K-Means Clustering

Sargur Srihari
srihari@cedar.buffalo.edu

Topics in Mixture Models and EM

- Mixture models
- K-means Clustering
- Mixtures of Gaussians
  - Maximum Likelihood
  - EM for Gaussian mistures
- EM Algorithm
  - Gaussian mixture models motivates EM
  - Latent variable viewpoint
  - K-means seen as non-probabilistic limit of EM applied to mixture of Gaussians
  - EM in generality
- Bernoulli Mixture Models
- Infinite Mixture Models

K-means Clustering

- Given data set $\{\boldsymbol{x}_1,..,\boldsymbol{x}_N\}$ in D-dimensional Euclidean space
- Partition into K clusters, which is given
- One of K coding
- Indicator variable $r_{nk} \in \{0,1\}$ where $k = 1,..,K$
  - Describes which of K clusters data point $\boldsymbol{x}_n$ is assigned to

Distortion measure

- Sum of squared errors

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

μ

- Goal is to find values for $\{r_{nk}\}$ and the $\{\mu_k\}$ so as to minimize J
  - Can be done by iterative procedure
  - Each iteration has two steps
    - Successive optimization w.r.t. r_{nk} and μ_k

Two Updating Stages

- First choose initial values for μ_k
- First phase:
  - minimize J w.r.t. r_{nk} keeping μ_k fixed
- Second phase:
  - minimize J w.r.t. μ_k keeping r_{nk} fixed
- Two stages correspond to E $(\mathrm{expectation})$ and M $(\mathrm{maximization})$ of EM algorithm
  - Expectation: what is the expected class?
  - Maximization: what is the mle of the mean?

E: Determination of Indicator r_{nk}

- Because $$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| \boldsymbol{x}_n - \boldsymbol{\mu}_k \|^2$$
  is a linear function of r_{nk} this optimization is performed easily (closed form solution)
- Terms involving different n are independent
  - Therefore can optimize for each n separately
  - Choosing r_{nk} to be 1 for whichever value of k gives minimum value of $||\boldsymbol{x}_n - \mu_k||^2$
- Thus
$$r_{nk} = \begin{cases} 1 \text{ if } k = \arg\min_j || \boldsymbol{x}_n - \mu_j ||^2 \\ 0 \quad \text{otherwise} \end{cases}$$
- Interpretation:
  - Assign $\boldsymbol{x}_n$ to cluster whose mean is closest

M: Optimization of μ_k

- Hold r_{nk} fixed
- Objective function $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| \boldsymbol{x}_n - \mu_k \|^2$ is a quadratic function of μ_k
- Minimized by setting derivative w.r.t. μ_k to zero
- Thus $2\sum_{n=1}^{N} r_{nk}(\boldsymbol{x}_n - \mu_k) = 0$
- Which is solved to give $$\mu_k = \frac{\sum_n r_{nk}\boldsymbol{x}_n}{\sum_n r_{nk}}$$

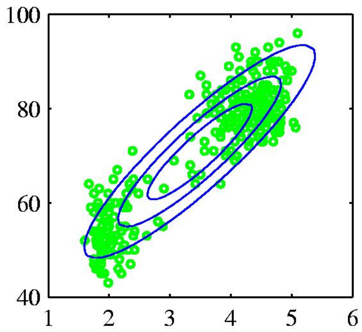
  Equal to no of points assigned to cluster k
- Interpretation:
  - Set μ_k to mean of all data points $\boldsymbol{x}_n$ assigned to cluster k

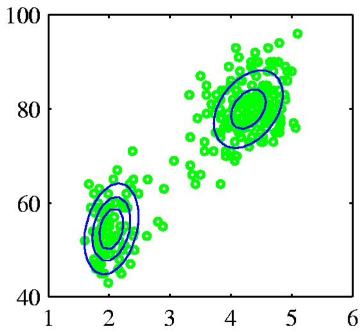
Termination of K-Means

- Two phases
  - re-assigning data points to clusters
  - Re-computing means of clusters
- Done repeatedly until no further change in assignments
- Since each phase reduces J convergence is assured
- May converge to local minimum of J

Illustration of K-means

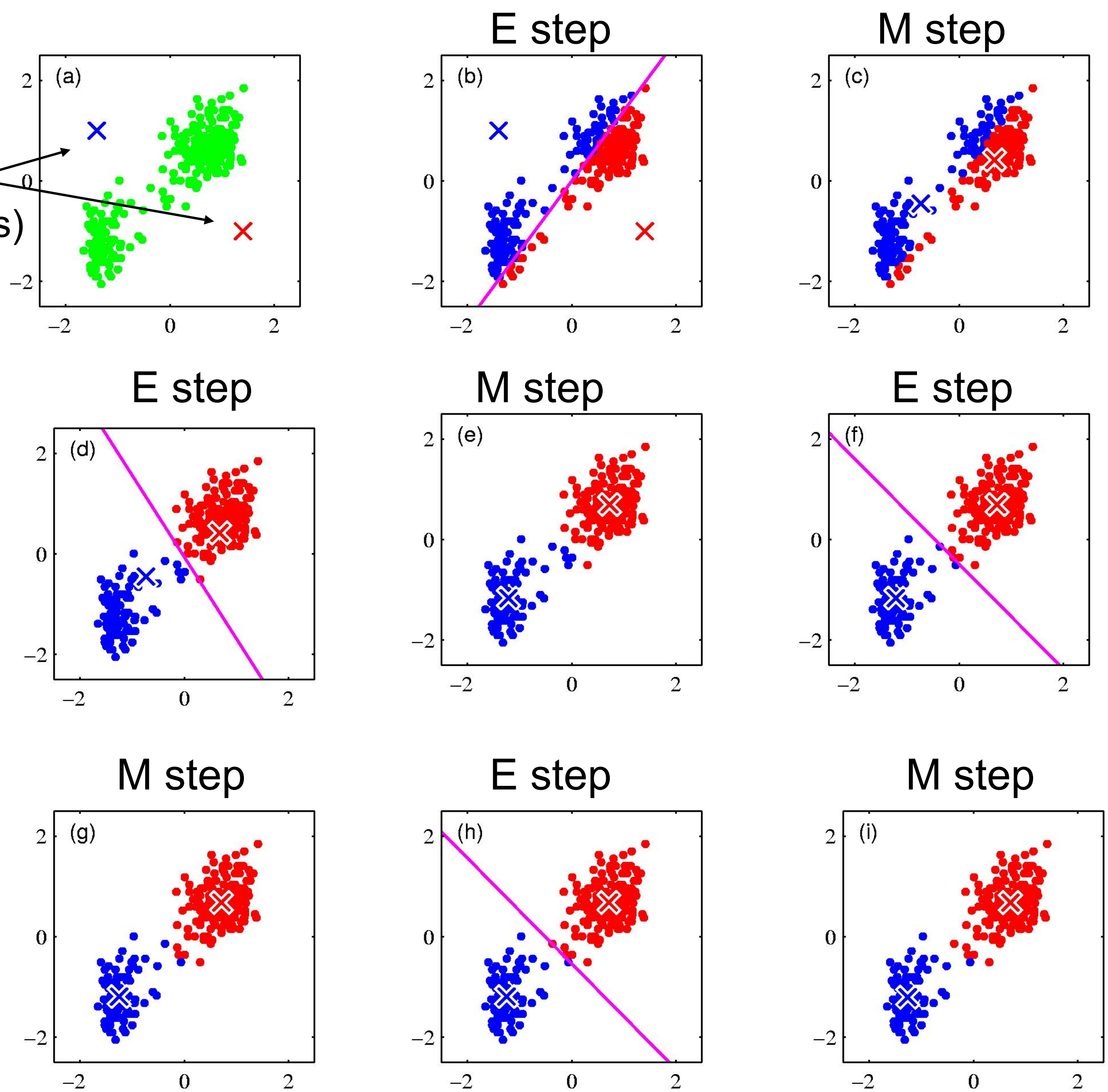
- Old Faithful dataset
- Single Gaussian is a poor fit
- We choose $K = 2$
- Data set is standardized so each variable has zero mean and unit standard deviation
- Assignment of each data point to nearest cluster center is equivalent to
  - which side of the perpendicular bisector of line joining cluster centers





K-means iterations

Initial Choice of Means (Parameters)

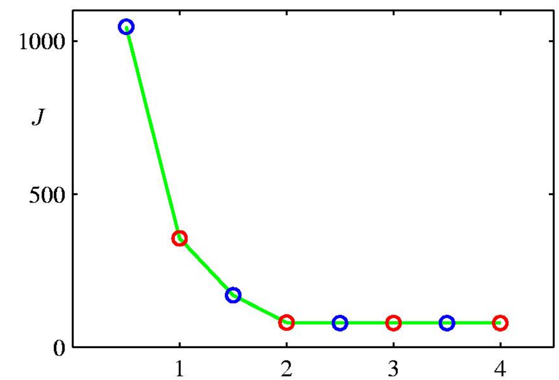


E step: parameters are fixed Distributions are optimized

M step: distributions are fixed Parameters are optimized

Final Clusters And Means

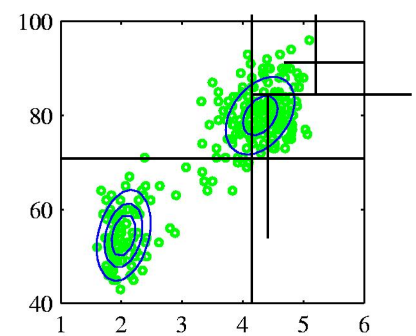
Cost Function after Iteration



- J for Old Faithful Data
- Poor initial value chosen for cluster centers
  - Several steps needed for convergence
  - Better choice is to assign μ_k to random subset of k data points
- $K\text{-means}$ is itself used to initialize parameters for Gaussian mixture model before applying EM

Faster Implementation of K-means

- Direct implementation can be slow
  - In E step Euclidean distances are computed between every mean and every data point
    - $||\boldsymbol{x}_n - \mu_k||^2$ is computed for $n=1,..N$ and $k=1,..K$
- Faster implementations exist
  - Precomputing trees where nearby points are on same sub-tree
  - Use of triangle inequality to avoid unnecessary distance calculation



On-line Stochastic Version

- Instead of batch processing entire data set
- Apply Robbins-Monro procedure
  - To finding roots of the regression function given by the derivative of J w.r.t μ_k

$$\mu_k^{new} = \mu_k^{old} + \eta_n \left(\boldsymbol{x}_n - \mu_k^{old} \right)$$

  - *where* η_n is a learning rate parameter made to decrease monotonically as more samples are observed

Dissimilarity Measure

- Euclidean distance has limitations
  - Inappropriate for categorical labels
  - Cluster means are non-robust to outliers
- Use more general dissimilarity measure $\nu(\boldsymbol{x},\boldsymbol{x}')$ and distortion measure

$$\tilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \nu(\boldsymbol{x}_n, \mu_k)$$

- Which gives the k-medoids algorithm
- M-step is potentially more complex than for k-means

Limitation of K-means

- Every data point is assigned uniquely to one and only one cluster
- A point may be equidistant from two cluster centers
- A probabilistic approach will have a 'soft' assignment of data points reflecting the level of uncertainty

Image Segmentation and Compression

- Goal: partition image into regions
  - each of which has homogeneous visual appearance
  - or corresponds to objects
  - or parts of objects
- Each pixel is a point in R_G_B space
- K-means clustering is used with a palette of K colors
- Method does not take into account proximity of different pixels

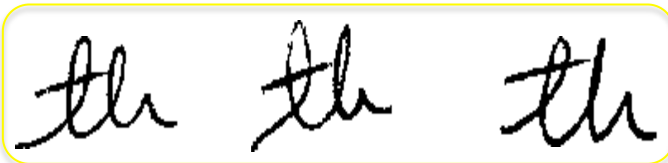
K-means in Image Segmentation

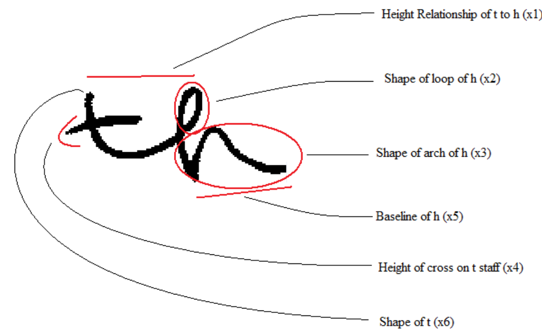
Two examples where 2, 3, and 10 colors are chosen to encode a color image

Data Compression

- Lossless data compression
  - Able to reconstruct data exactly from compressed representation
- Lossy data compression
  - Accept some error in return for greater compression
- K-means for Lossy compression
  - For each of N data points store only identity k of cluster center to which it is assigned
  - Store values of cluster centers μ_k where $K<<N$
  - Vectors $\mathbf{m}_k$ are called *code-book vectors*
  - Method is called *Vector Quantization*
  - Data compression achieved
    - Original image needs $24N$ bits (R,G,B need 8 bits each)
    - Compressed image needs $24K+N\log_2\text{K}$ bits
    - For $K=2,3$ and 10, compression ratios are $4\%,8\%$ and 17%

Data Set: Handwriting Style Features





Meuhlberger, et. al.
Journal of Forensic Sciences, 1975

R = **Height Relationship of *t* to *h***	L = **Shape of Loop of *h***	A = **Shape of Arch of *h***	C = **Height of Cross on *t* staff**	B = **Baseline of *h***	S = **Shape of *t***
r^0 = *t* shorter than *h*	l^0 = retraced	a^0 = rounded arch	c^0 = upper half of staff	b^0 = slanting upward	s^0 = tented
r^1 = *t* even with *h*	l^1 = curved right side and straight left side	a^1 = pointed	c^1 = lower half of staff	b^1 = slanting downward	s^1 = single stroke
r^2 = *t* taller than *h*	l^2 = curved left side and straight right side	a^2 =no set pattern	c^2 = above staff	b^2 = baseline even	s^2 = looped
r^3 = no set pattern	l^3 = both sides curved		c^3 = no fixed pattern	b^3 = no set pattern	s^3 = closed
	l^4 = no fixed pattern				s^4 = mixture of shapes

$$|Val(X)| = 4 \text{ x } 5 \text{ x } 3 \text{ x } 4 \text{ x } 4 \text{ x } 5 = 4{,}800$$

$P(R,L,A,C,B,T)$

No of parameters = $4{,}799$

Clustering of handwriting styles

- Letters “th” characterized by six multinomial features