Machine Learning in Robotics
Lecture 4: Unsupervised Clustering

Prof. Dongheui Lee

Institute of Automatic Control Engineering
Technische Universität München

dhlee@tum.de

Today Lecture Outline

- Unsupervised Clustering
- Similarity measures
- Criterion functions
- Iterative optimization algorithms
- K-means & Variations
- Hierarchical clustering

Supervised vs. Unsupervised learning

- Supervised learning
  - A pattern is a pair of variables $\{\boldsymbol{x}, \omega\}$ where $\boldsymbol{x}$ is a collection of observations or features (feature vector) and ω is the concept behind the observation (label)
- Unsupervised learning
  - Use unlabeled data, a collection of feature vectors without the class label ω

Supervised vs. Unsupervised learning

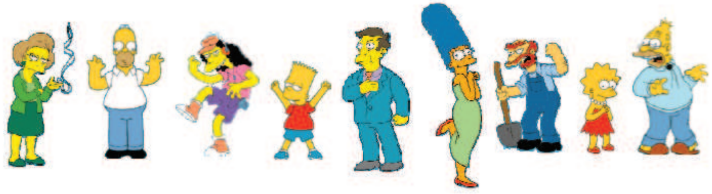
- Supervised learning
  - A pattern is a pair of variables $\{\boldsymbol{x}, \omega\}$ where $\boldsymbol{x}$ is a collection of observations or features (feature vector) and ω is the concept behind the observation (label)
- Unsupervised learning
  - Use unlabeled data, a collection of feature vectors without the class label ω
  - These methods are called unsupervised because they are not provided the correct answer

Supervised vs. Unsupervised learning

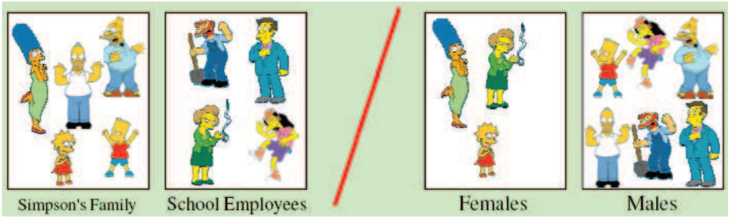
- Supervised learning
  - A pattern is a pair of variables $\{\boldsymbol{x}, \omega\}$ where $\boldsymbol{x}$ is a collection of observations or features (feature vector) and ω is the concept behind the observation (label)
- Unsupervised learning
  - Use unlabeled data, a collection of feature vectors without the class label ω
  - These methods are called unsupervised because they are not provided the correct answer
  - Unsupervised methods may appear to have limited capabilities, but they are useful
    - Labeling large data sets can be a costly procedure
    - Class labels may not be known beforehand
    - Large datasets can be compressed by finding a small set of proto-types

TEST : Unsupervised clustering

What is the natural grouping among those objects?



Many possibilities!!!



Two approaches for unsupervised learning

- Parametric approaches
  - Functional forms for the underlying class-conditional densities are assumed, and we must estimate the parameters

$$p(\boldsymbol{x}|\theta) = \sum_{i=1}^{K} p(x|\omega_i, \theta_i)p(\omega_i)$$

Two approaches for unsupervised learning

- Parametric approaches
  - Functional forms for the underlying class-conditional densities are assumed, and we must estimate the parameters

$$p(\boldsymbol{x}|\theta) = \sum_{i=1}^{K} p(x|\omega_i, \theta_i)p(\omega_i)$$

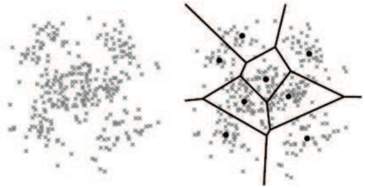
- Non-parametric approaches
  - No assumptions are made about the underlying densities
  - Instead, seek a partition of the data into clusters
  - These methods are typically referred to as clustering

Vector Quantization (VQ)

- Vector Quantization is a lossy data compression method
- Mapping n feature vectors $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}\}$ to K classes of feature vectors $\boldsymbol{Y} = \{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(K)}\}$

$$\boldsymbol{y}^{(j)} = c(\boldsymbol{x}^{(i)}), \quad j = 1, \ldots, K; \quad i = 1, \ldots, n; \quad K < n$$

- Code vector $\boldsymbol{y}^{(j)}$
  - Cluster centers
  - Points with high density in feature space
- Code book $\boldsymbol{Y} = \{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(K)}\}$
  - The set of all code vectors



Nonparametric clustering

Nonparametric clustering involves three steps:

- Defining a measure of (dis)similarity between examples
- Defining a criterion (Distortion) function for clustering
- Defining an algorithm to minimize (or maximize) a criterion (distortion) function

Similarity Measures

- A measuring rule of $d(\boldsymbol{x},\boldsymbol{y})$ for the distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is considered a metric if it satisfies the following properties.
  - non-negativity: $d(\boldsymbol{x},\boldsymbol{y}) \geq 0$
  - reflexivity: $d(\boldsymbol{x},\boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$
  - symmetry: $d(\boldsymbol{x},\boldsymbol{y}) = d(\boldsymbol{y},\boldsymbol{x})$
  - triangle inequality: $d(\boldsymbol{x},\boldsymbol{y}) + d(\boldsymbol{y},\boldsymbol{z}) \geq d(\boldsymbol{x},\boldsymbol{z})$

Similarity Measures

- The most general form of distance metric is the power norm

$$d(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{m} |x_i - y_i|^q \right)^{1/q}$$

  - where $q \geq 1$ is a selectable parameter - general Minkowski metric
  - When $q = 2$, Euclidean metric $d(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T(\boldsymbol{x}-\boldsymbol{y})}$
  - When $q = 1$, Manhattan or city block metric - sum of the absolute distances along each of the m coordinate axes.
- Notice that the above distance metrics are measures of dissimilarity.

Similarity Measures

- Inner product

$$s(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$$

- When the features are binary-valued (0 or 1), the normalized inner product is a measure of the relative possession of common attributes.
- One variation is Tanimoto distance, a ratio of the number of shared attributes to the number possessed by $\boldsymbol{x}$ or $\boldsymbol{y}$.

$$s(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\boldsymbol{x}^T\boldsymbol{x} + \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{x}^T\boldsymbol{y}}$$

Similarity measure

Criterion (Distortion) function

- Once a (dis)similarity measure has been determined, we need to define a criterion function to be optimized.
- This function measures the clustering quality of any partition of the data.
  - The most widely used criterion function for clustering is the *sum-of-square-error*

$$J = \sum_{i=1}^{K} \sum_{\boldsymbol{x} \in \omega_i} (\boldsymbol{x} - \boldsymbol{y}^{(i)})^2 \quad \text{where} \ \boldsymbol{y}^{(i)} = \frac{1}{N_i} \sum_{\boldsymbol{x} \in \omega_i} \boldsymbol{x}$$

    Here, N_i is the number of samples in the ω_i
  - This criterion measures how well the data set is represented by the cluster centers
- Other criterion functions exist, based on the scatter matrices used in Linear Discriminant Analysis (LDA).
  - Trace criterion $tr[\boldsymbol{S}_w]$
  - Determinant criterion $|\boldsymbol{S}_w|$
  - Invariant criterion $tr[\boldsymbol{S}_T^{-1}\boldsymbol{S}_w]$

Iterative optimization

- Once a criterion function has been defined, we must find a partition of the data set that minimizes the criterion.
- Exhaustive enumeration of all partitions, which guarantees the optimal solution, is unfeasible.
- Common approach is to proceed in an iterative fashion
  - Find reasonable initial partition
  - Move samples from one cluster to another in order to reduce the criterion function
- These iterative methods produce sub-optimal solution but are computationally tractable

Iterative optimization

- Once a criterion function has been defined, we must find a partition of the data set that minimizes the criterion.
- Exhaustive enumeration of all partitions, which guarantees the optimal solution, is unfeasible.
- Common approach is to proceed in an iterative fashion
  - Find reasonable initial partition
  - Move samples from one cluster to another in order to reduce the criterion function
- These iterative methods produce sub-optimal solution but are computationally tractable
- Approaches
  - K-means algorithm
  - LBG algorithm
  - Fuzzy K-means algorithm
  - Basic iterative minimum-squared-error clustering

k-means, EM algorithm

- EM algorithm
  - Useful when estimating an optimal solution of a problem including hidden information
  - An iterative method which alternates between performing an expectation (E) step and a maximization (M) step
  - E step: computes the expectation of the log-likelihood evaluated using the current estimate for the latent variables
  - M step: computes parameters maximizing the expected log-likelihood found on the E step
  - Local convergence, No guarantee for global convergence
- K-means
  - A simple example of the EM optimization algorithm

k-means algorithm (Lloyd, 1982)

Cluster dataset $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(n)}\right\}$, given K

1. Initialization: Choose K random vectors $\boldsymbol{Y} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \dots, \boldsymbol{y}^{(K)}\right\}$ as an initial mean set

k-means algorithm (Lloyd, 1982)

Cluster dataset $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(n)}\right\}$, given K

1. Initialization: Choose K random vectors $\boldsymbol{Y} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \dots, \boldsymbol{y}^{(K)}\right\}$ as an initial mean set
2. E-step: For each data point, find the closest class and label them. Dataset is divided into K classes

$$\boldsymbol{X}_i = \left\{\boldsymbol{x}^{(j)} \mid d(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(i)}) \le d(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(k)}),\ j = 1, \dots, n;\ k = 1, \dots, K\right\}$$

k-means algorithm (Lloyd, 1982)

Cluster dataset $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(n)}\right\}$, given K

1. Initialization: Choose K random vectors $\boldsymbol{Y} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \dots, \boldsymbol{y}^{(K)}\right\}$ as an initial mean set
2. E-step: For each data point, find the closest class and label them. Dataset is divided into K classes
$$\boldsymbol{X}_i = \left\{\boldsymbol{x}^{(j)} \mid d(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(i)}) \leq d(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(k)}),\ j = 1, \dots, n;\ k = 1, \dots, K\right\}$$
3. M-step: From the current clusters, their mean vectors are updated
$$\boldsymbol{y}^{(i)} = \frac{1}{N_i} \sum_{\boldsymbol{x} \in \omega_i} \boldsymbol{x} = \mathcal{C}(\boldsymbol{X}_i)$$

k-means algorithm (Lloyd, 1982)

Cluster dataset $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}\right\}$, given K

1. Initialization: Choose K random vectors $\boldsymbol{Y} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(K)}\right\}$ as an initial mean set
2. E-step: For each data point, find the closest class and label them. Dataset is divided into K classes

$$\boldsymbol{X}_i = \left\{\boldsymbol{x}^{(j)} \mid d(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(i)}) \leq d(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(k)}),\ j = 1, \ldots, n;\ k = 1, \ldots, K\right\}$$

3. M-step: From the current clusters, their mean vectors are updated

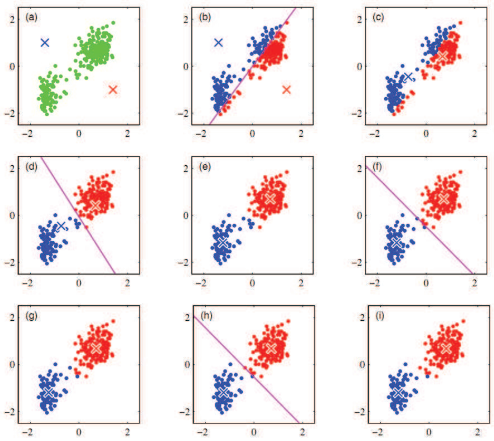
$$\boldsymbol{y}^{(i)} = \frac{1}{N_i} \sum_{\boldsymbol{x} \in \omega_i} \boldsymbol{x} = \mathcal{C}(\boldsymbol{X}_i)$$

4. Calculate the total distortion, the sum of the distance between each datapoint and its closest cluster mean.

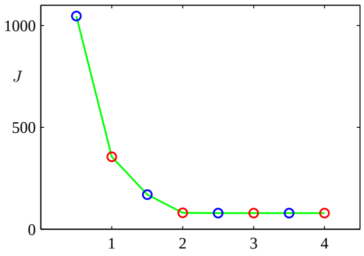
$$J = \sum_{i=1}^{K} \sum_{\boldsymbol{x} \in \omega_i} (\boldsymbol{x} - \boldsymbol{y}^{(i)})^2$$

5. Evaluate the convergence. If converged, stop. Else, go to step 2.

Illustration of k-means algorithm



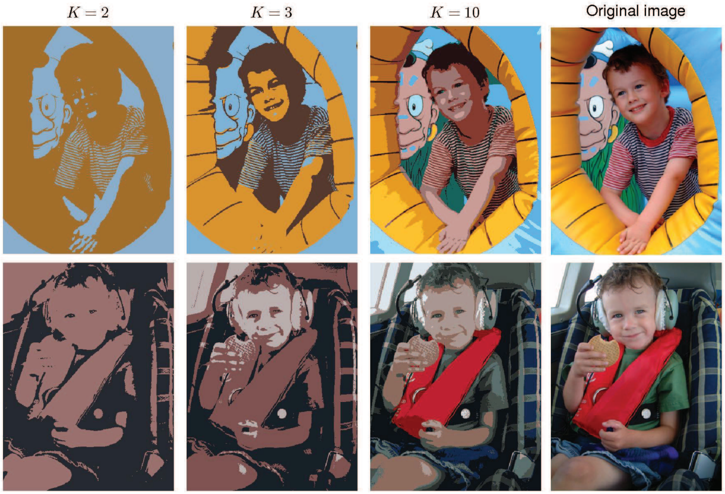
K=2



Blue : after E-step
Red : after M-step

Source: C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

k-means clustering applications



Source: C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

k-means clustering: Remarks

- The way to initialize the means is not specified. One popular way to start is to randomly choose k of the samples.
- A local optimization algorithm
    - Converge to a local rather than global minimum of J
    - Convergence properties of the k-means algorithm [MacQueen 1967]
    - The results depend on the initial values for the means. The standard solution is to try a number of different starting points.
- Uniform search
- Slow convergence
- It can happen that the set of samples closest to $\boldsymbol{y}^{(i)}$ is empty, so that $\boldsymbol{y}^{(i)}$ cannot be updated.
- The results depend on the value of K.

Basic Iterative Minimum-Squared-Error Clustering

- A sequential version of the k-means clustering algorithm.
  - k-means procedure waits until all n samples have been reclassified before updating.
  - the Basic Iterative Minimum-Squared-Error Clustering updates after each sample is reclassified.
- Disadvantages
  - more susceptible to being trapped in local minima
  - results depend on the order in which the candidates are selected
- Merits
  - at least a stepwise optimal procedure
  - suitable to problems in which samples are acquired sequentially and clustering must be done online

Fuzzy k-means algorithm

- Allows one piece of data to belong to two or more clusters
- Minimizing the below objective function

$$J = \sum_{k=1}^{K} \sum_{i=1}^{n} P(\omega_k|\boldsymbol{x}^{(i)})^m (\boldsymbol{x}^{(i)} - \boldsymbol{y}^{(k)})^2$$

  - m is a parameter to adjust the blending of different clusters. If $m = 0$, k-means. For $m \geq 1$, the criterion allows belonging to multiple clusters
  - $P(\omega_k|\boldsymbol{x}^{(i)})$ is the degree of membership of $\boldsymbol{x}^{(i)}$ in the cluster k
- an iterative optimization of the objective function shown above, with the update of membership $P(\omega_k|\boldsymbol{x}^{(i)})$ and the cluster centers $\boldsymbol{y}^{(k)}$ by:

$$P(\omega_k|\boldsymbol{x}^{(i)}) = \frac{1}{\sum_{j=1}^{K} \left(\frac{||\boldsymbol{x}^{(i)} - \boldsymbol{y}^{(k)}||}{||\boldsymbol{x}^{(i)} - \boldsymbol{y}^{(j)}||} \right)^{\frac{2}{m-1}}}$$

$$\boldsymbol{y}^{(k)} = \frac{\sum_{i=1}^{n} P(\omega_k|\boldsymbol{x}^{(i)})^m \boldsymbol{x}^{(i)}}{\sum_{i=1}^{n} P(\omega_k|\boldsymbol{x}^{(i)})^m}$$

Fuzzy k-means algorithm : Pseudocode

Cluster dataset, given K

1. Initialization: Initialize K random mean vectors $\boldsymbol{Y} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(K)}\right\}$. Initialize $P(\omega_k|\boldsymbol{x}^i)$ for $i = 1..n$, $k = 1..K$
2. Normalize $P(\omega_k|\boldsymbol{x}^{(i)})$ so that $\sum_k^K P(\omega_k|\boldsymbol{x}^{(i)}) = 1$ for $i = 1..n$

Fuzzy k-means algorithm : Pseudocode

Cluster dataset, given K

1. Initialization: Initialize K random mean vectors $\boldsymbol{Y} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(K)}\right\}$. Initialize $P(\omega_k|\boldsymbol{x}^i)$ for $i = 1..n$, $k = 1..K$
2. Normalize $P(\omega_k|\boldsymbol{x}^{(i)})$ so that $\sum_k^K P(\omega_k|\boldsymbol{x}^{(i)}) = 1$ for $i = 1..n$
3. Update mean vectors
$$\boldsymbol{y}^{(k)} = \frac{\sum_{i=1}^{n} P(\omega_k|\boldsymbol{x}^{(i)})^m \boldsymbol{x}^{(i)}}{\sum_{i=1}^{n} P(\omega_k|\boldsymbol{x}^{(i)})^m}$$
4. Update the degree of membership of $\boldsymbol{x}^{(i)}$ in the cluster k
$$P(\omega_k|\boldsymbol{x}^{(i)}) = \frac{1}{\sum_{j=1}^{K} \left(\frac{||\boldsymbol{x}^{(i)} - \boldsymbol{y}^{(k)}||}{||\boldsymbol{x}^{(i)} - \boldsymbol{y}^{(j)}||} \right)^{\frac{2}{m-1}}}$$

Fuzzy k-means algorithm : Pseudocode

Cluster dataset, given K

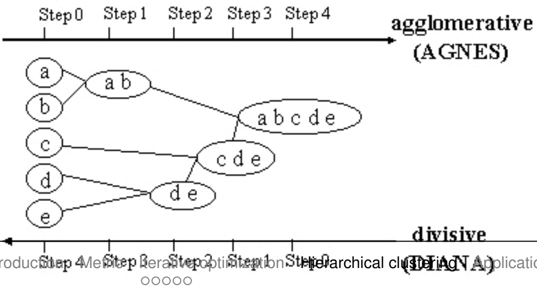
1. Initialization: Initialize K random mean vectors $\boldsymbol{Y} = \left\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(K)}\right\}$. Initialize $P(\omega_k|\boldsymbol{x}^i)$ for $i = 1..n$, $k = 1..K$
2. Normalize $P(\omega_k|\boldsymbol{x}^{(i)})$ so that $\sum_k^K P(\omega_k|\boldsymbol{x}^{(i)}) = 1$ for $i = 1..n$
3. Update mean vectors
$$\boldsymbol{y}^{(k)} = \frac{\sum_{i=1}^n P(\omega_k|\boldsymbol{x}^{(i)})^m \boldsymbol{x}^{(i)}}{\sum_{i=1}^n P(\omega_k|\boldsymbol{x}^{(i)})^m}$$
4. Update the degree of membership of $\boldsymbol{x}^{(i)}$ in the cluster k
$$P(\omega_k|\boldsymbol{x}^{(i)}) = \frac{1}{\sum_{j=1}^K \left(\frac{\|\boldsymbol{x}^{(i)} - \boldsymbol{y}^{(k)}\|}{\|\boldsymbol{x}^{(i)} - \boldsymbol{y}^{(j)}\|}\right)^{\frac{2}{m-1}}}$$
5. If $\max_{i,k} P(\omega_k|\boldsymbol{x}^{(i)})$ is converged, then stop. Else, go to step 2.

Another variation of k-means

- ISODATA, which stands for Iterative Self-Organizing Data Analysis Technique (Algorithm) is an extension to the k-means algorithm with some heuristics to automatically select the number of clusters
- The algorithm works in an iterative fashion
  (1) Perform k-means clustering
  (2) Split any clusters whose samples are sufficiently dissimilar
  (3) Merge any two clusters sufficiently close
  (4) Go to (1)

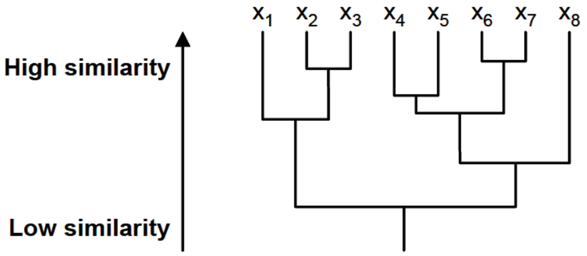
Hierarchical clustering

- k-means and ISODATA create disjoint clusters, resulting in a ”flat” data representation
  - However, sometimes it is desirable to obtain a hierarchical representation of data, with clusters and sub-clusters arranged in a tree-structured fashion (i.e., biological taxonomy)
- Hierarchical clustering methods can be grouped in two general classes
  - Agglomerative (bottom-up, merging) : Starting with n singleton clusters, successively merge clusters until one cluster is left
  - Divisive (top-down, splitting) : Starting with a unique cluster, successively split the clusters until n singleton examples are left



Dendrograms

- The preferred representation for hierarchical clusters is the dendrogram
- The dendrogram is a binary tree that shows the structure of the clusters
    - In addition to the binary tree, the dendrogram provides the similarity measure between clusters (the vertical axis)
- An alternative representation is based on sets
    - $\{\{x_1, \{x_2, x_3\}\}, \{\{\{x_4, x_5\}, \{x_6, x_7\}\}, x_8\}\}$
    - However, unlike the dendrogram, sets cannot express quantitative information



Divisive clustering

1. Start with one large cluster
2. Find ”worst” cluster
3. Split it
4. If $K < n$, go to step 2

- How to choose the ”worst” cluster
- How to split clusters
- The computations required by divisive clustering are more intensive than for agglomerative clustering methods.

Non-uniform Binary Split Algorithm

1. Initialization: Calculate a center for all data points. $k := 1$

$$\boldsymbol{y}^{(1)} = \mathcal{C}(\boldsymbol{X})$$

Non-uniform Binary Split Algorithm

1. Initialization: Calculate a center for all data points. $k := 1$

$$\boldsymbol{y}^{(1)} = \mathcal{C}(\boldsymbol{X})$$

2. Choose a class which has the largest distortion among current classes

$$J_i \geq J_j, \quad \forall j = 1, \ldots, k$$

Non-uniform Binary Split Algorithm

1. Initialization: Calculate a center for all data points. $k := 1$
$$\boldsymbol{y}^{(1)} = \mathcal{C}(\boldsymbol{X})$$

2. Choose a class which has the largest distortion among current classes
$$J_i \geq J_j, \quad \forall j = 1, \ldots, k$$

3. Split the class into two subclasses by using a small random vector
$$\boldsymbol{X}_a = \left\{ \boldsymbol{x} \text{ which is closer to } \boldsymbol{y}^{(i)} + \boldsymbol{v}_i \right\}$$
$$\boldsymbol{X}_b = \left\{ \boldsymbol{x} \text{ which is closer to } \boldsymbol{y}^{(i)} - \boldsymbol{v}_i \right\}$$

Non-uniform Binary Split Algorithm

1. Initialization: Calculate a center for all data points. $k := 1$

$$\boldsymbol{y}^{(1)} = \mathcal{C}(\boldsymbol{X})$$

2. Choose a class which has the largest distortion among current classes

$$J_i \geq J_j, \quad \forall j = 1, \ldots, k$$

3. Split the class into two subclasses by using a small random vector

$$\boldsymbol{X}_a = \left\{ \boldsymbol{x} \text{ which is closer to } \boldsymbol{y}^{(i)} + \boldsymbol{v}_i \right\}$$
$$\boldsymbol{X}_b = \left\{ \boldsymbol{x} \text{ which is closer to } \boldsymbol{y}^{(i)} - \boldsymbol{v}_i \right\}$$

4. Update the code vectors

$$\boldsymbol{y}^{(i)} := \mathcal{C}(\boldsymbol{X}_a) \text{ and } \boldsymbol{y}^{(k+1)} := \mathcal{C}(\boldsymbol{X}_b)$$

Non-uniform Binary Split Algorithm

1. Initialization: Calculate a center for all data points. $k := 1$
$$\boldsymbol{y}^{(1)} = \mathcal{C}(\boldsymbol{X})$$

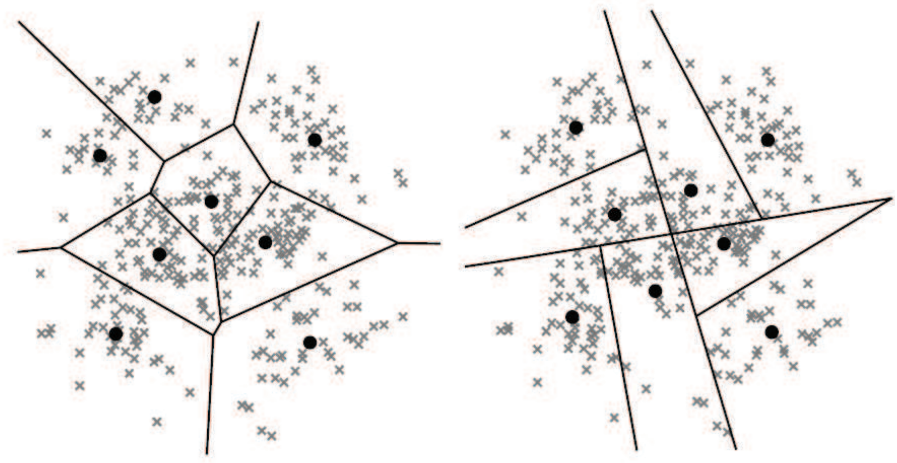
2. Choose a class which has the largest distortion among current classes
$$J_i \geq J_j, \quad \forall j = 1, \ldots, k$$

3. Split the class into two subclasses by using a small random vector
$$\boldsymbol{X}_a = \left\{\boldsymbol{x} \text{ which is closer to } \boldsymbol{y}^{(i)} + \boldsymbol{v}_i\right\}$$
$$\boldsymbol{X}_b = \left\{\boldsymbol{x} \text{ which is closer to } \boldsymbol{y}^{(i)} - \boldsymbol{v}_i\right\}$$

4. Update the code vectors
$$\boldsymbol{y}^{(i)} := \mathcal{C}(\boldsymbol{X}_a) \text{ and } \boldsymbol{y}^{(k+1)} := \mathcal{C}(\boldsymbol{X}_b)$$

5. If $k = K$, stop. Else, $k := k + 1$ and go to step 2.

k-means vs. Non-uniform binary split



(a) k-means (b) Binary split

Non-uniform binary split: Not good quality of codebook. But very fast

LBG algorithm

- Combination of k-means and Binary split
- Proposed by Linde, Buzo, and Gray (1980)
- Instead of random initial codebook, let's use codebook from binary split
- Faster convergence and better quality codebook than k-means algorithm

Pseudo-code of LBG algorithm

1. Initialization: Calculate a center for all data points. $k := 1$

$$\boldsymbol{y}^{(1)} = \mathcal{C}(\boldsymbol{X})$$

Pseudo-code of LBG algorithm

1. Initialization: Calculate a center for all data points. $k := 1$

$$\boldsymbol{y}^{(1)} = \mathcal{C}(\boldsymbol{X})$$

2. Splitting: For each cluster, split it into two subclasses. $k := 2k$

$$\boldsymbol{y}_a^{(i)} = \boldsymbol{y}^{(i)} + \boldsymbol{v}_i \text{ and } \boldsymbol{y}_b^{(i)} = \boldsymbol{y}^{(i)} - \boldsymbol{v}_i$$

Pseudo-code of LBG algorithm

1. Initialization: Calculate a center for all data points. $k := 1$

$$\boldsymbol{y}^{(1)} = \mathcal{C}(\boldsymbol{X})$$

2. Splitting: For each cluster, split it into two subclasses. $k := 2k$

$$\boldsymbol{y}_a^{(i)} = \boldsymbol{y}^{(i)} + \boldsymbol{v}_i \text{ and } \boldsymbol{y}_b^{(i)} = \boldsymbol{y}^{(i)} - \boldsymbol{v}_i$$

3. Iteration (k-means algorithm)
   - a E-step: For each data point, find the closest cluster which achieves the minimum distance measure
   - b M-step: From the current cluster, update their mean vectors $\boldsymbol{y}^{(i)} = \mathcal{C}(\boldsymbol{X}_i)$
   - c Iterate E-step and M-step until it converges

Pseudo-code of LBG algorithm

1. Initialization: Calculate a center for all data points. $k := 1$

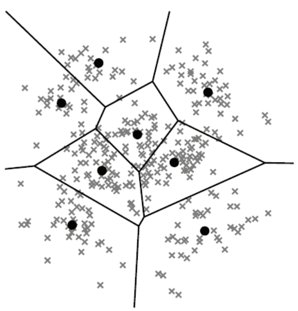
$$\boldsymbol{y}^{(1)} = \mathcal{C}(\boldsymbol{X})$$

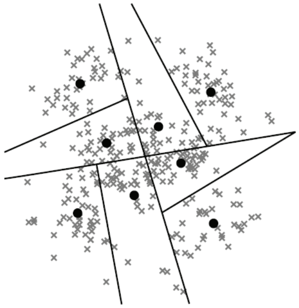
2. Splitting: For each cluster, split it into two subclasses. $k := 2k$

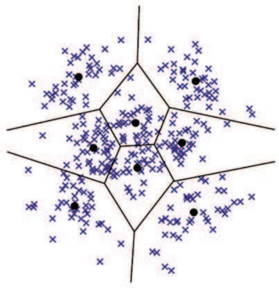
$$\boldsymbol{y}_a^{(i)} = \boldsymbol{y}^{(i)} + \boldsymbol{v}_i \text{ and } \boldsymbol{y}_b^{(i)} = \boldsymbol{y}^{(i)} - \boldsymbol{v}_i$$

3. Iteration (k-means algorithm)
   - a E-step: For each data point, find the closest cluster which achieves the minimum distance measure
   - b M-step: From the current cluster, update their mean vectors $\boldsymbol{y}^{(i)} = \mathcal{C}(\boldsymbol{X}_i)$
   - c Iterate E-step and M-step until it converges
4. If the desired number of code vectors is obtained ($k = K$), stop. Else, go to step 2.

Comparison


kmeans


Binary Split


LBG

Agglomerative clustering

1. Start with n singleton cluster
2. Find nearest clusters
3. Merge them
4. If $K > 1$, go to step 2

- How to find the "nearest" pair of clusters
  - Minimum distance $d_{min}(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{x} \in \omega_i, \boldsymbol{x}' \in \omega_j} \| \boldsymbol{x} - \boldsymbol{x}' \|$
  - Maximum distance $d_{max}(\boldsymbol{X}_i, \boldsymbol{X}_j) = \max_{\boldsymbol{x} \in \omega_i, \boldsymbol{x}' \in \omega_j} \| \boldsymbol{x} - \boldsymbol{x}' \|$
  - Average distance $d_{avg}(\boldsymbol{X}_i, \boldsymbol{X}_j) = \frac{1}{N_i N_j} \sum_{\boldsymbol{x} \in \omega_i} \sum_{\boldsymbol{x}' \in \omega_j} \| \boldsymbol{x} - \boldsymbol{x}' \|$
  - Mean distance $d_{mean}(\boldsymbol{X}_i, \boldsymbol{X}_j) = \| \boldsymbol{y}^{(i)} - \boldsymbol{y}^{(j)} \|$

Agglomerative clustering

Minimum distance

- When d_{min} is used to measure distance between clusters, the algorithm is called the *nearest neighbor* or *single-linkage* clustering algorithm
- If the algorithm is allowed to run until only one cluster remains, the result is a minimum spanning tree (MST)

Maximum distance

- When d_{max} is used to measure distance between clusters, the algorithm is called the *farthest neighbor* or *complete-linkage* clustering algorithm
- From a graph-theoretic point of view, each cluster constitutes a complete sub-graph

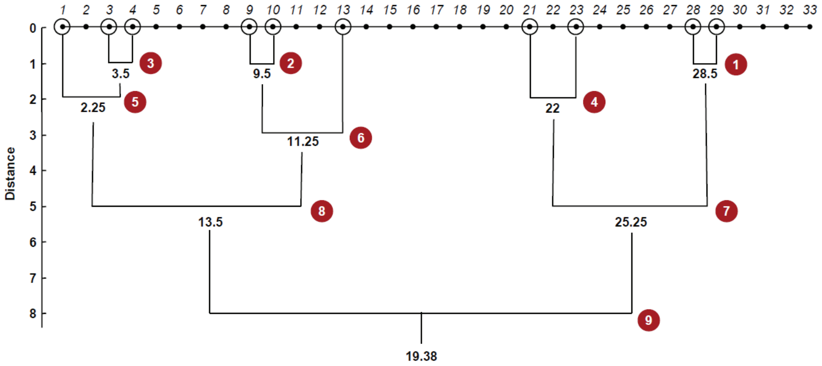
Average and mean distance

- The average and mean distance approaches are more robust to outliers
- Of the two, the mean distance is computationally more attractive, since Notice that the average distance approach involves the computation of N_iN_j distances for each pair of clusters

Agglomerative clustering example

Perform agglomerative clustering on the following dataset using the single-linkage metric

- $X = \{1, 3, 4, 9, 10, 13, 21, 23, 28, 29\}$
- In case of ties, always merge the pair of clusters with the largest mean
- Indicate the order in which the merging operations occur

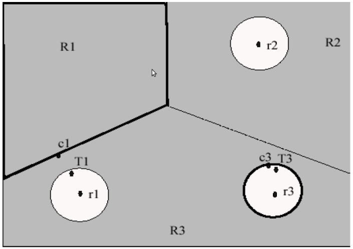


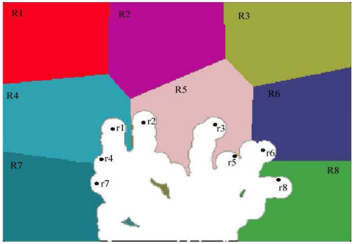
Multi-Robot Exploration

- Task definition
  - Exploration of unknown areas by means of k mobile robots.
  - Application: Search/Rescue robot, planetary exploration, reconnaissance
- Problems
  - Reduce the difference of waiting time among different regions of a workspace.
  - Ensure a balanced exploration of the environment

Wu L., Puig D., and Garcia M. A. *Balanced Multi-Robot Exploration through a Global Optimization Strategy*. Journal of Physical Agents. 2010.

Multi-Robot Exploration





- K-means is used to partition an unexplored space in regions.
- Each region is assigned to a robot solving an optimization problem.

Wu L., Puig D., and Garcia M. A. *Balanced Multi-Robot Exploration through a Global Optimization Strategy*. Journal of Physical Agents. 2010.

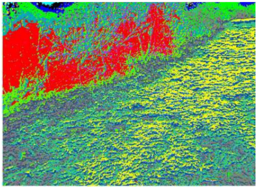
Outdoor Robots Navigation

- Task definition
  - Autonomous navigation for outdoor, unstructured environments.
  - Recognize navigable terrain and avoid obstacles, based on their appearance.
- Problem
  - Do reliable segmentation of outdoor scenes in an efficient manner (online).

Blas M. R., Agrawal M., Sundaresan A., and Konolige K. *Fast Color/Texture Segmentation For Outdoor Robots*. IEEE International Conference on Intelligent Robots and Systems. 2008.

Outdoor Robots Navigation

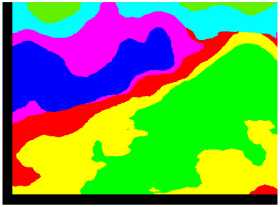
a)

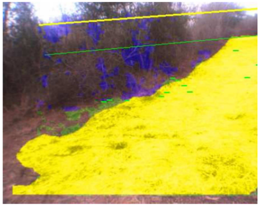

b)

- Use compact texture/color descriptors (feature vectors) and fast unsupervised clustering algorithms.
- K-means is used to cluster neighborhood feature vectors in a small set of basis vectors (*textons*).
- Each pixel is classified as belonging to one cluster using Euclidean distance (b).

Blas M. R., Agrawal M., Sundaresan A., and Konolige K. *Fast Color/Texture Segmentation For Outdoor Robots*. IEEE International Conference on Intelligent Robots and Systems. 2008.

Outdoor Robots Navigation


c)


d)

- Extract an histogram counting textons in a neighborhood of each pixel
- K-means to extract a set of histogram profiles ($k = 8$).
- *Earth Movers Distance* is used to merge similar clusters (c).

Blas M. R., Agrawal M., Sundaresan A., and Konolige K. *Fast Color/Texture Segmentation For Outdoor Robots*. IEEE International Conference on Intelligent Robots and Systems. 2008.

Summary and Next Lecture

- What we have learned
  - k-means algorithm
  - hierarchical clustering
- Reading: Duda Chap. 10.4, 10.6-9, Bishop Chap. 9.1, Michell Chap. 6.12
- Next Lecture
  - Maximum Likelihood Estimation
  - Gaussian Mixture Model