Machine Learning 2

Topics for Today

- Binary Classification
  - Logistic Regression
  - Support Vector Machine (SVM)
- Clustering
  - K-Means
  - Spectral Clustering

Topics for Today

- Binary Classification
  - **Logistic Regression**
  - Support Vector Machine (SVM)
- Clustering
  - K-Means
  - Spectral Clustering

Logistic Regression

An Introduction

- Logistic regression is a method for analyzing relative probabilities between discrete outcomes (binary or categorical dependent variables)
  - **Binary outcome**: standard logistic regression
    - ie. Dead (1) or NonDead (0)
  - **Categorical outcome**: multinomial logistic regression
    - ie. Zombie (1) or Vampire (2) or Mummy (3) or Rasputin (4)

Logistic Regression Model

- $\Pr(y = 1|x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$
- $\log \frac{\Pr(y = 1|x)}{1-\Pr(y = 1|x)} = \alpha + \beta x$

- This is a nonlinear model
  - A given change in x will often have less impact when Pr(y=1|x) is close to the extremes (0 or 1) compared to middle values.
- Buying new or used car (from Agresti 2002)
  - Increasing family income by $50,000 would have less effect if x=$1,000,000 (for which Pr(y=1|x) is near 1) compared to x=$50,000

Interpreting Coefficients

- A positive coefficient, β, indicates that higher levels of x are associated with an increase in Pr(y=1|x).
- A negative coefficient indicates that higher levels of x are associated with a decrease in Pr(y=1|x).
- When β=0, y and x are independent of one another.

How It All Works

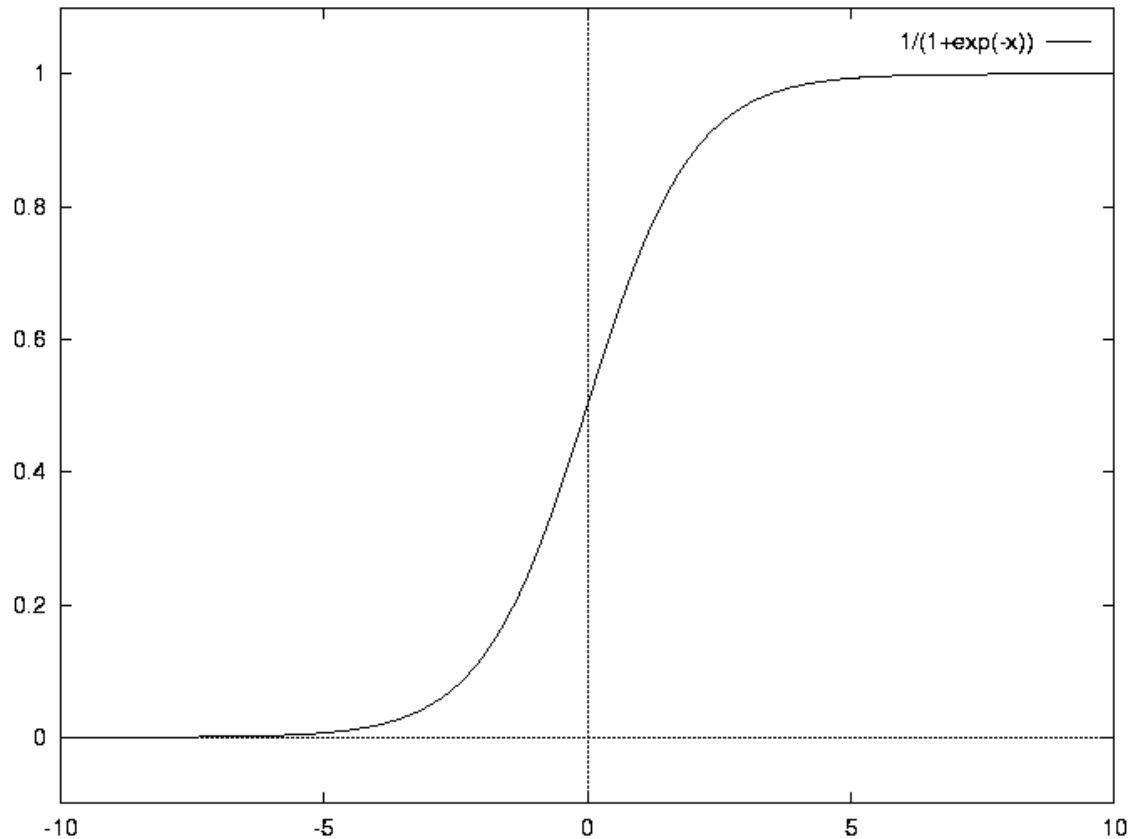
- The logistic equation is written as a function of *z*, where *z* is a measure of the total contribution of each variable *x* used to predict the outcome

$$f(z) = \frac{1}{1 + e^{-z}}$$

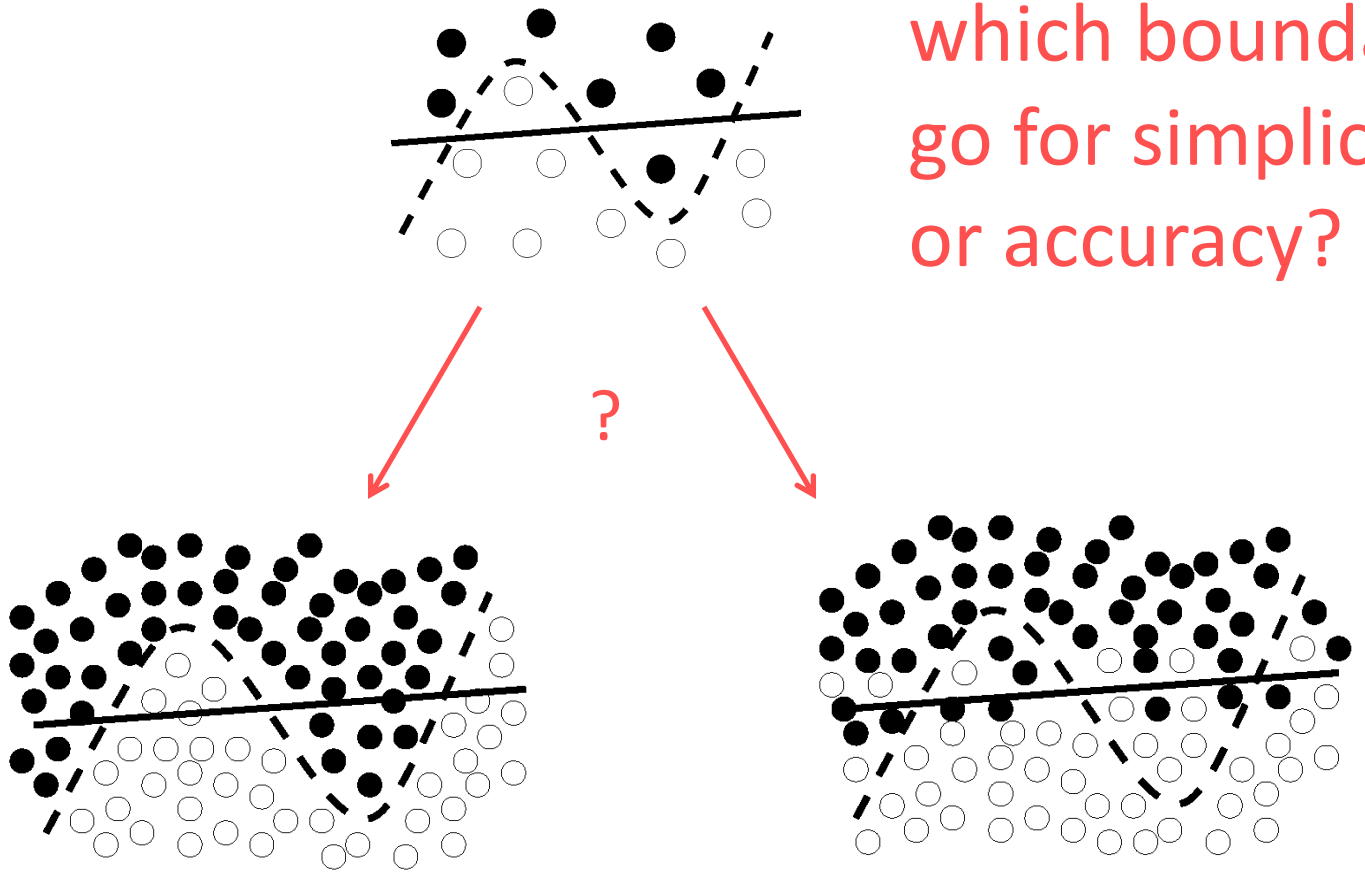
$$z = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

- Coefficients determined by maximum likelihood estimation (MLE), so larger sample sizes are needed than for OLS

Graph of the Logistic Function



Balancing generalization and overfitting



more training data makes the choice more obvious

logistic-regression.ipynb

Topics for Today

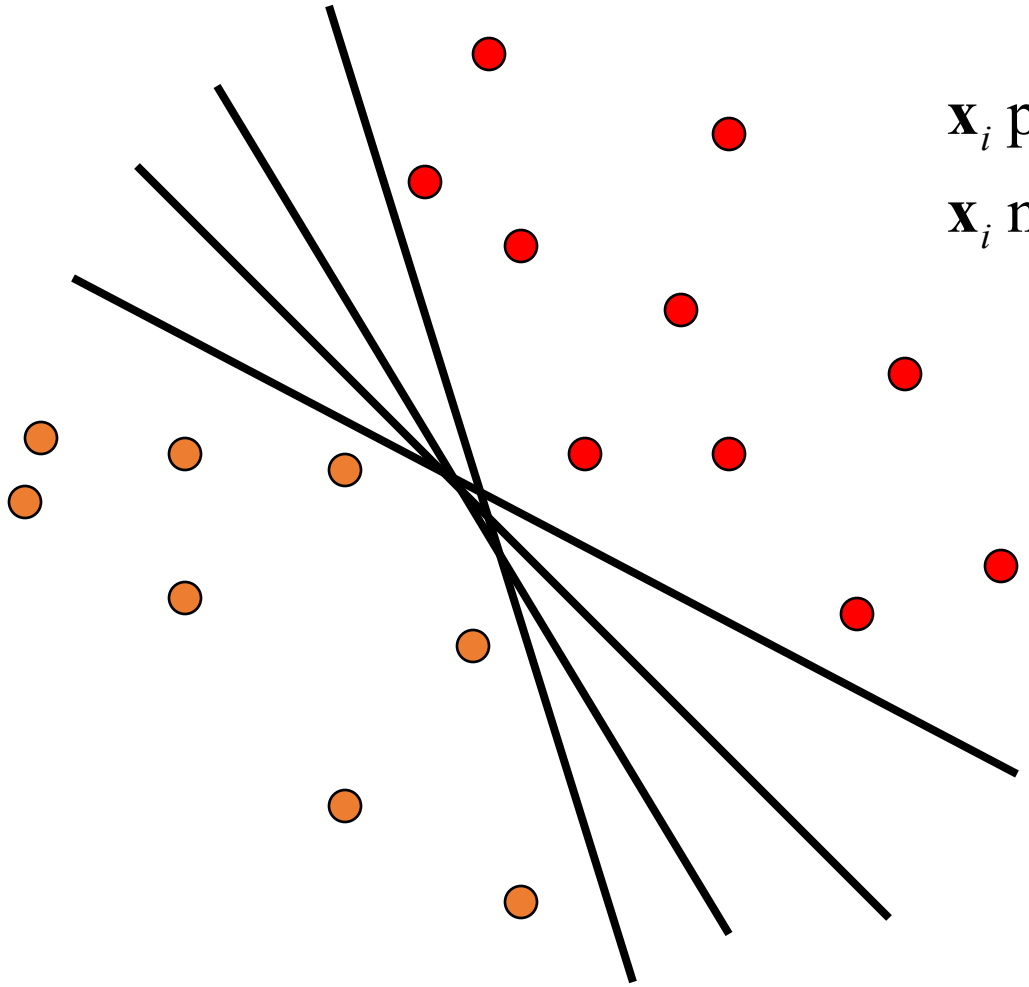
- Binary Classification
  - Logistic Regression
  - **Support Vector Machine (SVM)**
- Clustering
  - K-Means
  - Spectral Clustering

Support Vector Machine (SVM)

Maximal Margin Classification

Linear classifiers

Find linear function (*hyperplane*) to separate positive and negative examples



$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Which hyperplane is best?

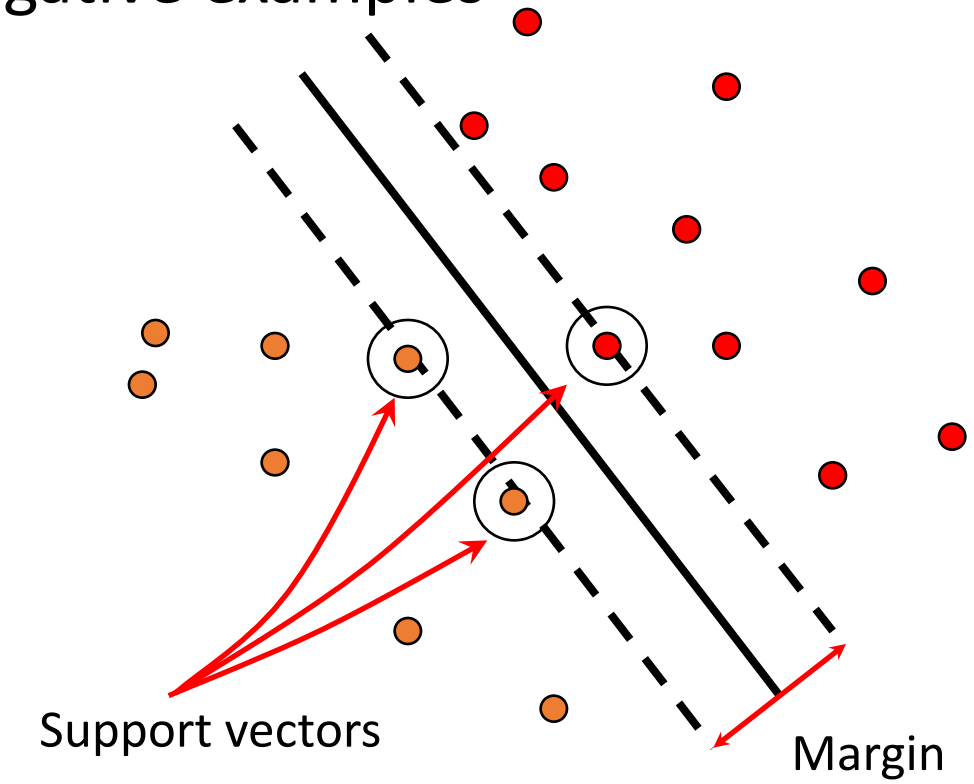
Support vector machines (SVM)

- Find hyperplane that maximizes the *margin* between the positive and negative examples

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Support vector machines (SVM)

- Find hyperplane that maximizes the *margin* between the positive and negative examples



$\mathbf{x}_i$ positive $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support vectors, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance between point and hyperplane: $\dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

Therefore, the margin is $2 / \|\mathbf{w}\|$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

svm.ipynb

Topics for Today

- Binary Classification
  - Logistic Regression
  - Support Vector Machine (SVM)
- **Clustering**
  - K-Means
  - Spectral Clustering

Clustering

Introducing Clustering

- Clustering is an important data mining task.
- Clustering makes it possible to “almost automatically” summarize large amounts of high-dimensional data.
- Clustering (aka segmentation) is applied in many domains: retail, web, image analysis, bioinformatics, psychology, brain science, medical diagnosis etc.

Clustering : Key Idea

- Given: (i) a data set D and (ii) similarity function **s**

Goal: break up D into groups such that items which are similar to each other end in the same group AND items which are dis-similar to each other end up in different groups.

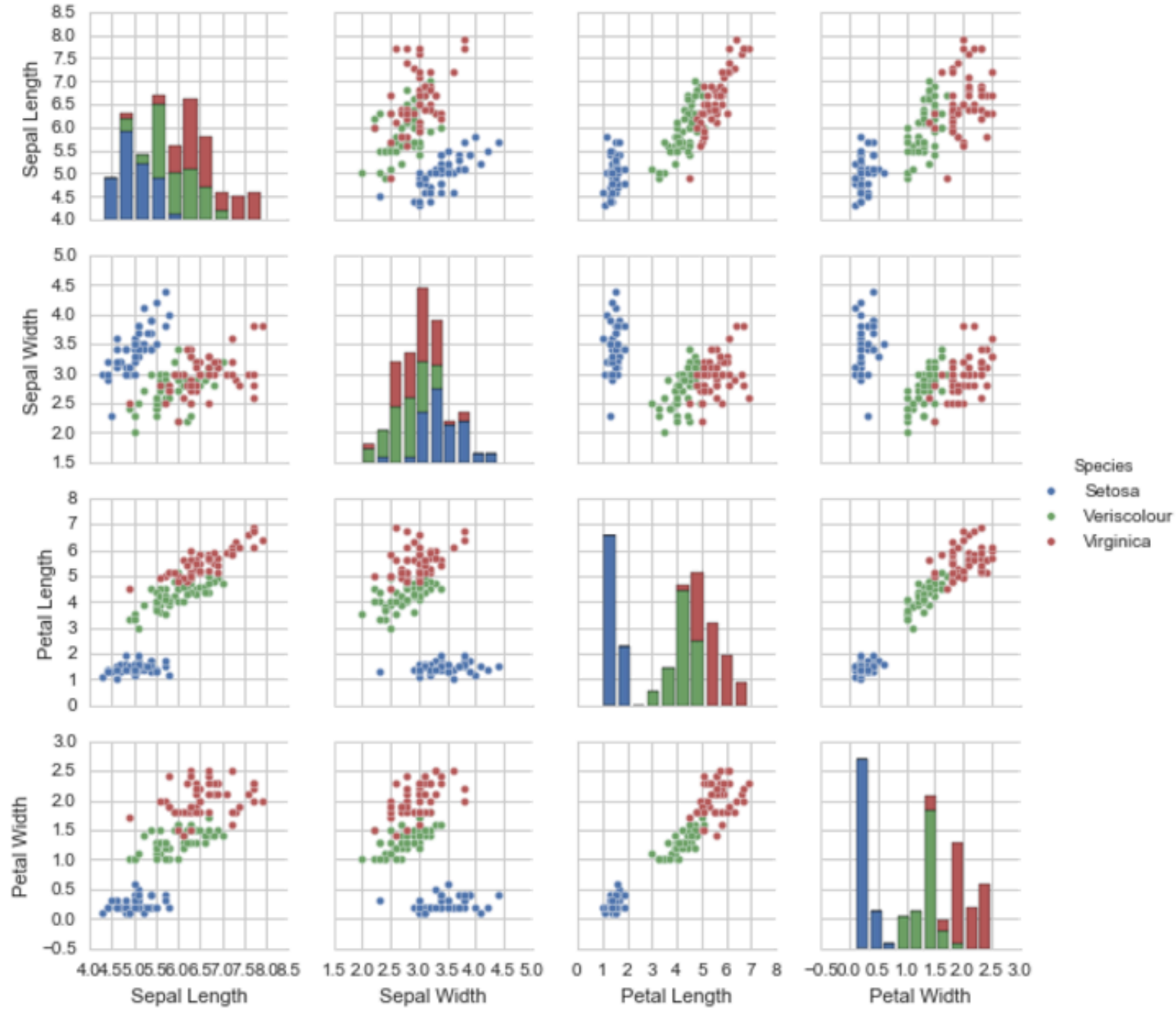
  - Example: customers with similar buying patterns end up in the same cluster.
  - People with similar “biological signals” end in the same cluster.
  - Regions with similar “weather” patterns.

Clustering Example: Iris Data

- Iris data set is a “very small” data set consisting of attributes of 3 flower types.
  - Flower types: Iris_setosa; Iris_versicolor; Iris-virginica
  - 4 attributes: sepal-length; sepal-width; petal-length; petal-width
  - Thus each flower is “data vector” in a four-dimensional space. The “fifth” dimension is a label.
  - Example: [5.1,3.5,1.4,0.2,iris-setosa]

  - It is hard to visualize in four-dimensions!

Which is the “best” pair of attributes to distinguish the three flower types ?



Attributes and Features

- In almost all cases the labels are not given but we can acquire attributes and construct new features.

- Notice the terminology: attributes, features, variables, dimensions – they are often used interchangeably.

- I prefer to distinguish between attributes and features.
  - We acquire attributes and construct features. Thus there are finitely many attributes but infinitely many possible features.
  - For example, Iris data set had four attributes. Lets construct a new feature: sepal-length/sepal-width.

Similarity and Distance Functions

- We have already seen the cosine similarity function.
  - sim(x,y) = (x.y)/||x|||y||

- We can also define a "distance" function which in some sense is like an "opposite" of similarity.
  - Entities which are highly similar to each other have a small distance between them; while items which are less similar have a large distance between them.
- Example: dist(x,y) = 1 – sim(x,y)
  - "identical" objects: sim(x,y) = 1 → dist(x,y) = 0
  - sim(x,y) = 0 → dist(x,y) = 1

Euclidean Distance

- Take two data vectors d1 and d2

- D1 = (x1,x2,…,xn)
- D2 = (y1,y2,…yn)

- Then Euclidean Distance between D1 and D2 is

$$dist(D1, D2) = \sqrt{(x1 - y1)^2 + (x2 - y2)^2 + \ldots + (xn - yn)^2}$$

Euclidean Distance: Examples

- D1=(1,1,1); D2=(1,1,1)
  - dist(D1,D2) = 0

- D1=(1,2,1); D2= (2,1,1)

$$dist(D1, D2) = \sqrt{(1-2)^2 + (2-1)^2 + (1-1)^2} = 1.34$$

Simple Clustering: One-dimensional/one-centroid

- Three data points: 1,2 and 6
- Want to summarize with one point
- The average of (1,2,6) = 9/3 = 3
- 3 is called the centroid

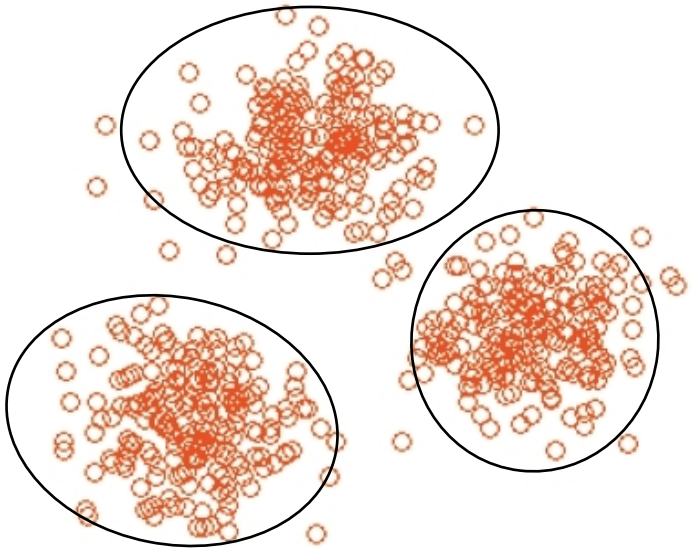
K-means algorithm

```
Let C = initial k cluster centroids
Mark C as unstable
While <C is unstable>
    Assign all data points to their nearest centroid in C.
    Compute the centroids of the points assigned to each
        element of C.
    Update C as the set of new centroids.
    Mark C as stable or unstable by comparing with previous
        set of centroids.
End While
```

Where : n:num of points; k: num of clusters; d: dimension; I: num of iterations
Complexity is linear in n.

When does K-means work?

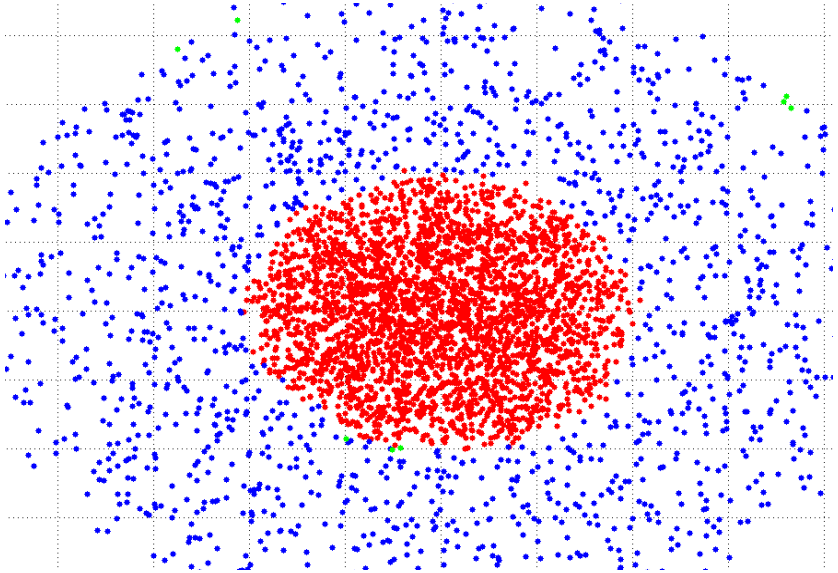
- K-means works perfectly when clusters are **“linearly separable”**
- Clusters are compact and well separated

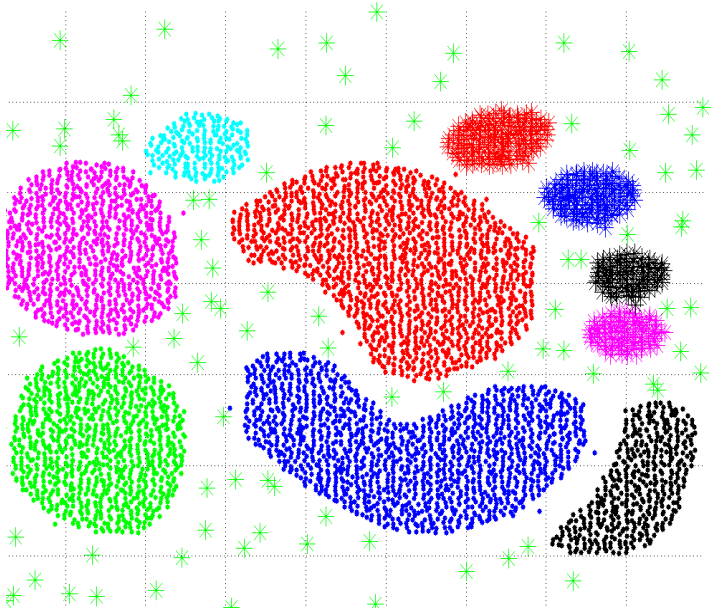


When does K-means not work?

When clusters are “not-linearly separable”

Data contains arbitrarily shaped clusters of different densities

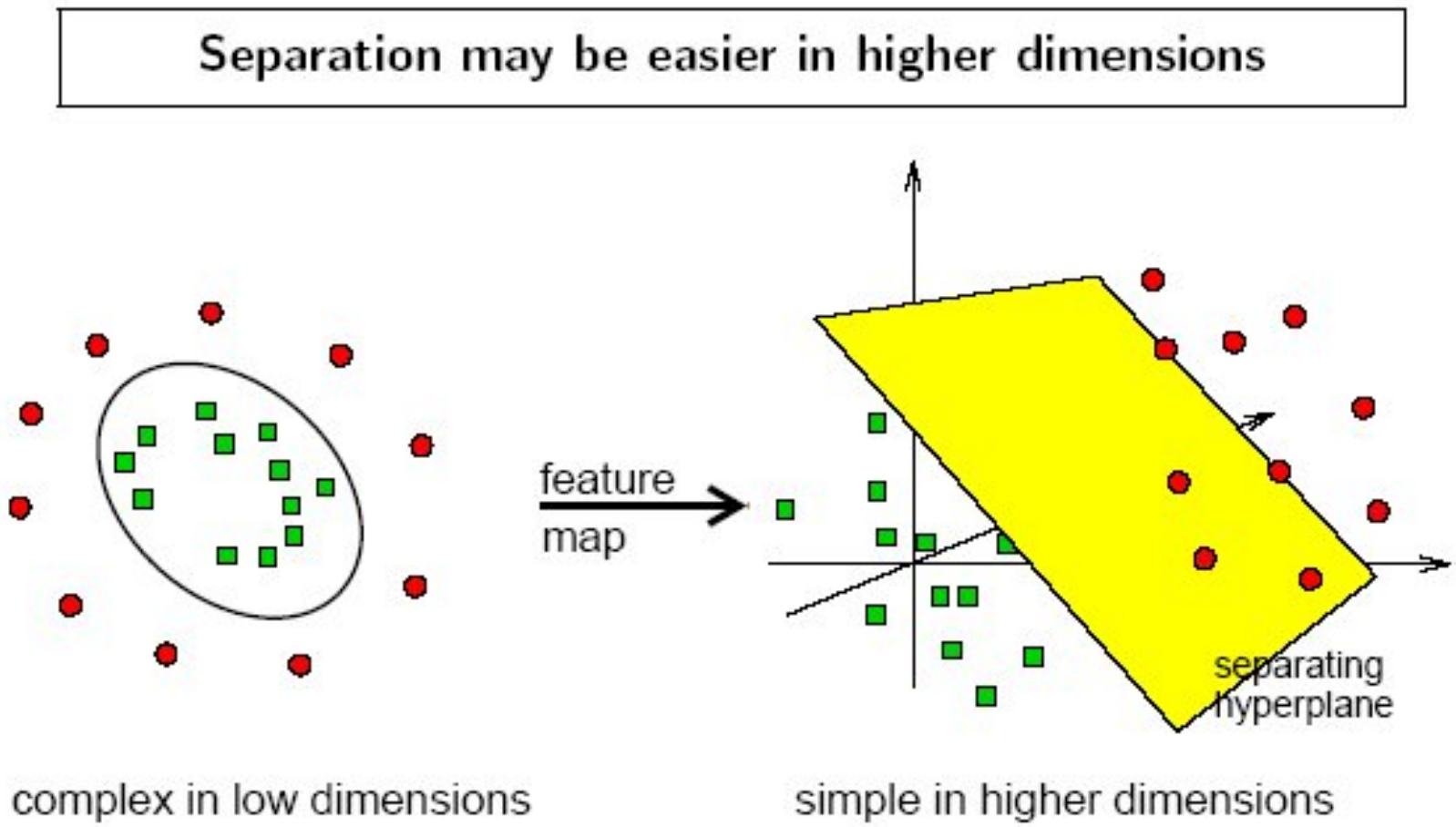




More Advanced Clustering

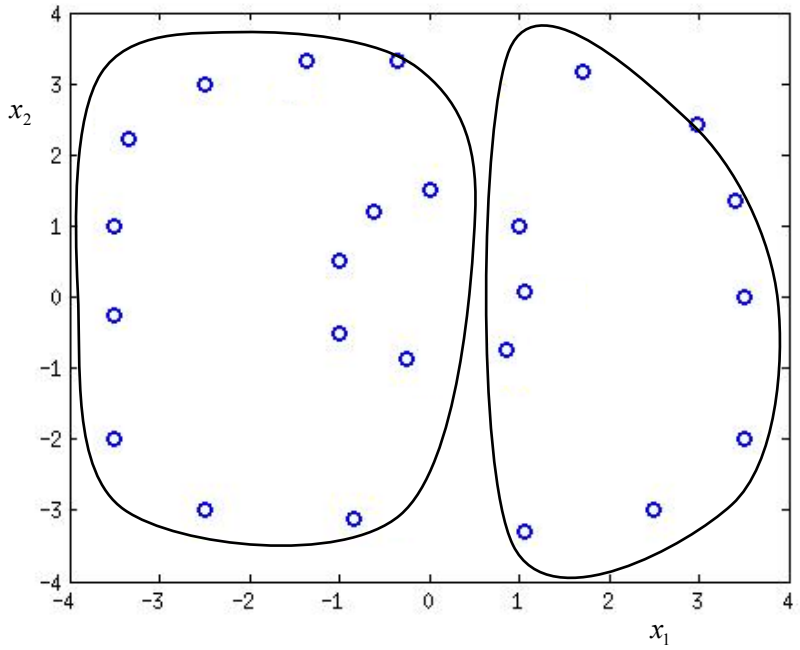
- Kernel K-means
  - K-means applied in Kernel space
- Spectral clustering
  - Eigen subspace of the affinity matrix (Kernel matrix)

The Kernel Trick Revisited



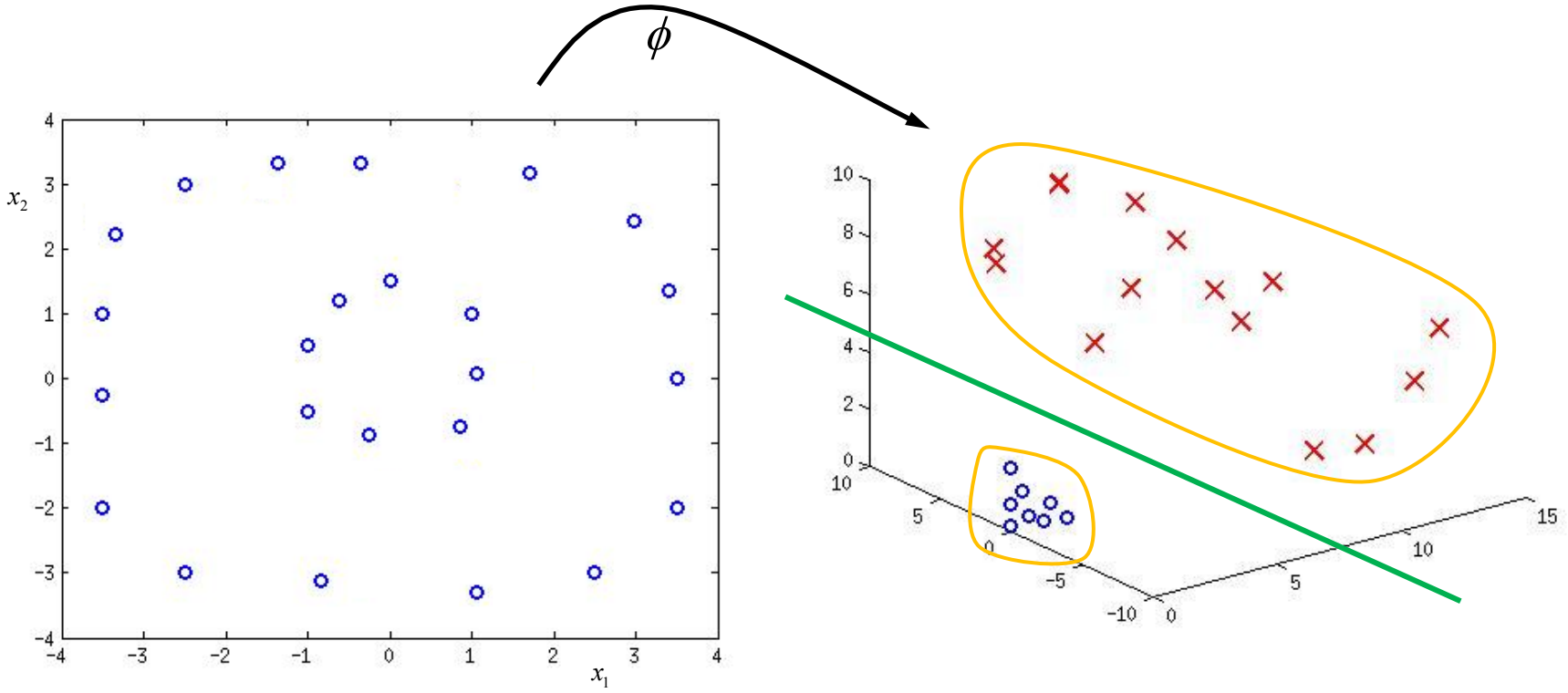
Example

$k = 2$



Data with clusters

Example



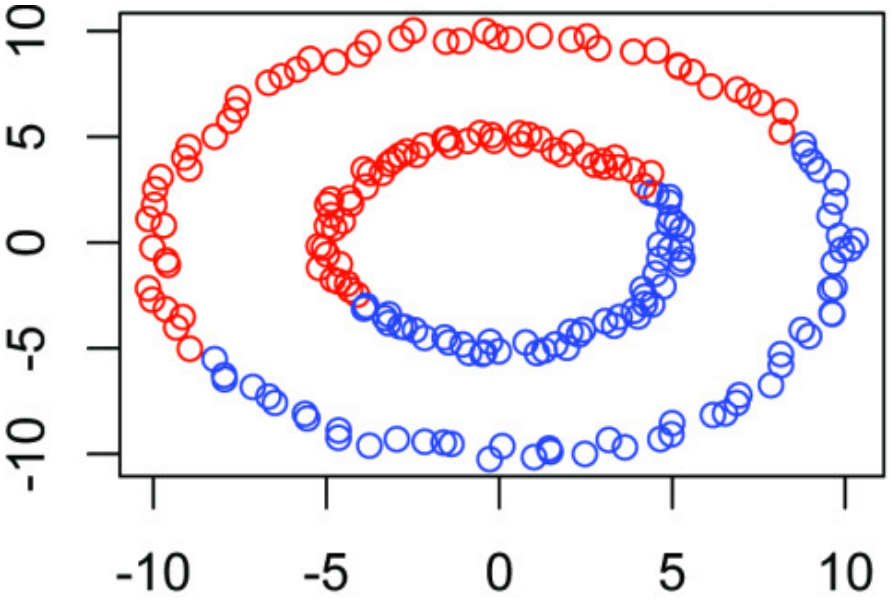
Kernel k-means

$$\text{Polynomial kernel } K(x, y) = (x' y)^2$$

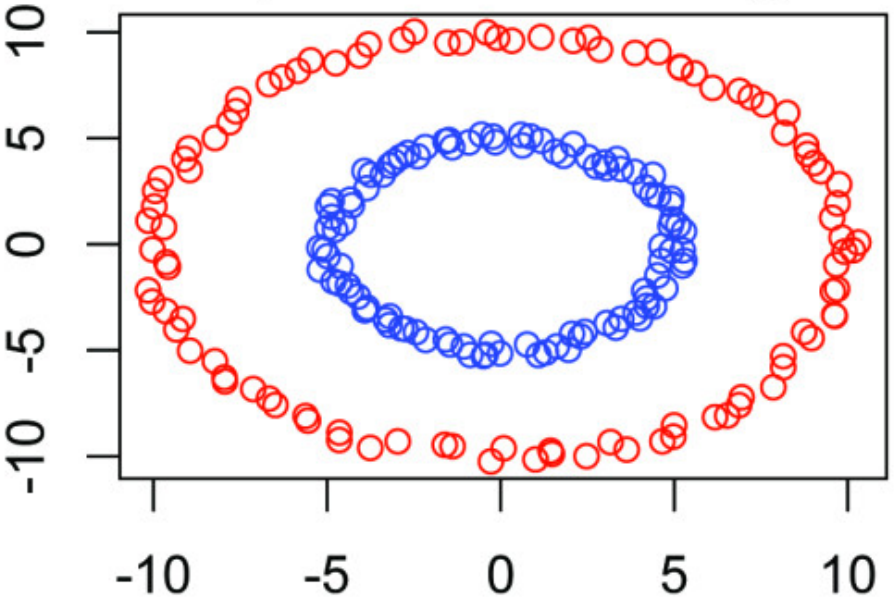
$$\phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

k-means Vs. Kernel k-means

k-means



Kernel k-means



kmeans.ipynb

Performance of Kernel K-means

Clustering accuracy (%) achieved by each clustering algorithm for 10 data sets

Data set	Conventional				Kernel			
	k-means	FCM	Average	Mountain	*k*-means	FCM	Average	Mountain
BENSAID	79.59	73.47	100.0	85.71	83.67	93.88	100.0	100.0
DUNN	70.00	70.00	100.0	83.33	71.11	95.56	100.0	100.0
IRIS	89.33	89.33	90.67	52.67	96.00	93.33	89.33	93.33
ECOLI	42.86	49.11	76.49	51.19	68.75	61.01	77.38	69.05
CIRCLE	50.76	52.79	62.44	55.84	100.0	93.40	82.74	62.94
BLE-3	65.67	65.67	56.00	70.33	76.33	74.67	100.0	71.67
BLE-2	88.50	87.75	100.0	85.25	100.0	94.00	100.0	100.0
UE-4	77.25	66.00	71.45	73.50	100.0	98.50	100.0	84.75
UE-3	95.83	95.00	100.0	51.17	98.83	96.67	100.0	95.67
ULE-4	76.25	94.75	76.25	96.25	98.00	96.25	100.0	96.25
Avg. (%)	73.60	74.39	83.33	70.52	89.27	89.73	94.95	87.37

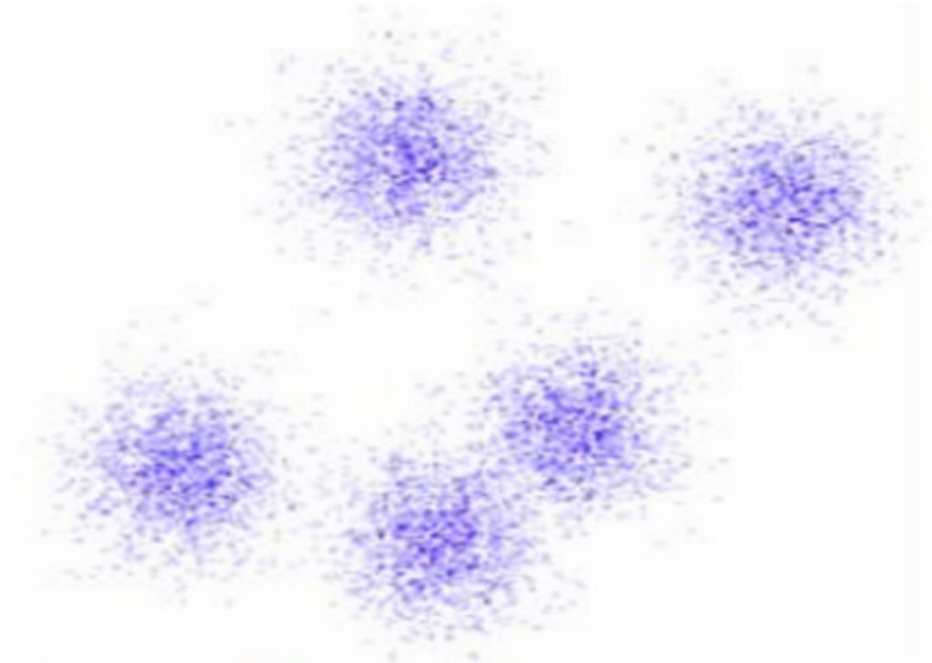
Evaluation of the performance of clustering algorithms in kernel-induced feature space, Pattern Recognition, 2005

Limitations of Kernel K-means

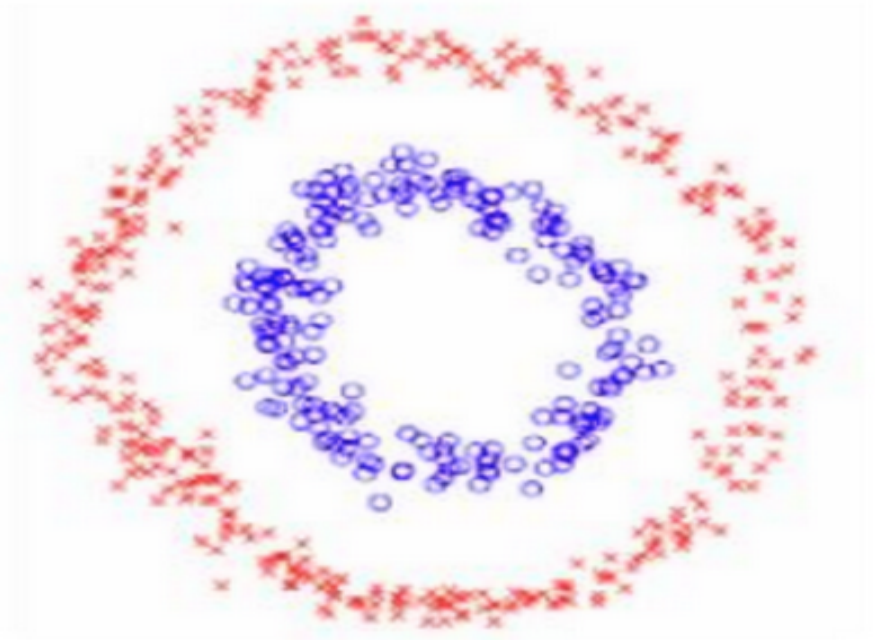
- More complex than k-means
  - Need to compute and store n x n kernel matrix
- What is the largest n that can be handled?
  - Intel Xeon E7-8837 Processor (Q2'11), Oct-core, 2.8GHz, 4TB max memory
  - < 1 million points with "single" precision numbers
  - May take several days to compute the kernel matrix alone
- Use distributed and approximate versions of kernel k-means to handle large datasets

Motivation

- Compactness, e.g., k-means, mixture models
- Connectivity, e.g., spectral clustering



Compactness

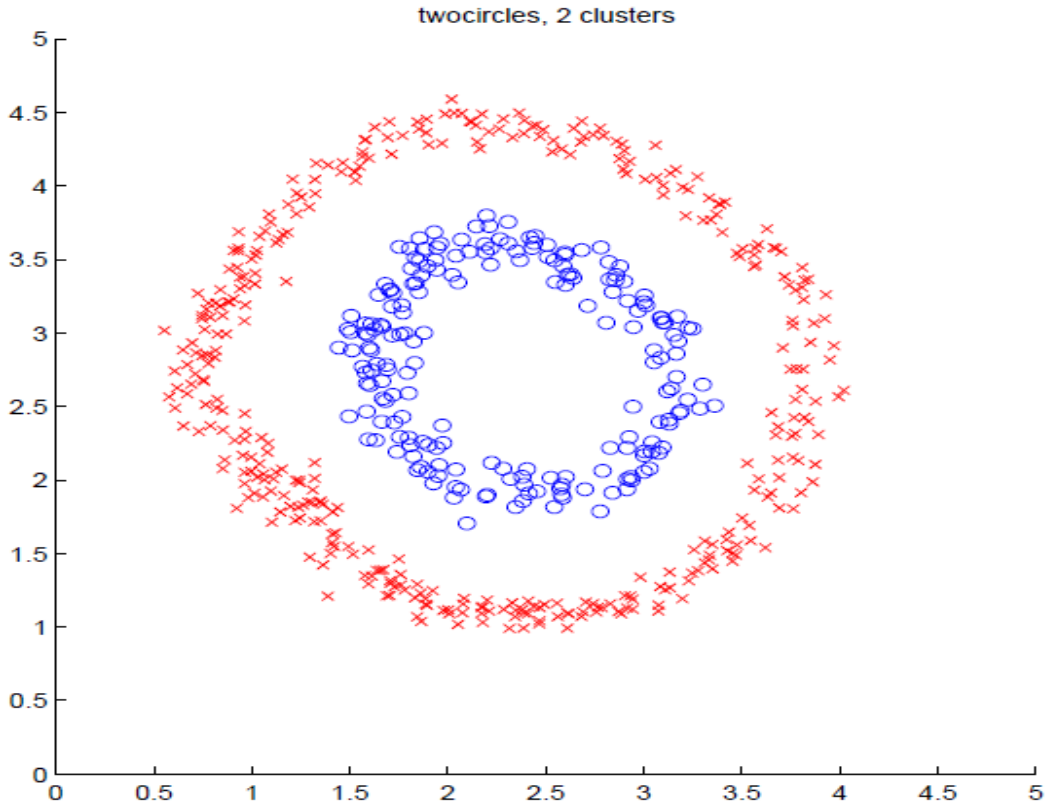


Connectivity

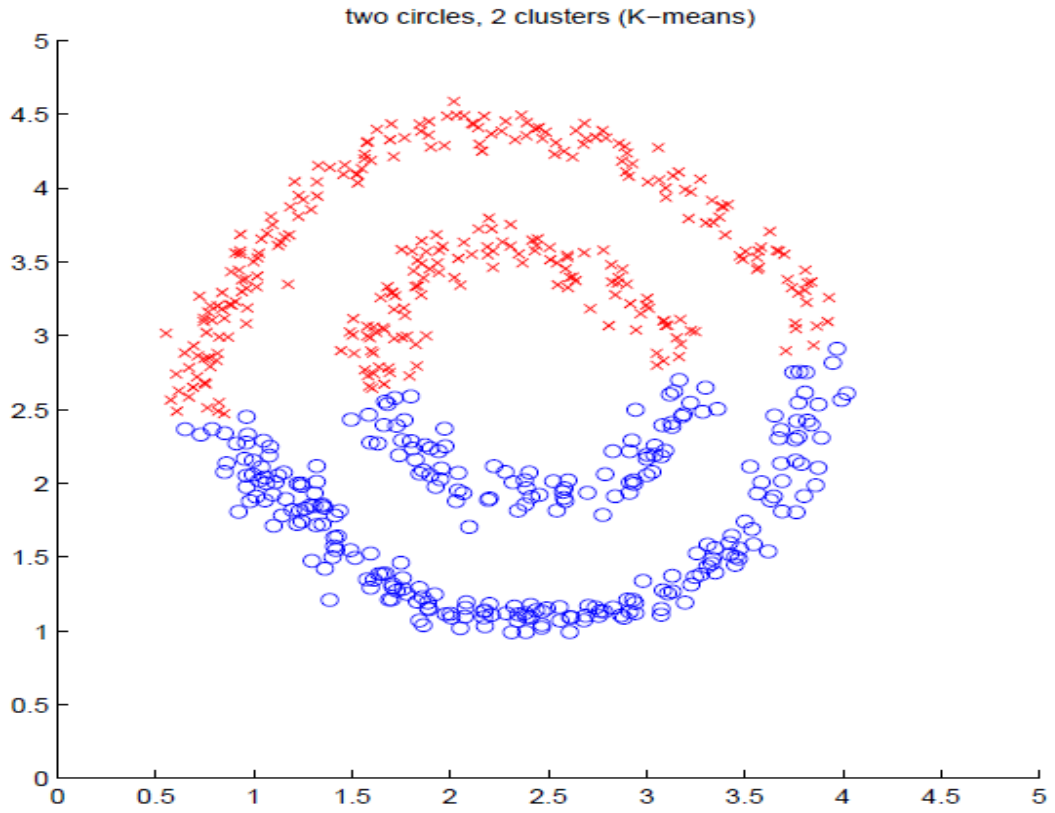
http://charlesmartin14.wordpress.com/2012/10/09/spectral-clustering/

Spectral Clustering & K-means

Spectral Clustering

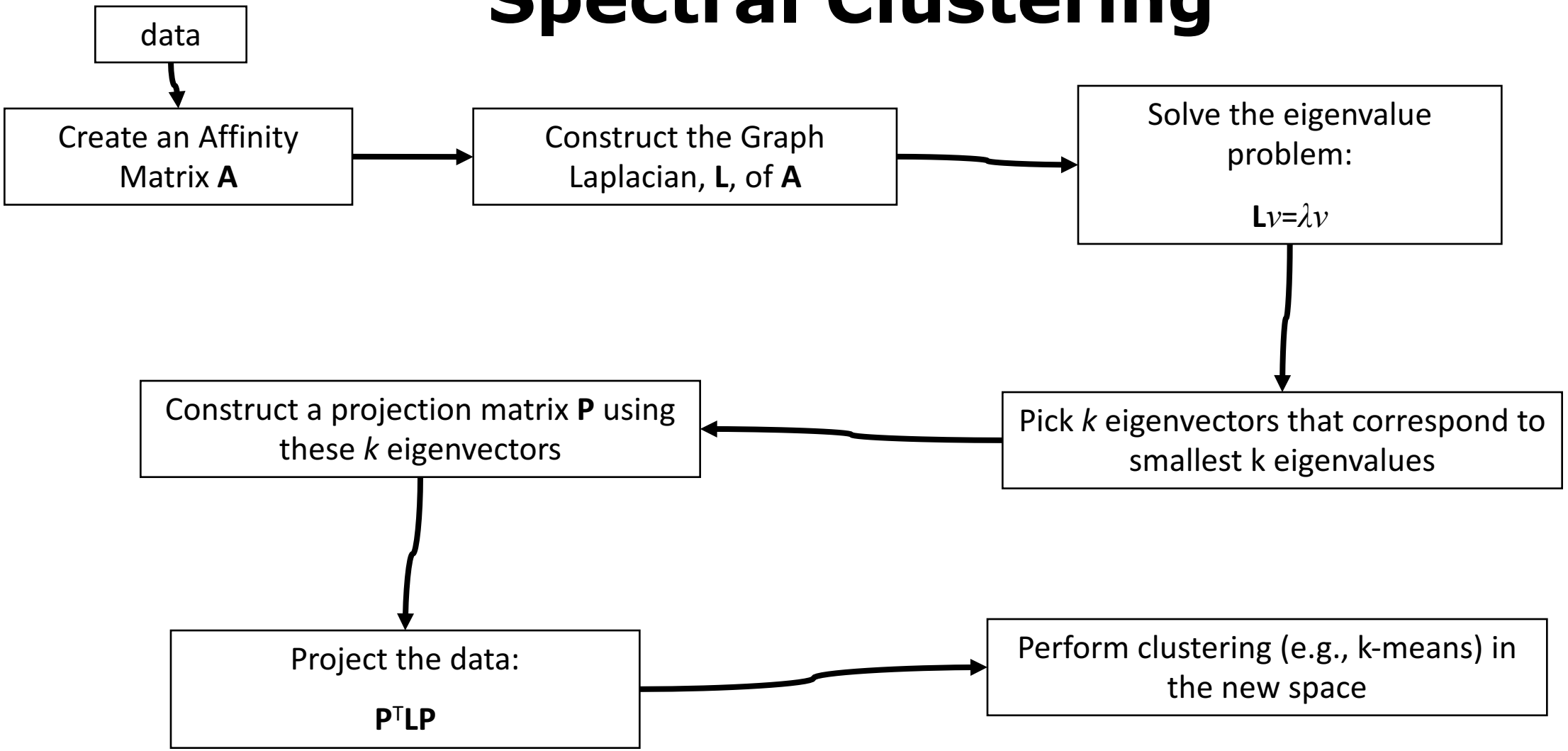


K-means Clustering



Ng et al., NIPS 2001

Spectral Clustering



Affinity (Similarity matrix)

Some examples

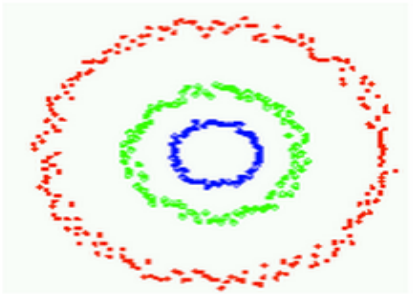
1. The ε-neighborhood graph: Connect all points whose pairwise distances are smaller than ε
2. K-nearest neighbor graph: connect vertex v_m to v_n if v_m is one of the *k*-nearest neighbors of v_n.
3. The fully connected graph: Connect all points with each other with positive (and symmetric) similarity score, e.g., Gaussian similarity function:

$$\exp(-||x_i - x_j||^2/(2\sigma^2))$$

http://charlesmartin14.files.wordpress.com/2012/10/mat1.png

Laplacian Matrix

- For good clustering, we expect to have block diagonal Laplacian matrix



OR

$$L = \begin{bmatrix} L_1 & & 0 \\ & L_2 & \\ 0 & & L_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

First three eigenvectors

http://charlesmartin14.wordpress.com/2012/10/09/spectral-clustering/

Laplacian Matrix

- Given a graph *G* with *n* vertices, its n x n Laplacian matrix L is defined as:

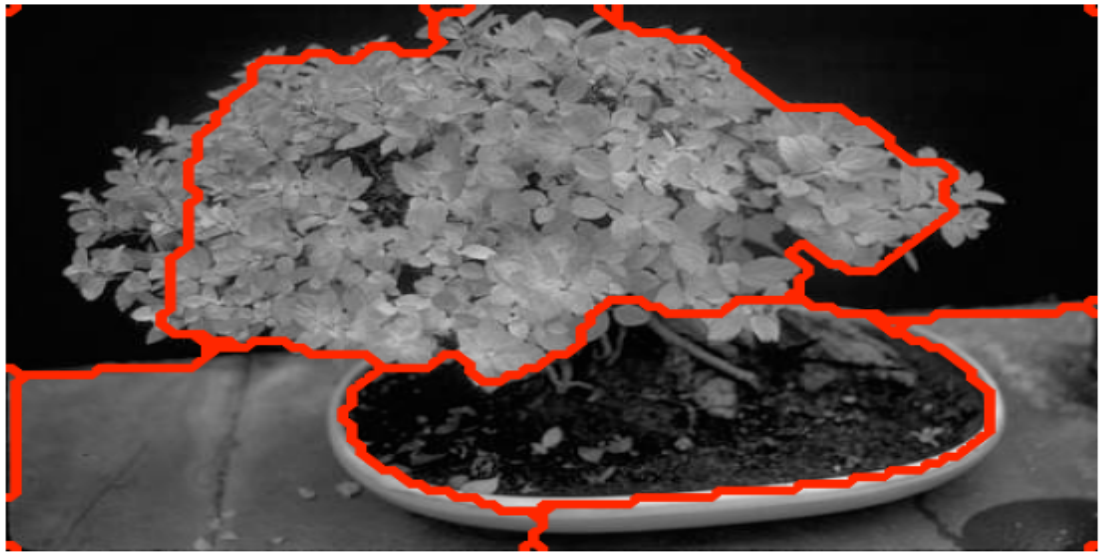
$$L = D - A$$

- L is the difference of the degree matrix *D* and the adjacency matrix *A* of the graph
- Spectral graph theory studies the properties of graphs via the eigenvalues and eigenvectors of their associated graph matrices: adjacency matrix and the graph Laplacian and its variants
- The most important application of the Laplacian is spectral clustering that corresponds to a computationally tractable solution to the graph partitioning problem

Application: social Networks

- Corporate email communication (Adamic and Adar, 2005)

Application: Image Segmentation



(from Zelnik-Manor/Perona, 2005)