Clustering

Shadi Albarqouni, M.Sc.
Graduate Research Assistant | PhD Candidate
shadi.albarqouni@tum.de

Outline

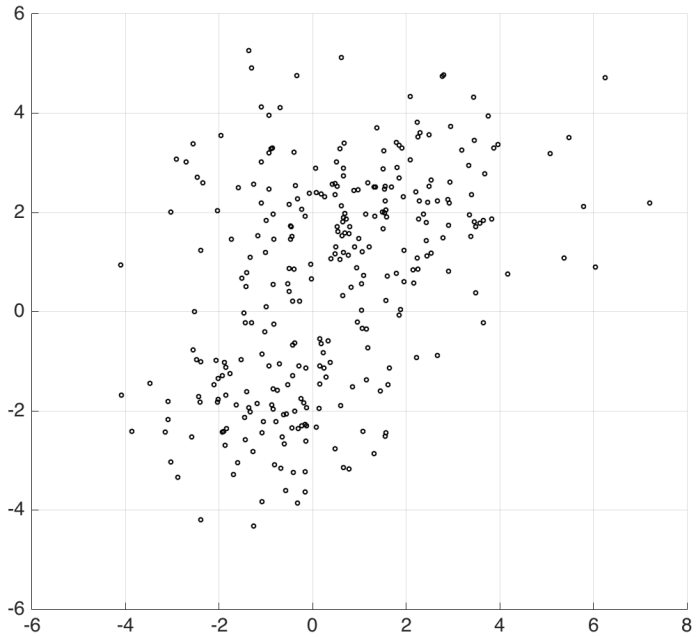
What is clustering?

Definition (Clustering)

Given n unlabelled data points, separate them into K clusters.

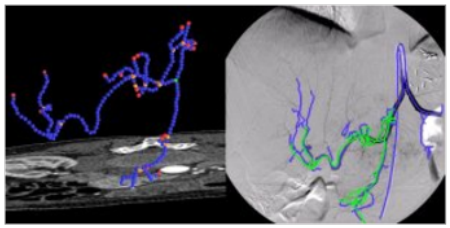
Dilemma! [6]

- What is a Cluster?
  (Compact vs. Connected)
- How many K clusters?
  (Parametric vs.
  Non-parametric)
- Soft vs. Hard clustering.
  (Model vs. Cost based)
- Data representation.
  (Vector vs. Similarities)
- Classification vs. Clustering.
- Stability [7].



Applications

- Image Retrieval
- Image Compression
- Image Segmentation
- Pattern Recognition



Notation

- $\mathcal{X}^T = \{x_1, x_2, ..., x_N\} \in \mathbb{R}^{d \times N}$ is the data set.
- d is the feature dimension of x_i.
- N is the number of instances.
- K is the number of clusters.
- $\nabla = \{C_1, C_2, ..., C_K\}$, where C_k is a partition of $\mathcal{X}$.
- $c(x_i)$ is the label/cluster of instance x_i.
- r_{nk} where n is the index of instance and k is the index of cluster.

Objective

Find the clusters ∇ minimizing the cost function $\mathcal{L}(\nabla)$.

Parametric, cost-based clustering

Parametric: K is defined.
Cost-based: It is hard-clustering based on the cost function.
Selected Algorithms:

- K-Means [8].
- K-Medoids [11].
- Kernel K-Means [12].
- Spectral Clustering [10].

K-Means

- K-Means algorithm:
  1. **Initialize:** Pick K random samples from the dataset $\mathcal{X}^T$ as the cluster centroids $\mu_k = \{\mu_1, \mu_2, ..., \mu_K\}$.
  2. **Assign Points to the clusters**: Partition data points $\mathcal{X}^T$ into K clusters $\nabla = \{C_1, C_2, ..., C_K\}$ based on the Euclidean distance between the points and centroids (searching for the closest centroid).
  3. **Centroid update**: Based on the points assigned to each cluster, a new centroid is computed μ_k.
  4. **Repeat**: Do step 2 and 3 until convergence.
  5. **Convergence**: if the cluster centroids barley change, or we have compact and/or isolated clusters. Mathematically, when the cost (distortion) function $\mathcal{L}(\nabla) = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2$ is minimum.
- **Practical issues**:
  a) The initialization. b) Pre-processing.

K-Means – Algorithm

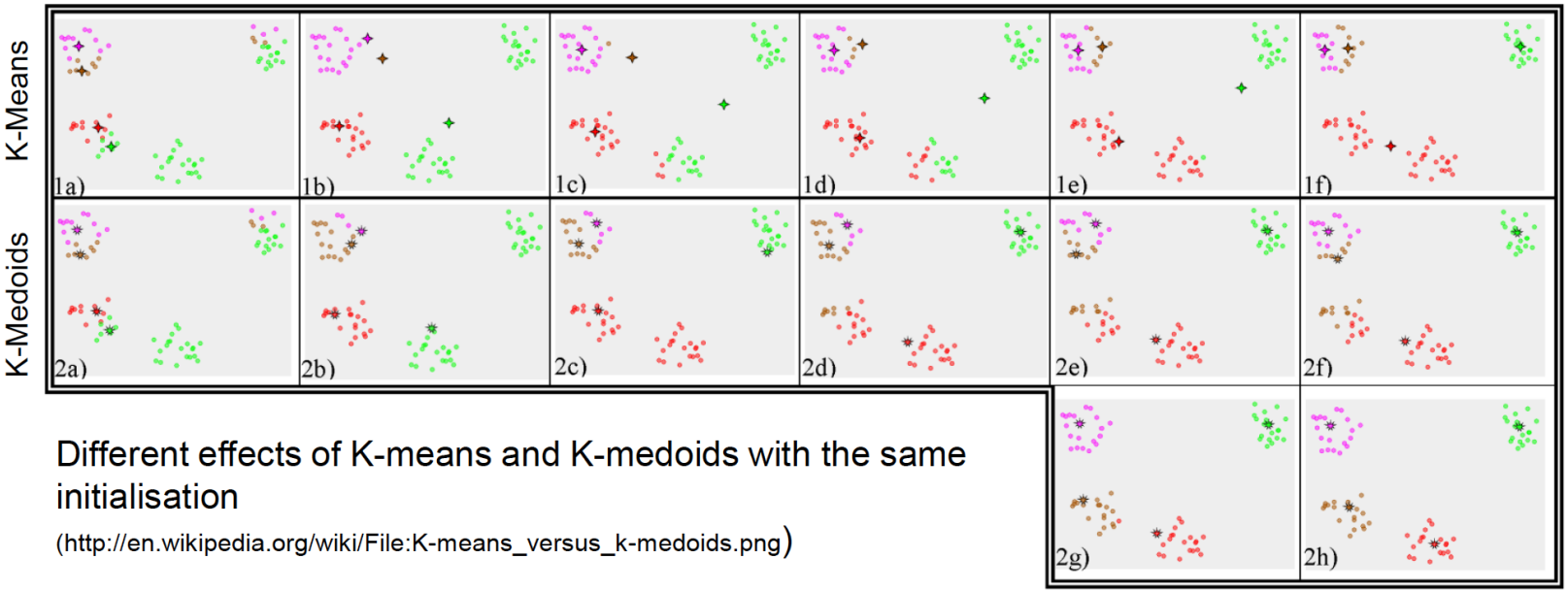
input : Data points $\mathcal{X}^T = \{x_1, x_2, ..., x_N\}$, number of clusters K
output: Clusters, $\nabla = \{C_1, C_2, ..., C_K\}$

Pick K random samples as the cluster centroids μ_k.
repeat
  for $i = 1$ **to** N **do**
    $c(x_i) = \min_{k \in K} \|x_i - \mu_k\|_2^2$ %Assign points to clusters
  end
  for $k = 1$ **to** K **do**
    $\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$ %Update the cluster centroid
  end
until *convergence*;

K-Medoids I

- K-Medoids algorithm:
  1. **Initialize:** Pick K random samples from the dataset $\mathcal{X}^T$ as the medoids $\mu_k = \{\mu_1, \mu_2, ..., \mu_K\}$.
  2. **Assign Points to the clusters**: Partition data points $\mathcal{X}^T$ into K clusters $\nabla = \{C_1, C_2, ..., C_K\}$ based on the dissimilarity (Manhattan) distance between the points and medoids (searching for the min. dissimilarity).
  3. **Medoids update**: Based on the points assigned to each cluster, swap the medoid with a new data point and compute the cost. (undo the swap if the cost is getting increased).
  4. **Repeat**: Do step 2 and 3 until convergence.
  5. **Convergence**: if the cluster medoids barley change. Mathematically, when the cost (distortion) function $\mathcal{L}(\nabla) = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|$ is minimum.

K-Medoids II



Different effects of K-means and K-medoids with the same initialisation

(http://en.wikipedia.org/wiki/File:K-means_versus_k-medoids.png)

Figure : K-Means vs. K-Medoids

Kernel K-Means I

Definition

It is a generalization of the standard K-Means algorithm.

- What happens if the clusters are not linearly separable?
- Euclidean distance vs. Geodesic distance.

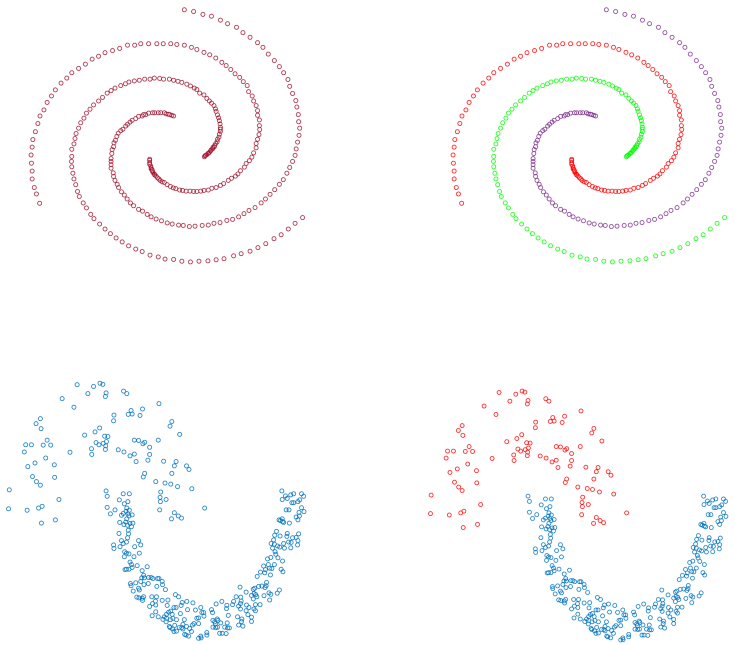


Figure : Spiral and Jain datasets

Kernel K-Means II

- K-Means can not be applied right away.
- Map the data points $x_i \in \mathcal{X}^T$ to a high dimensional feature space $\mathcal{M}$ using a nonlinear function $\phi(x_i)$.
- Assume the clusters in the high dimensional feature space (RKHS) is linearly separable, hence K-Means can be applied.
- The cost function would be

$$\mathcal{L}_{\mathcal{K}}(\nabla) = \sum_{k=1}^{K} \sum_{i \in C_k} \|\phi(x_i) - \phi(\mu_k)\|^2 ,$$

where $\|\phi(x_i) - \phi(\mu_k)\|^2 =$
$\phi(x_i)^T \cdot \phi(x_i) - 2\phi(x_i)^T \cdot \phi(\mu_k) + \phi(\mu_k)^T \cdot \phi(\mu_k)$.

Kernel K-Means III

- Using the kernel trick, $K_{ij} = \phi(x_i)^T \cdot \phi(x_j)$, the Euclidean distance in $\mathcal{L}_{\mathcal{K}}(\nabla)$ can be computed easily using any kernel function K_{ij} without explicitly knowing the nonlinear transformation $\phi(x_i)$.
- Examples of kernel functions (positive semidefinite)
  1. Hom. Polynomial kernel: $K_{ij} = (x_i^T x_j)^{\delta}$
  2. Inho. Polynomial kernel: $K_{ij} = (x_i^T x_j + \gamma)^{\delta}$
  3. Gaussian kernel: $K_{ij} = \mathrm{e}^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}$
  4. Laplacian kernel $K_{ij} = \mathrm{e}^{\frac{-\|x_i - x_j\|}{\sigma}}$
  5. Sigmoid kernel: $K_{ij} = \tanh(\gamma(x_i^T x_j) + \theta)$

Kernel K-Means – Algorithm

input : Data points $\mathcal{X}^T = \{x_1, x_2, ..., x_N\}$, Kernel matrix K_{ij}, number of clusters K

output: Clusters, $\nabla = \{C_1, C_2, ..., C_K\}$

Pick K random samples as the cluster centroids μ_k.

repeat

  for $i = 1$ **to** N **do**

    for $k = 1$ **to** K **do**

      Compute $\|\phi(x_i) - \phi(\mu_k)\|^2$ using K_{ij}

    end

    $c(x_i) = \min_{k \in K} \|\phi(x_i) - \phi(\mu_k)\|^2$

  end

  for $k = 1$ **to** K **do**

    $\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$

  end

until *convergence*;

Spectral Clustering

Graph - Overview

- Fully connected, undirected, and wighted graph with N vertices
- The graph is represented by $G = \{\nu, \varepsilon, \omega\}$, where ν is a set of vertices N, ε is a set of edges, and ω is a set of weights are assigned using a heat kernel as follows to build the Adjacency matrix W

$$W_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|_2^2}{\sigma^2}} & e_{ij} \in \varepsilon \\ 0 & else \end{cases}$$

- The degree matrix D, where its diagonal elements $D_{ij} = \sum_j W_{ij}$
- Compute the Normalized graph Laplacian Matrix

$$\tilde{\mathcal{L}} = I - D^{-1/2} W D^{-1/2}$$

Spectral Clustering – Algorithm

input : Normalized Laplacian Matrix $\tilde{\mathcal{L}}$, number of clusters K
output: Clusters, $\nabla = \{C_1, C_2, ..., C_K\}$

Compute the firsts K eigenvectors $U = \{u_1, u_2, ..., u_K\} \in \mathbb{R}^{n \times K}$ of $\tilde{\mathcal{L}}$.
Compute $\tilde{U}$ by normalising the rows to norm 1.
Do K-Means on $\tilde{U} \in \mathbb{R}^{n \times K}$ such that your data points are the rows vectors which have K-dimensions or simply: $\mathcal{D} \leftarrow \tilde{U}^T$.
Pick K random samples as the cluster centroids μ_k.
repeat
  for $i = 1$ **to** n **do**
    $c(x_i) = \min_{k \in K} \|x_i - \mu_k\|_2^2$ %Assign points to clusters
  end
  for $k = 1$ **to** K **do**
    $\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$ %Update the cluster centroid
  end
until *convergence*;

Extensions I

- Alternative cost (distortion) function:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\|x_i - x_j\|^2 = \underbrace{\sum_{k=1}^{K}\sum_{i,j\in C_k}\|x_i - x_j\|^2}_{\text{Intracluster distance}} + \underbrace{\sum_{k=1}^{K}\sum_{i\in C_k}\sum_{j\notin C_k}\|x_i - x_j\|^2}_{\text{Intercluster distance}}$$

1. Intracluster distance:

$$\mathcal{L}(\nabla) = \sum_{k=1}^{K}\sum_{i,j\in C_k}\|x_i - x_j\|^2 + constant$$

2. Interclsuter distance:

$$\mathcal{L}(\nabla) = -\sum_{k=1}^{K}\sum_{i\in C_k}\sum_{l\notin C_k}\|x_i - x_j\|^2 + constant$$

Extensions II

3. K-Median:

$$\mathcal{L}(\nabla) = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|$$

- Alternative Initialization:
  1. K-Means++ [1]
  2. Global Kernel K-Means [13]
- On selecting K [1]:
  1. Rule of thumb: $K = \sqrt{N/2}$
  2. Elbow Method
  3. Silhouette
- Soft clustering: Fuzzy C-Means [2]
- Variant: Spectral Clustering [14]
- Hierarchical Clustering

[1] https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set

Comparison

Algorithm	Data Rep.	Comp.	Out.	Cent.
K-Means	Vectors	Low	No	$\notin \mathcal{X}^T$
K-Medians	Vectors	High	No	$\notin \mathcal{X}^T$
K-Medoids	Similarity	High	Yes	$\in \mathcal{X}^T$
Kernel K-Means	Kernel	High	N/A	$\notin \mathcal{X}^T$
Spectral Clustering	Similarity	High	N/A	$\notin \mathcal{X}^T$

[2]

[2] **Data Rep:** Data Representation, **Comp.:** Computational cost, **Out.:** Handling outliers, **Cent.:** Centroids.

Parametric, model-based clustering

Parametric: K and the density function are defined (i.e. Gaussian)
Model-based: It is soft-clustering based on the mixture density $f(x)$.

$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x), \qquad s.t. \qquad \pi_k \geq 0, \sum_{K} \pi_k = 1,$$

where $f_k(x)$ is the component of mixture. $f(x)$ is a Gaussian Mixture Model (GMM) when $f_k(x) \sim \mathcal{N}(x; \mu_k, \sigma_k^2)$.
Degree of Membership:

$$\gamma_{ki} = P[x_i \in C_k] = \frac{\pi_k f_k(x_i)}{f(x_i)}$$

GMM Parameter: $\theta = \{\pi_{1:K}, \mu_{1:K}, \sigma_{1:K}\}$.
Selected Algorithm to estimate the parameter: EM-Algorithm [5].

Expectation-Maximization (EM) Algorithm

- Given data points $\mathcal{X}^T$ sampled i.i.d from an unknown distribution f
- We need to model the distribution using Maximum Likelihood (ML) principle (log-likelihood):

$$l(\theta) = \ln f_\theta(\mathcal{X}) = \sum_{i=1}^{N} \ln f_\theta(x_i)$$

$$l(\theta) = \sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k f_k(x_i)$$

The objective:
$\theta^{ML} = argmax_\theta l(\theta)$

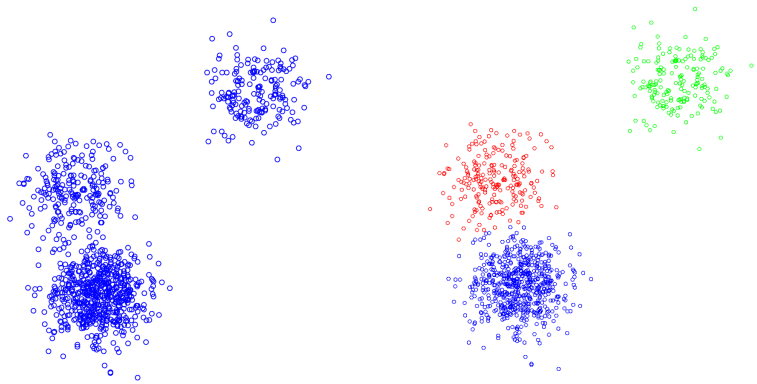


Figure : GMM Clustering

EM – Algorithm

input : data points $\mathcal{X}^T$, number of clusters K
output: Parameters, $\theta^{ML} = \{\pi_{1:K}, \mu_{1:K}, \sigma_{1:K}\}$

Initialize the parameters θ at random.
repeat
  for $i = 1$ **to** N **do**
    for $k = 1$ **to** K **do**
      $\gamma_{ik} = \frac{\pi_k f_k(x_i)}{f(x_i)}$ %E-Step
    end
  end
  for $k = 1$ **to** K **do**
    $\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}$ %M-Step
    $\mu_k = \frac{1}{N\pi_k} \sum_{i=1}^{N} \gamma_{ik} x_i$
    $\sigma_k = \frac{1}{N\pi_k} \sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$
  end
until *convergence*;

Non-parametric, model-based clustering

Idea: group the points by the peak of data density
Parameter: shape and number of clusters K are defined by the algorithm, however, you should define:

1. smoothness of density estimate (h)[3]
2. what is a peak

Selected Algorithm:

- Mean-shift [4].

[3] Rule of thumb: $h = (\frac{4\hat{\sigma}}{3n})^{1/5}$, where $\hat{\sigma}$ is the standard deviation of the samples.

Mean-shift Algorithm

- Given data points $\mathcal{X}^T$ sampled i.i.d from an unknown density f
- We need to define the shape of the density using Kernel Density Estimation (KDE) principle:

$$f_h(x) = \frac{1}{Nh^d} \sum_{i=1}^{N} K(\frac{x - x_i}{h}),$$

where $K(\cdot)$ is a kernel function, must be positive, symmetric and differentiable, i.e. Gaussian kernel $K(z) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{\|z\|^2}{2}}$

- The objective: find the peaks of $f_h(x)$ by equating $\nabla f_h(x) = 0$
- That results in

$$x = \underbrace{\frac{\sum_{i=1}^{N} x_i K(\frac{x-x_i}{h})}{\sum_{i=1}^{N} K(\frac{x-x_i}{h})}}_{mean-shift}$$

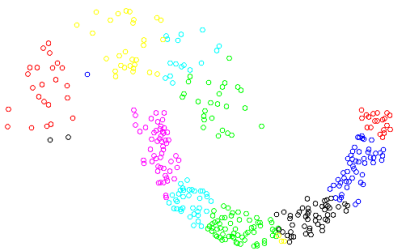
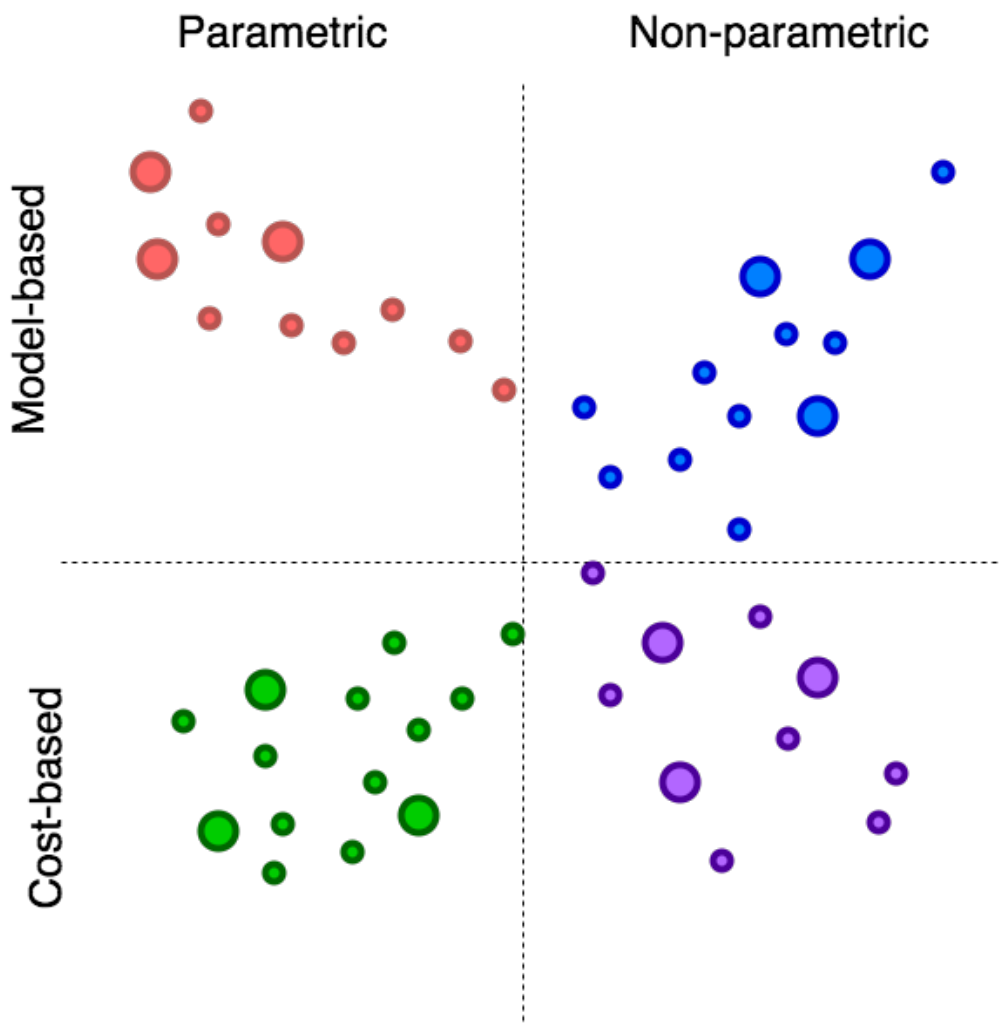


Figure : Mean-shift Clustering

Summary



Acknowledgment

This tutorial is done with the help of

- Bishop's book [3],
- Meila's slides in MLSS 2011 [9], and
- Lichao's slides form summer semester (SS15)

References I

David Arthur and Sergei Vassilvitskii.

k-means++: The advantages of careful seeding.

In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

James C Bezdek.

Pattern recognition with fuzzy objective function algorithms.

Springer Science & Business Media, 2013.

Christopher M Bishop.

Pattern recognition and machine learning.

springer, 2006.

Yizong Cheng.

Mean shift, mode seeking, and clustering.

Pattern Analysis and Machine Intelligence, IEEE Transactions on, 17(8):790–799, 1995.

Arthur P Dempster, Nan M Laird, and Donald B Rubin.

Maximum likelihood from incomplete data via the em algorithm.

Journal of the royal statistical society. Series B (methodological), pages 1–38, 1977.

Anil K Jain and Martin HC Law.

Data clustering: A user's dilemma.

In *Pattern Recognition and Machine Intelligence*, pages 1–10. Springer, 2005.

Tilman Lange, Volker Roth, Mikio L Braun, and Joachim M Buhmann.

Stability-based validation of clustering solutions.

Neural computation, 16(6):1299–1323, 2004.

References II

S. Lloyd.

Least squares quantization in pcm.
Information Theory, IEEE Transactions on, 28(2):129–137, Mar 1982.

Marina Meila.

Classic and modern data clustering.
Machine Learning Summer School (MLSS), 2011.

Andrew Y Ng, Michael I Jordan, Yair Weiss, et al.

On spectral clustering: Analysis and an algorithm.
Advances in neural information processing systems, 2:849–856, 2002.

Hae-Sang Park and Chi-Hyuck Jun.

A simple and fast algorithm for k-medoids clustering.
Expert Systems with Applications, 36(2):3336–3341, 2009.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller.

Nonlinear component analysis as a kernel eigenvalue problem.
Neural computation, 10(5):1299–1319, 1998.

Grigorios F Tzortzis and Aristidis C Likas.

The global kernel-means algorithm for clustering in feature space.
Neural Networks, IEEE Transactions on, 20(7):1181–1194, 2009.

Ulrike Von Luxburg.

A tutorial on spectral clustering.
Statistics and computing, 17(4):395–416, 2007.