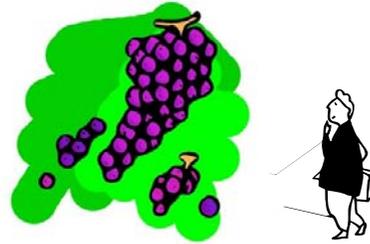


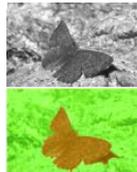
Clustering

Class 14. 18 Oct 2011

Clustering

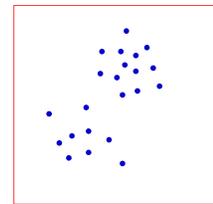


How



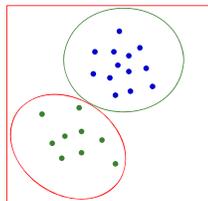
Clustering

- What is clustering
 - Clustering is the determination of naturally occurring grouping of data/instances (with low within-group variability and high between-group variability)



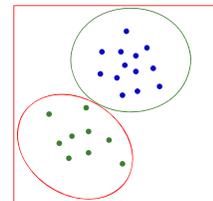
Clustering

- What is clustering
 - Clustering is the determination of naturally occurring grouping of data/instances (with low within-group variability and high between-group variability)



Clustering

- What is clustering
 - Clustering is the determination of naturally occurring grouping of data/instances (with low within-group variability and high between-group variability)
- How is it done
 - Find groupings of data such that the groups optimize a "within-group-variability" objective function of some kind



Clustering

- What is clustering
 - Clustering is the determination of naturally occurring grouping of data/instances (with low within-group variability and high between-group variability)
- How is it done
 - Find groupings of data such that the groups optimize a "within-group-variability" objective function of some kind
 - The objective function used affects the nature of the discovered clusters
 - E.g. Euclidean distance and distance from center result in different clusters in this example

18 Oct 2011 7

Clustering

- What is clustering
 - Clustering is the determination of naturally occurring grouping of data/instances (with low within-group variability and high between-group variability)
- How is it done
 - Find groupings of data such that the groups optimize a "within-group-variability" objective function of some kind
 - The objective function used affects the nature of the discovered clusters
 - E.g. Euclidean distance and distance from center result in different clusters in this example

18 Oct 2011 8

Why Clustering

- Automatic grouping into "Classes"
 - Different clusters may show different behavior
- Quantization
 - All data within a cluster are represented by a single point
- Preprocessing step for other algorithms
 - Indexing, categorization, etc.

18 Oct 2011 9

Clustering criteria

- Compactness criterion
 - Measure that shows how "good" clusters are
 - The objective function
- Distance of a point from a cluster
 - To determine the cluster a data vector belongs to

18 Oct 2011 10

"Compactness" criteria for clustering

- Distance based measures
 - Total distance between each element in the cluster and every other element in the cluster

18 Oct 2011 11

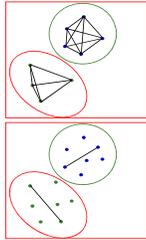
"Compactness" criteria for clustering

- Distance based measures
 - Total distance between each element in the cluster and every other element in the cluster

18 Oct 2011 12

“Compactness” criteria for clustering

- Distance based measures
 - Total distance between each element in the cluster and every other element in the cluster
 - Distance between the two farthest points in the cluster

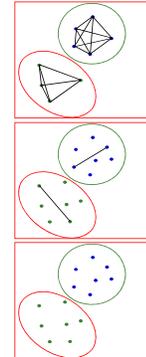


18 Oct 2011

13

“Compactness” criteria for clustering

- Distance based measures
 - Total distance between each element in the cluster and every other element in the cluster
 - Distance between the two farthest points in the cluster
 - Total distance of every element in the cluster from the centroid of the cluster

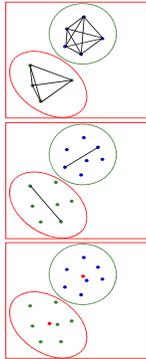


18 Oct 2011

14

“Compactness” criteria for clustering

- Distance based measures
 - Total distance between each element in the cluster and every other element in the cluster
 - Distance between the two farthest points in the cluster
 - Total distance of every element in the cluster from the centroid of the cluster

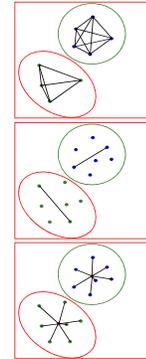


18 Oct 2011

15

“Compactness” criteria for clustering

- Distance based measures
 - Total distance between each element in the cluster and every other element in the cluster
 - Distance between the two farthest points in the cluster
 - Total distance of every element in the cluster from the centroid of the cluster



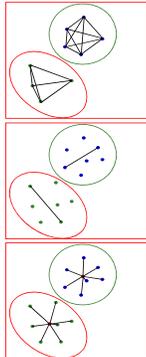
18 Oct 2011

16

“Compactness” criteria for clustering

- Distance based measures
 - Total distance between each element in the cluster and every other element in the cluster
 - Distance between the two farthest points in the cluster
 - Total distance of every element in the cluster from the centroid of the cluster
 - Distance measures are often weighted Minkowski metrics

$$dist = \sqrt[w]{w_1|a_1 - b_1|^w + w_2|a_2 - b_2|^w + \dots + w_M|a_M - b_M|^w}$$



18 Oct 2011

17

Clustering: Distance from cluster

- How far is a data point from a cluster?
 - Euclidean or Minkowski distance from the centroid of the cluster

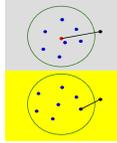


18 Oct 2011

18

Clustering: Distance from cluster

- How far is a data point from a cluster?
 - Euclidean or Minkowski distance from the centroid of the cluster
 - Distance from the closest point in the cluster

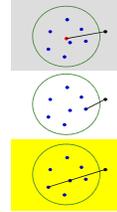


18 Oct 2011

19

Clustering: Distance from cluster

- How far is a data point from a cluster?
 - Euclidean or Minkowski distance from the centroid of the cluster
 - Distance from the closest point in the cluster
 - Distance from the farthest point in the cluster

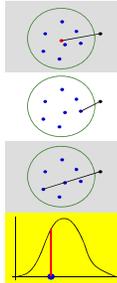


18 Oct 2011

20

Clustering: Distance from cluster

- How far is a data point from a cluster?
 - Euclidean or Minkowski distance from the centroid of the cluster
 - Distance from the closest point in the cluster
 - Distance from the farthest point in the cluster
 - Probability of data measured on cluster distribution

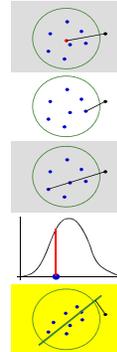


18 Oct 2011

21

Clustering: Distance from cluster

- How far is a data point from a cluster?
 - Euclidean or Minkowski distance from the centroid of the cluster
 - Distance from the closest point in the cluster
 - Distance from the farthest point in the cluster
 - Probability of data measured on cluster distribution
 - Fit of data to cluster-based regression



18 Oct 2011

22

Optimal clustering: Exhaustive enumeration

- All possible combinations of data must be evaluated
 - If there are M data points, and we desire N clusters, the number of ways of separating M instances into N clusters is

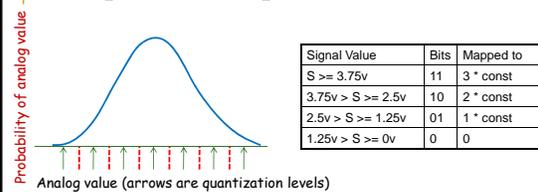
$$\frac{1}{M!} \sum_{i=0}^N (-1)^i \binom{N}{i} (N-i)^M$$

- Exhaustive enumeration based clustering requires that the objective function (the "Goodness measure") be evaluated for every one of these, and the best one chosen
- This is the only correct way of optimal clustering
 - Unfortunately, it is also computationally unrealistic

18 Oct 2011

23

Not-quite non sequitur: Quantization



- Linear quantization (uniform quantization):
 - Each digital value represents an equally wide range of analog values
 - Regardless of distribution of data
 - Digital-to-analog conversion represented by a "uniform" table

18 Oct 2011

24

Not-quite non sequitur: Quantization

Signal Value	Bits	Mapped to
$S \geq 4v$	11	4.5
$4v > S \geq 2.5v$	10	3.25
$2.5v > S \geq 1v$	01	1.25
$1.0v > S \geq 0v$	0	0.5

- Non-Linear quantization:
 - Each digital value represents a different range of analog values
 - Finer resolution in high-density areas
 - Mu-law / A-law assumes a gaussian-like distribution of data
 - Digital-to-analog conversion represented by a "non-uniform" table

18 Oct 2011 25

Non-uniform quantization

- If data distribution is not Gaussianish?
 - Mu-law / A-law are not optimal
 - How to compute the optimal ranges for quantization
 - Or the optimal table

18 Oct 2011 26

The Lloyd Quantizer

- Lloyd quantizer: An iterative algorithm for computing optimal quantization tables for non-uniformly distributed data
- Learned from "training" data**

18 Oct 2011 27

Lloyd Quantizer

- Randomly initialize quantization points
- Right column entries of quantization table

18 Oct 2011 28

Lloyd Quantizer

- Randomly initialize quantization points
 - Right column entries of quantization table
- Assign all training points to the nearest quantization point
 - Draw boundaries

18 Oct 2011 29

Lloyd Quantizer

- Randomly initialize quantization points
 - Right column entries of quantization table
- Assign all training points to the nearest quantization point
 - Draw boundaries
- Reestimate quantization points

18 Oct 2011 30

Lloyd Quantizer

- Randomly initialize quantization points
 - Right column entries of quantization table
- Assign all training points to the nearest quantization point
 - Draw boundaries
- Reestimate quantization points
- Iterate until convergence

18 Oct 2011 31

Generalized Lloyd Algorithm: K-means clustering

- K means is an iterative algorithm for clustering **vector** data
 - McQueen, J. 1967. "Some methods for classification and analysis of multivariate observations." Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 281-297
- General procedure:
 - Initially group data into the required number of clusters somehow (initialization)
 - Assign each data point to the closest cluster
 - Once all data points are assigned to clusters, redefine clusters
 - Iterate

18 Oct 2011 32

K-means

- Problem: Given a set of data vectors, find natural clusters
- Clustering criterion is **scatter**: distance from the centroid
 - Every cluster has a centroid
 - The centroid represents the cluster
- Definition: The **centroid** is the weighted mean of the cluster
 - Weight = 1 for basic scheme

$$m_{cluster} = \frac{1}{\sum w_i} \sum_{i \in cluster} w_i x_i$$

18 Oct 2011 33

K-means

- Initialize a set of centroids randomly

18 Oct 2011 34

K-means

- Initialize a set of centroids randomly
- For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$

18 Oct 2011 35

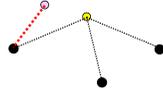
K-means

- Initialize a set of centroids randomly
- For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
- Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum

18 Oct 2011 36

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum



18 Oct 2011

37

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum

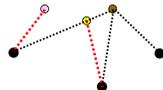


18 Oct 2011

38

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum

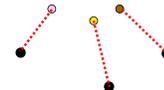


18 Oct 2011

39

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum



18 Oct 2011

40

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum

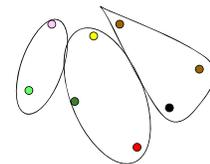


18 Oct 2011

41

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum



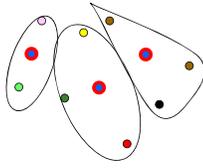
18 Oct 2011

42

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum
4. When all data points are clustered, recompute centroids

$$m_{cluster} = \frac{1}{\sum_{i \in cluster} w_i} \sum_{i \in cluster} w_i x_i$$

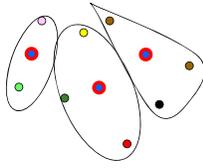


18 Oct 2011 43

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum
4. When all data points are clustered, recompute centroids

$$m_{cluster} = \frac{1}{\sum_{i \in cluster} w_i} \sum_{i \in cluster} w_i x_i$$
5. If not converged, go back to 2



18 Oct 2011 44

K-Means comments

- The distance metric determines the clusters
 - In the original formulation, the distance is L2 distance
 - Euclidean norm, $w_i = 1$

$$\text{distance}_{cluster}(x, m_{cluster}) = \|x - m_{cluster}\|_2 \quad m_{cluster} = \frac{1}{N_{cluster}} \sum_{i \in cluster} x_i$$

- If we replace every x by $m_{cluster}(x)$, we get *Vector Quantization*
- K-means is an instance of *generalized EM*
- Not guaranteed to converge for all distance metrics

18 Oct 2011 45

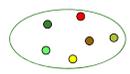
Initialization

- Random initialization
- Top-down clustering
 - Initially partition the data into two (or a small number of) clusters using K means
 - Partition each of the resulting clusters into two (or a small number of) clusters, also using K means
 - Terminate when the desired number of clusters is obtained

18 Oct 2011 46

K-Means for Top-Down clustering

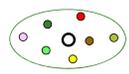
1. Start with one cluster



18 Oct 2011 47

K-Means for Top-Down clustering

1. Start with one cluster
2. Split each cluster into two:
 - Perturb centroid of cluster slightly (by < 5%) to generate two centroids



18 Oct 2011 48

K-Means for Top-Down clustering

1. Start with one cluster
2. Split each cluster into two:
 - Perturb centroid of cluster slightly (by < 5%) to generate two centroids

18 Oct 2011 49

K-Means for Top-Down clustering

1. Start with one cluster
2. Split each cluster into two:
 - Perturb centroid of cluster slightly (by < 5%) to generate two centroids
3. Initialize K means with new set of centroids

18 Oct 2011 50

K-Means for Top-Down clustering

1. Start with one cluster
2. Split each cluster into two:
 - Perturb centroid of cluster slightly (by < 5%) to generate two centroids
3. Initialize K means with new set of centroids
4. Iterate Kmeans until convergence

18 Oct 2011 51

K-Means for Top-Down clustering

1. Start with one cluster
2. Split each cluster into two:
 - Perturb centroid of cluster slightly (by < 5%) to generate two centroids
3. Initialize K means with new set of centroids
4. Iterate Kmeans until convergence

18 Oct 2011 52

K-Means for Top-Down clustering

1. Start with one cluster
2. Split each cluster into two:
 - Perturb centroid of cluster slightly (by < 5%) to generate two centroids
3. Initialize K means with new set of centroids
4. Iterate Kmeans until convergence
5. If the desired number of clusters is not obtained, return to 2

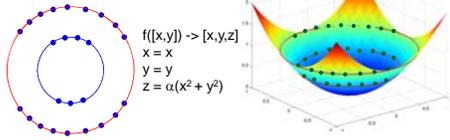
18 Oct 2011 53

Non-Euclidean clusters

- Basic K-means results in good clusters in Euclidean spaces
 - Alternately stated, will only find clusters that are "good" in terms of Euclidean distances
- Will not find other types of clusters

18 Oct 2011 54

Non-euclidean clusters

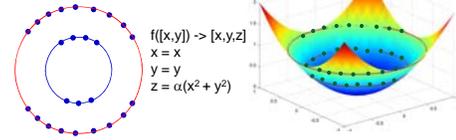


- For other forms of clusters we must modify the distance measure
 - E.g. distance from a circle
- May be viewed as a distance in a higher dimensional space
 - I.e. *Kernel* distances
 - Kernel K-means*
- Other related clustering mechanisms:
 - Spectral clustering
 - Non-linear weighting of adjacency
 - Normalized cuts..

18 Oct 2011

55

The Kernel Trick



- Transform the data into a synthetic higher-dimensional space where the desired patterns become natural clusters
 - E.g. the quadratic transform above
- Problem: What is the function/space?
- Problem: Distances in higher dimensional-space are more expensive to compute
 - Yet only carry the same information in the lower-dimensional space

18 Oct 2011

56

Distance in higher-dimensional space

- Transform data x through an *unknown* function $\Phi(x)$ into a higher (potentially infinite) dimensional space
 - $z = \Phi(x)$
- The distance between two points is computed in the higher-dimensional space
 - $d(x_1, x_2) = \|z_1 - z_2\|^2 = \|\Phi(x_1) - \Phi(x_2)\|^2$
- $d(x_1, x_2)$ can be computed without computing z
 - Since it is a direct function of x_1 and x_2

18 Oct 2011

57

Distance in higher-dimensional space

- Distance in lower-dimensional space: A combination of dot products
 - $\|x_1 - x_2\|^2 = (x_1 - x_2)^T(x_1 - x_2) = x_1 \cdot x_1 + x_2 \cdot x_2 - 2x_1 \cdot x_2$
- Distance in higher-dimensional space
 - $d(x_1, x_2) = \|\Phi(x_1) - \Phi(x_2)\|^2 = \Phi(x_1) \cdot \Phi(x_1) + \Phi(x_2) \cdot \Phi(x_2) - 2\Phi(x_1) \cdot \Phi(x_2)$
- $d(x_1, x_2)$ can be computed without knowing $\Phi(x)$ if:
 - $\Phi(x_1), \Phi(x_2)$ can be computed for any x_1 and x_2 without knowing $\Phi(\cdot)$

18 Oct 2011

58

The Kernel function

- A kernel function $K(x_1, x_2)$ is a function such that:
 - $K(x_1, x_2) = \Phi(x_1) \cdot \Phi(x_2)$
- Once such a kernel function is found, the distance in higher-dimensional space can be found in terms of the kernels
 - $d(x_1, x_2) = \|\Phi(x_1) - \Phi(x_2)\|^2 = \Phi(x_1) \cdot \Phi(x_1) + \Phi(x_2) \cdot \Phi(x_2) - 2\Phi(x_1) \cdot \Phi(x_2) = K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2)$
- But what is $K(x_1, x_2)$?

18 Oct 2011

59

A property of the dot product

- For any vector v , $v^T v = \|v\|^2 \geq 0$
 - This is just the length of v and is therefore non-negative
- For any vector $u = \sum_i a_i v_i$, $\|u\|^2 \geq 0$
 - $\Rightarrow (\sum_i a_i v_i)^T (\sum_i a_i v_i) \geq 0$
 - $\Rightarrow \sum_i \sum_j a_i a_j v_i \cdot v_j \geq 0$
- This holds for ANY real $\{a_1, a_2, \dots\}$

18 Oct 2011

60

The Mercer Condition

- If $\mathbf{z} = \Phi(\mathbf{x})$ is a high-dimensional vector derived from \mathbf{x} then for all real $\{a_1, a_2, \dots\}$ and any set $\{\mathbf{z}_1, \mathbf{z}_2, \dots\} = \{\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \dots\}$
 - $\sum_i \sum_j a_i a_j \mathbf{z}_i \cdot \mathbf{z}_j \geq 0$
 - $\sum_i \sum_j a_i a_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \geq 0$
- If $K(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1) \cdot \Phi(\mathbf{x}_2)$
 - $\sum_i \sum_j a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0$
- Any function $K()$ that satisfies the above condition is a valid kernel function

18 Oct 2011

61

The Mercer Condition

- $K(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1) \cdot \Phi(\mathbf{x}_2)$
 - $\sum_i \sum_j a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0$
- A corollary: If any kernel $K(\cdot)$ satisfies the Mercer condition

$$d(\mathbf{x}_1, \mathbf{x}_2) = K(\mathbf{x}_1, \mathbf{x}_1) + K(\mathbf{x}_2, \mathbf{x}_2) - 2K(\mathbf{x}_1, \mathbf{x}_2)$$
 satisfies the following requirements for a "distance"
 - $d(\mathbf{x}, \mathbf{x}) = 0$
 - $d(\mathbf{x}, \mathbf{y}) \geq 0$
 - $d(\mathbf{x}, \mathbf{w}) + d(\mathbf{w}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{y})$

18 Oct 2011

62

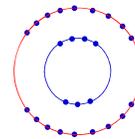
Typical Kernel Functions

- Linear: $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y} + c$
- Polynomial $K(\mathbf{x}, \mathbf{y}) = (\mathbf{a} \mathbf{x}^T \mathbf{y} + c)^n$
- Gaussian: $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / \sigma^2)$
- Exponential: $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\| / \lambda)$
- Several others
 - Choosing the right Kernel with the right parameters for your problem is an artform

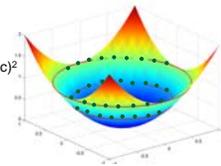
18 Oct 2011

63

Kernel K-means



$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + c)^2$$



- Perform the K-mean in the Kernel space
 - The space of $\mathbf{z} = \Phi(\mathbf{x})$
- The algorithm..

18 Oct 2011

64

K-means

- Initialize the clusters with a random set of K points
 - Cluster has 1 point
- For each data point \mathbf{x} , find the closest cluster

$$\text{cluster}(\mathbf{x}) = \min_{\text{cluster}} d(\mathbf{x}, \text{cluster}) = \min_{\text{cluster}} \|\Phi(\mathbf{x}) - \mathbf{m}_{\text{cluster}}\|^2$$

$$d(\mathbf{x}, \text{cluster}) = \|\Phi(\mathbf{x}) - \mathbf{m}_{\text{cluster}}\|^2 = \left(\Phi(\mathbf{x}) - C \sum_{i \in \text{cluster}} w_i \Phi(\mathbf{x}_i) \right)^T \left(\Phi(\mathbf{x}) - C \sum_{i \in \text{cluster}} w_i \Phi(\mathbf{x}_i) \right)$$

$$= \left(\Phi(\mathbf{x})^T \Phi(\mathbf{x}) - 2C \sum_{i \in \text{cluster}} w_i \Phi(\mathbf{x})^T \Phi(\mathbf{x}_i) + C^2 \sum_{i \in \text{cluster}} \sum_{j \in \text{cluster}} w_i w_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \right)$$

$$= K(\mathbf{x}, \mathbf{x}) - 2C \sum_{i \in \text{cluster}} w_i K(\mathbf{x}, \mathbf{x}_i) + C^2 \sum_{i \in \text{cluster}} \sum_{j \in \text{cluster}} w_i w_j K(\mathbf{x}_i, \mathbf{x}_j)$$

Computed entirely using only the kernel function!

18 Oct 2011

65

K-means

1. Initialize a set of centroids randomly



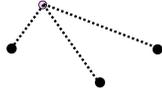
18 Oct 2011

66

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster

- $d_{cluster} = \text{distance}(x, m_{cluster})$



18 Oct 2011

67

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum

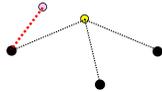


18 Oct 2011

68

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum



18 Oct 2011

69

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum

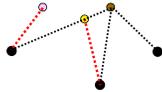


18 Oct 2011

70

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum



18 Oct 2011

71

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum



18 Oct 2011

72

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum

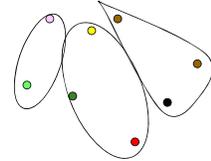


18 Oct 2011

73

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum

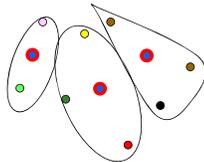


18 Oct 2011

74

K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum
4. When all data points are clustered, recompute centroids



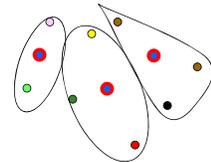
$$m_{cluster} = \frac{1}{\sum_{x \in cluster} w_x} \sum_{x \in cluster} w_x x_i$$

18 Oct 2011

75

Kernel K-means

1. Initialize a set of centroids randomly
2. For each data point x , find the distance from the centroid for each cluster
 - $d_{cluster} = \text{distance}(x, m_{cluster})$
3. Put data point in the cluster of the closest centroid
 - Cluster for which $d_{cluster}$ is minimum
4. When all data points are clustered, recompute centroids
5. If not converged, go back to 2



$$m_{cluster} = \frac{1}{\sum_{x \in cluster} w_x} \sum_{x \in cluster} w_x x_i$$

18 Oct 2011

76

How many clusters?

- Assumptions:
 - Dimensionality of kernel space > no. of clusters
 - Clusters represent separate *directions* in Kernel spaces
- Kernel correlation matrix \mathbf{K}
 - $\mathbf{K}_{ij} = K(x_i, x_j)$
- Find Eigen values Λ and Eigen vectors \mathbf{e} of kernel matrix
 - No. of clusters = no. of dominant λ_i ($1^T \mathbf{e}_i$) terms

18 Oct 2011

77

Spectral Methods

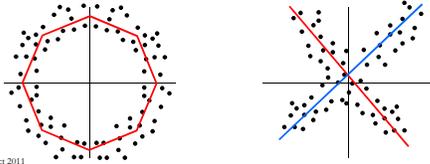
- “Spectral” methods attempt to find “principal” subspaces of the high-dimensional kernel space
- Clustering is performed in the principal subspaces
 - Normalized cuts
 - Spectral clustering
- Involves finding Eigenvectors and Eigen values of Kernel matrix
- Fortunately, provably analogous to Kernel K-means

18 Oct 2011

78

Other clustering methods

- Regression based clustering
- Find a regression representing each cluster
- Associate each point to the cluster with the best regression
 - Related to kernel methods



18 Oct 2011

79

Clustering..

- Many many other variants
- Many applications..
- Important: Appropriate choice of feature
 - Appropriate choice of feature may eliminate need for kernel trick..
 - Google is your friend.

18 Oct 2011

80