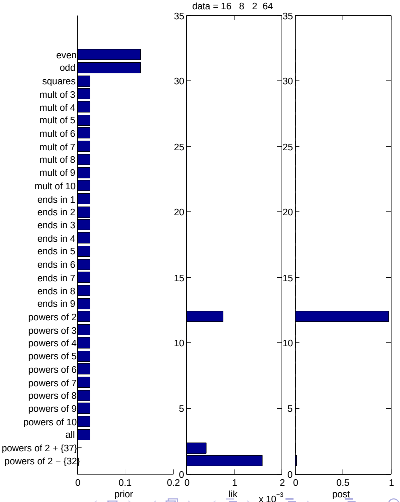
- Powers of two
- Even numbers
- Powers of two except 32
- Prime numbers
- Odd numbers

Bayesian Concept Learning



Bayesian Concept Learning





Unsupervised Learning

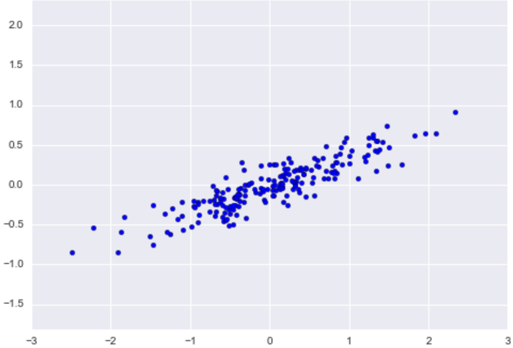
- Supervised learning: predict labels based on labelled training data
- No reference to any known labels
- Dimensionality reduction
- Clustering

Principal Component Analysis (PCA)

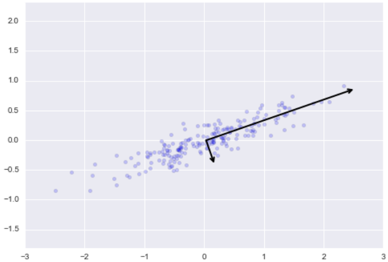
- Dimensionality reduction
- Visualisation
- Noise filtering
- Feature extraction

Principal Component Analysis (PCA)

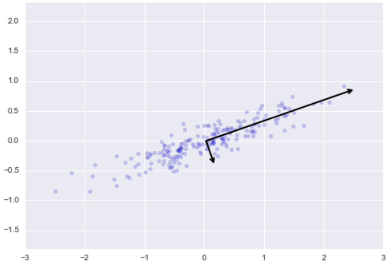
- Dimensionality reduction
- Visualisation
- Noise filtering
- Feature extraction

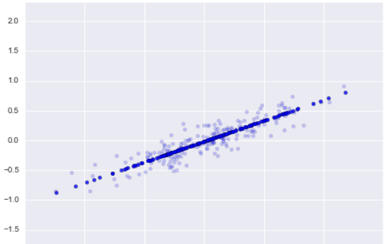


PCA for dimensionality reduction

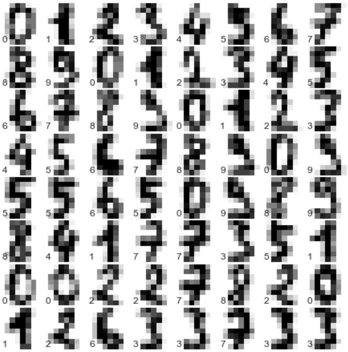


PCA for dimensionality reduction

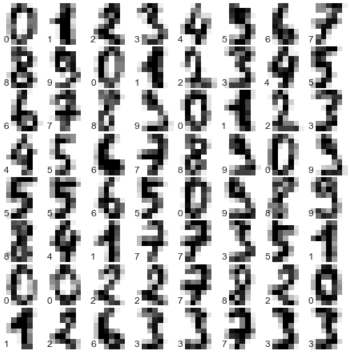




PCA for visualisation



PCA for visualisation



PCA for noise filtering and feature selection

- Reconstruction of images from just 150 of the ~3000 initial features.
- Dimensionality of the data is reduced by nearly a factor of 20
- The projected images contain enough information that we might, by eye, recognise the individuals in the image

PCA - Summary

- Effective in a wide variety of contexts
- Good starting point in order to visualize:
  - the relationship between observations
  - the main variance in the data
- Understand the intrinsic dimensionality of the data
- Offers a straightforward and efficient path to gain insight into high-dimensional data
- Weaknesses:
  - Highly affected by outliers in the data
  - Doesn't perform well with non-linear relationships in data
    - Manifold learning
    - Multidimensional scaling (MDS)

k-Means

Clustering seek to learn an optimal division or discrete labeling of groups of points.

The k-Means algorithm searches for a **pre-determined number** of clusters within an unlabeled multidimensional dataset.

Simple conception of what the optimal clustering looks like

- The "cluster center" is the arithmetic mean of all the points belonging to the cluster.
- Each point is closer to its own cluster center than to other cluster centers.

How does it work?

How does it work?

Objective

- Subdivide data points of a dataset into clusters based on nearest mean values
- Minimise the distance between points in each cluster

How does it work?

Objective

- Subdivide data points of a dataset into clusters based on nearest mean values
- Minimise the distance between points in each cluster

K

- denotes the number of clusters in the data
- must be specified (not determined by algorithm)

How does it work?

Objective

- Subdivide data points of a dataset into clusters based on nearest mean values
- Minimise the distance between points in each cluster

K

- denotes the number of clusters in the data
- must be specified (not determined by algorithm)

Input

- X – n data points (1, 2, n-dimensional)
- K – number of clusters

How does it work?

Objective

- Subdivide data points of a dataset into clusters based on nearest mean values
- Minimise the distance between points in each cluster

K

- denotes the number of clusters in the data
- must be specified (not determined by algorithm)

Input

- X – n data points (1, 2, n-dimensional)
- K – number of clusters

Output

- A set of k cluster centroids
- Labeling of X that assigns each of the points in X to a unique cluster

EM Algorithm

EM Algorithm

1. Guess some cluster centers

EM Algorithm

1. Guess some cluster centers
2. Repeat until converged

EM Algorithm

1. Guess some cluster centers
2. Repeat until converged
    1. E-Step: assign points to the nearest cluster center

EM Algorithm

1. Guess some cluster centers
2. Repeat until converged
    1. E-Step: assign points to the nearest cluster center
    2. M-Step: set the cluster centers to the mean

EM Algorithm

1. Guess some cluster centers
2. Repeat until converged
    1. E-Step: assign points to the nearest cluster center
    2. M-Step: set the cluster centers to the mean

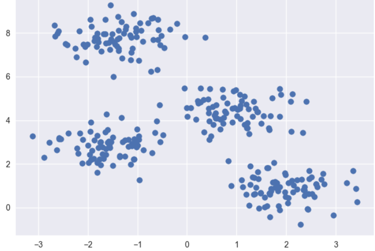
- E-Step: involves updating our expectation of which cluster each point belongs to

EM Algorithm

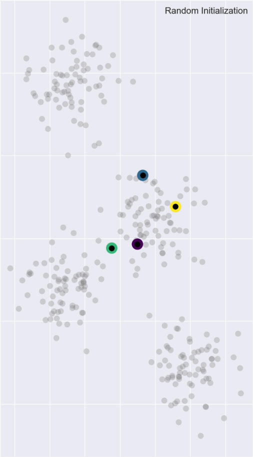
1. Guess some cluster centers
2. Repeat until converged
    1. E-Step: assign points to the nearest cluster center
    2. M-Step: set the cluster centers to the mean

- E-Step: involves updating our expectation of which cluster each point belongs to
- M-Step: involves maximizing some fitness function that defines the location of the cluster centersin this case,by taking a simple mean of the data in each cluster

Example

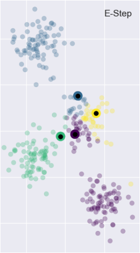


1. Guess some cluster centers (Initialise $\boldsymbol{\mu}_i$)

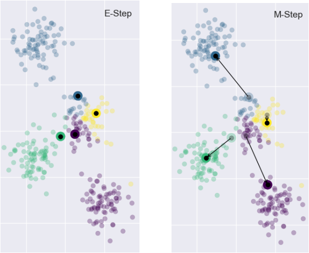


2. Repeat until converged

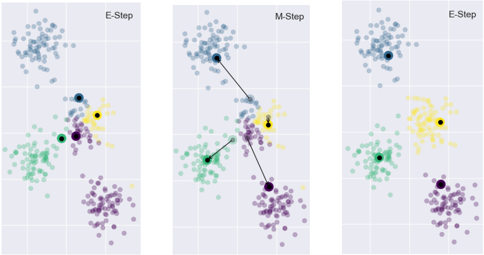
2. Repeat until converged



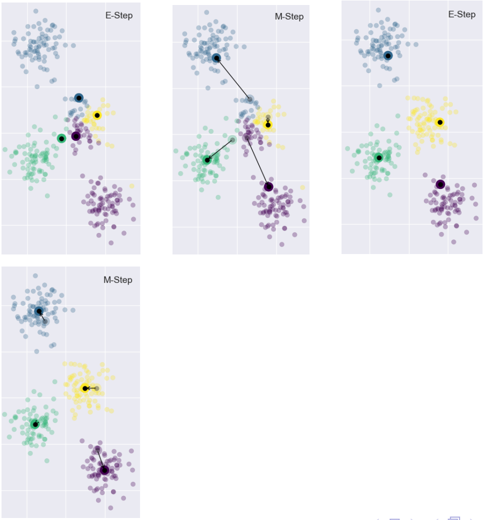
2. Repeat until converged



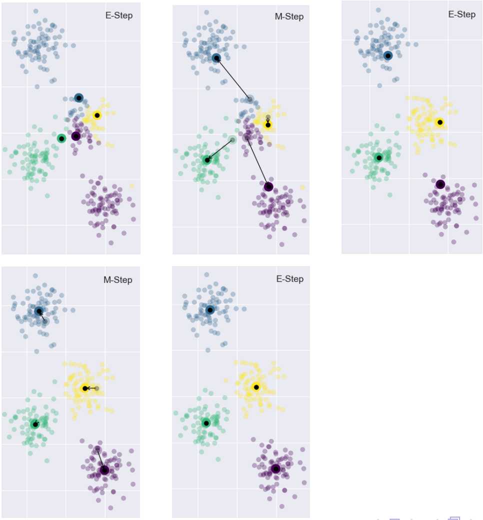
2. Repeat until converged



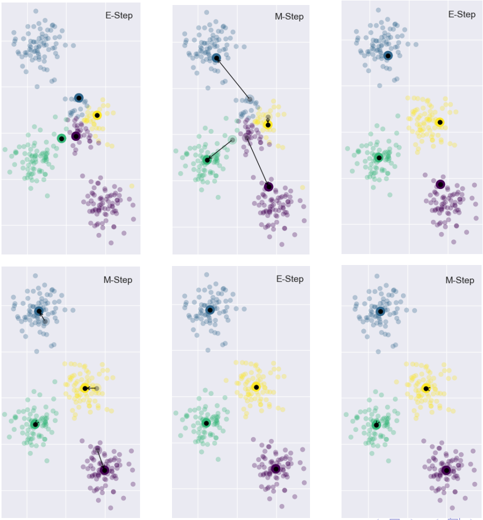
2. Repeat until converged



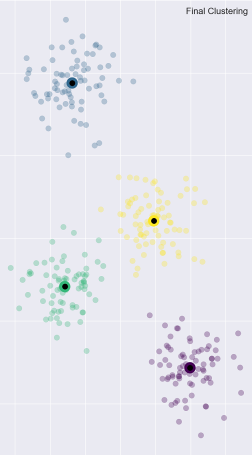
2. Repeat until converged



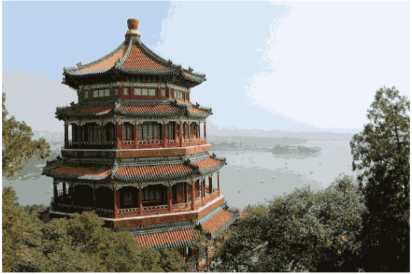
2. Repeat until converged



Final clustering



Colour compression



k-Means - Summary

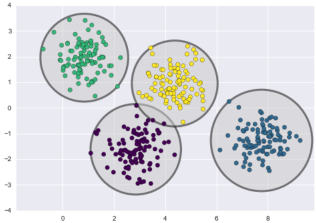
- Limited to linear cluster boundaries
- Can be slow for a large number of samples
- Lazy algorithm
  - Doesn't learn a discriminative function from training data, but memorises training data
  - In effect it means that k-means doesn't have a training step
  - With each prediction, the distances are calculated again

https://datasciencelab.wordpress.com/2013/12/12/clustering-with-k-means-in-python/

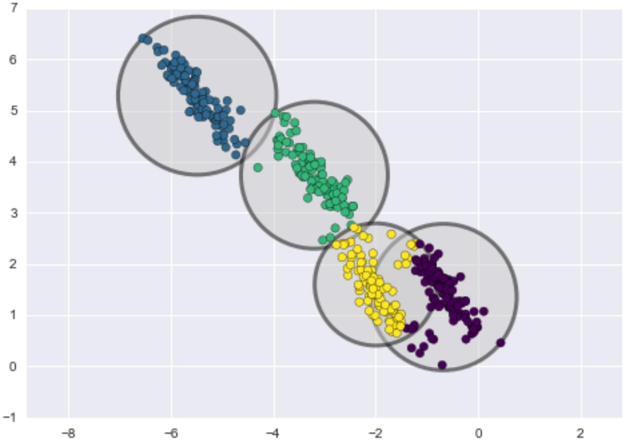
Gaussian Mixture Models (GMMs)

Motivation

- k-Means has no intrinsic measure of probability or uncertainty of cluster assignments.
- Places a circle (for 2-D) at the center of each cluster
- Radius of circle acts as a hard cutoff for cluster assignment within the training set
- any point outside this circle is not considered a member of the cluster



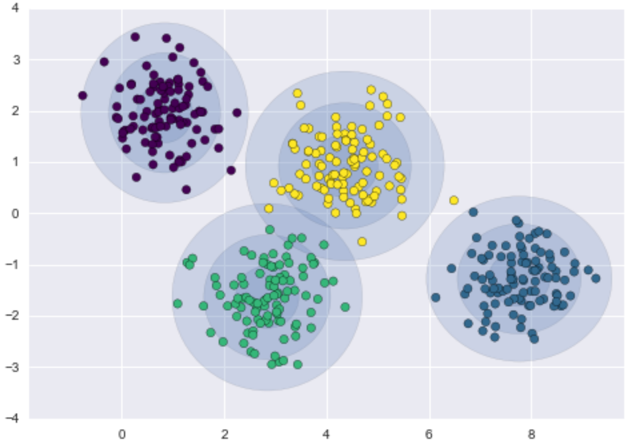
k-Means has no built-in way of accounting for elliptical clusters



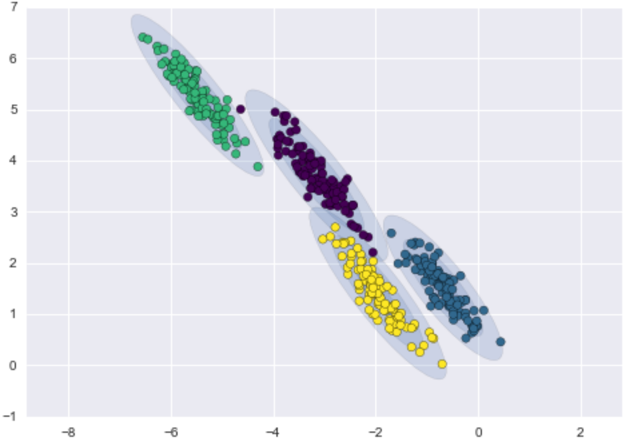
GMM as alternative

GMM

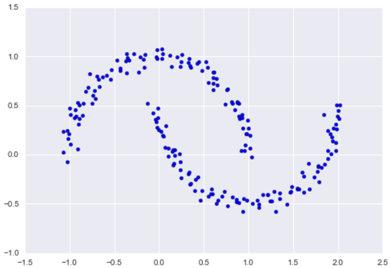
- A Gaussian mixture model (GMM) attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset
- In the simplest case, GMMs can be used for finding clusters in the same manner as k-means.
- Probabilistic in nature - ‘soft’ cluster assignments


4
3
2
1
0
−1
−2
−3
−4
0
2
4
6
8

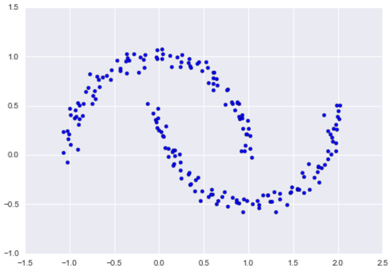
Define the covariance

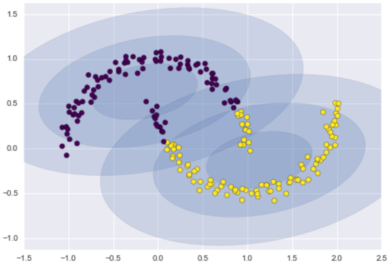


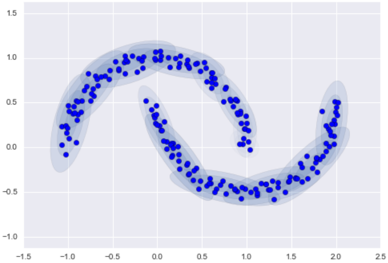
GMMs as Density Estimation

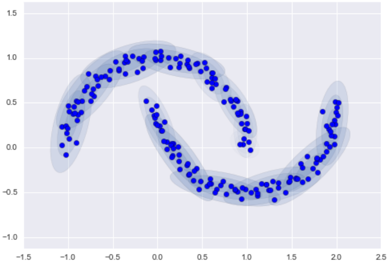


GMMs as Density Estimation



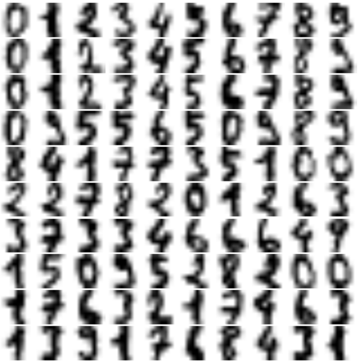



1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

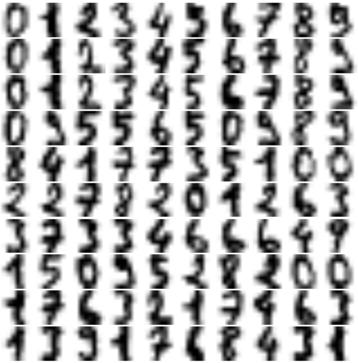


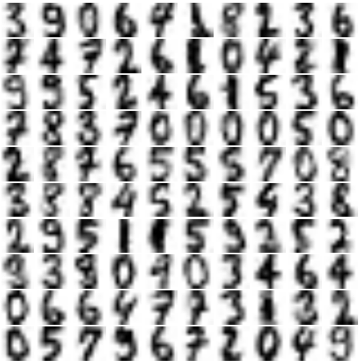
- Mixture of 16 Gaussians
- Cannot find separated clusters of data
- Rather fit the overall distribution of the data
- Generative model of the distribution
- The GMM gives us the recipe to generate new random data distributed similarly to our input

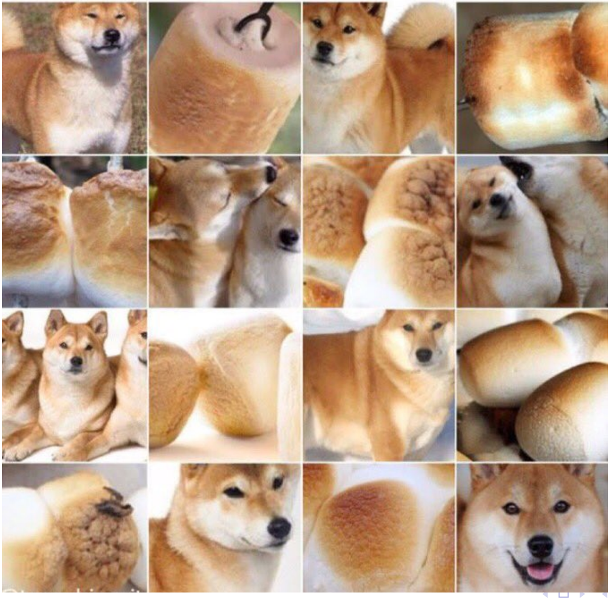
Digits dataset generated using GMM



Digits dataset generated using GMM








@teenybiscuit

Evaluation Techniques

- Generative models - likelihood of the data under the model
- Analytic criterion
  - Akaike Information Criterion (AIC)
  - Bayesian Information Criterion (BIC)
- Stability based methods

Resources

https://github.com/jakevdp/PythonDataScienceHandbook

https://rare-technologies.com/blog/

https://chrisalbon.com/

https://machinelearningflashcards.com/