



# Machine learning: lecture 16

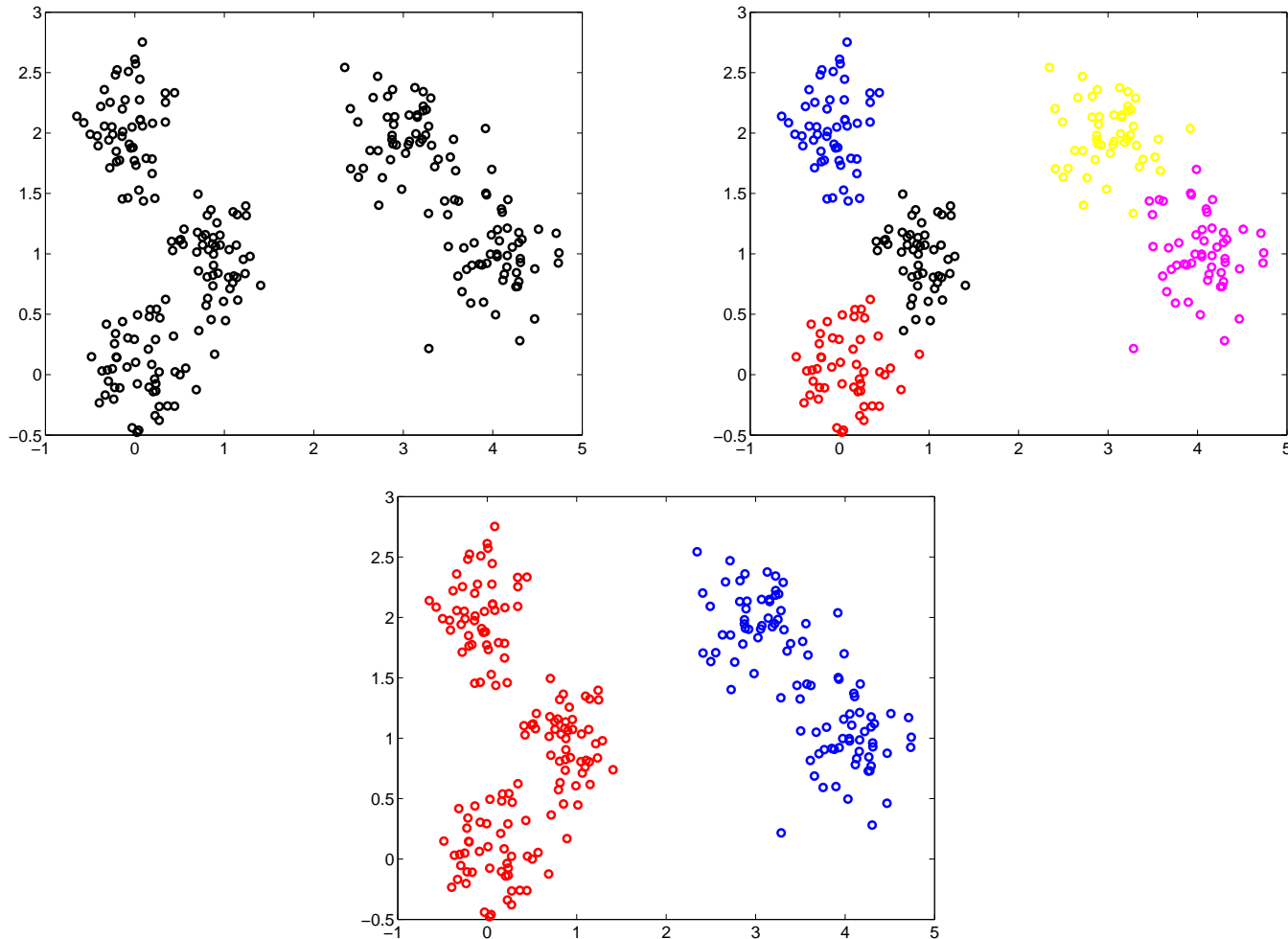
Tommi S. Jaakkola

MIT CSAIL

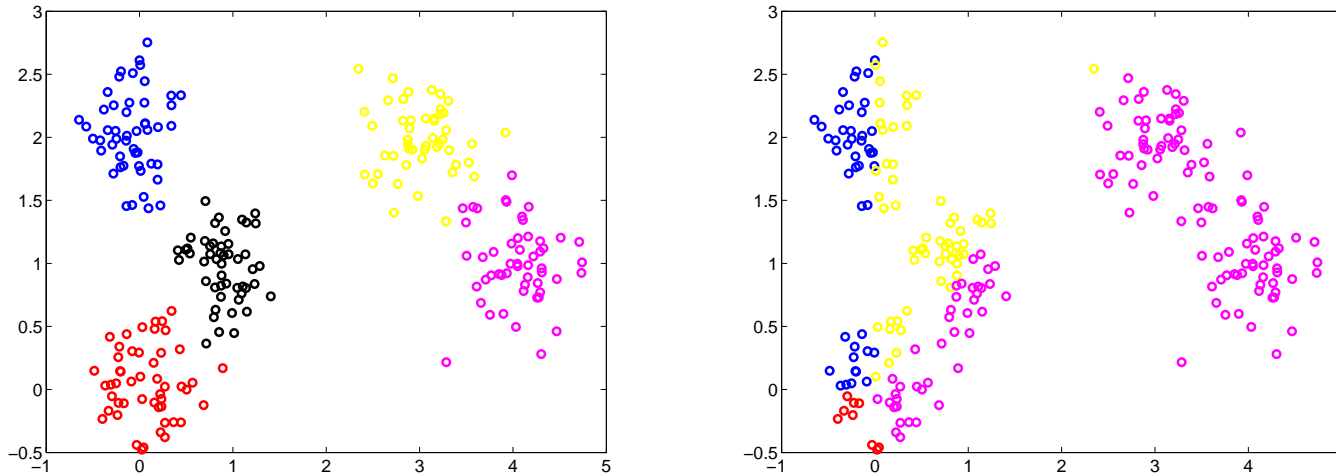
*tommi@csail.mit.edu*

# Finding structure in the data: clustering

- We can find structure in the data by isolating groups of examples that are similar in some well-defined sense



# Clustering: metric



- Clustering results are crucially dependent on the measure of similarity (or distance) between the “points” to be clustered

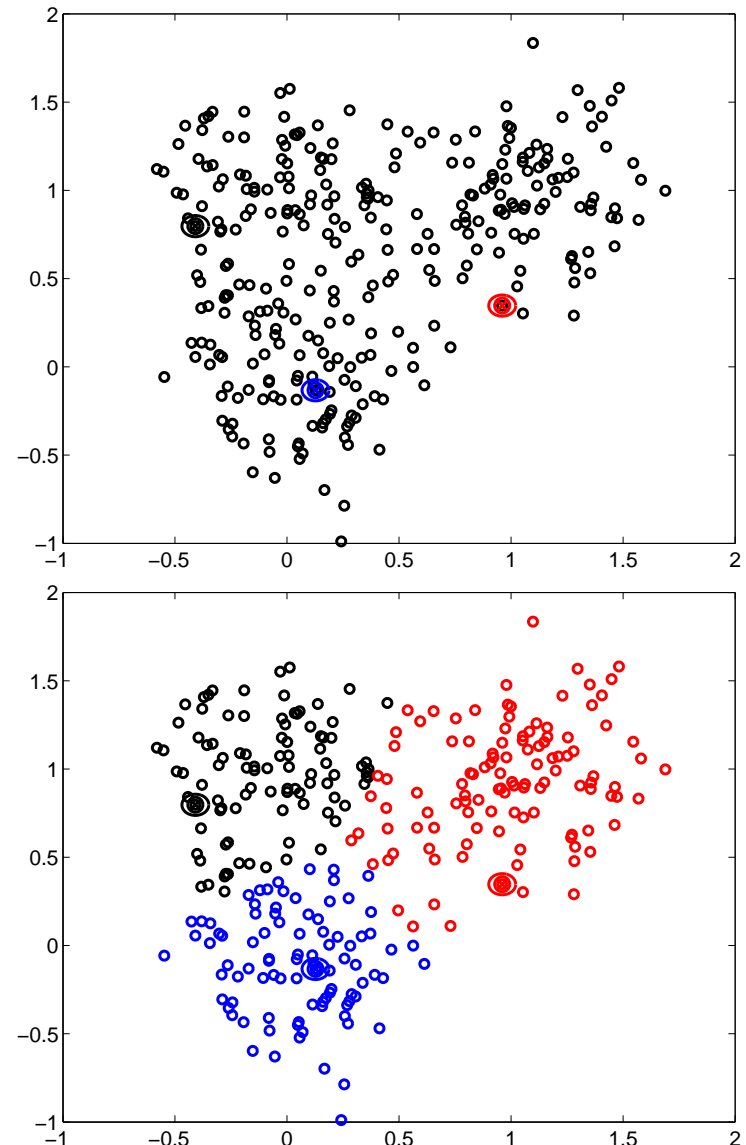


# Overview of clustering methods

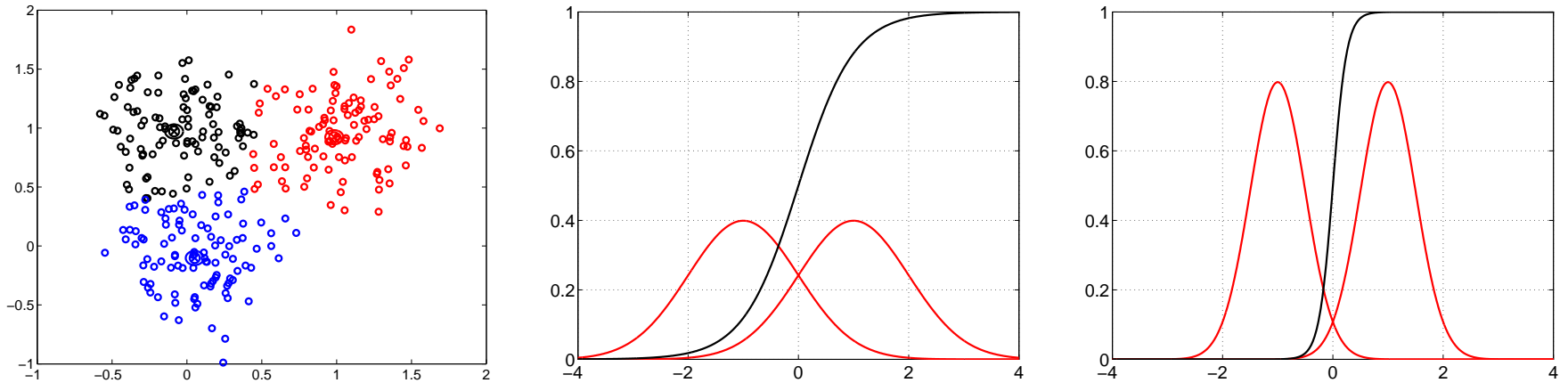
- Flat clustering methods
  - e.g., mixture models, k-means clustering
- Hierarchical clustering methods:
  - Top-down (splitting)
    - \* e.g., hierarchical mixture models
  - Bottom-up (merging)
    - \* e.g., hierarchical agglomerative clustering
- Spectral clustering
- Semi-supervised clustering
- Clustering by dynamics
- Etc.

# K-means clustering

- The procedure:
  1. Pick  $k$  arbitrary centroids (cluster means)
  2. Assign each example to its “closest” centroid (**E-step**)
  3. Adjust the centroids to be the means of the examples assigned to them (**M-step**)
  4. Goto step 2 (until no change)
- The algorithm is guaranteed to converge in a finite number of iterations

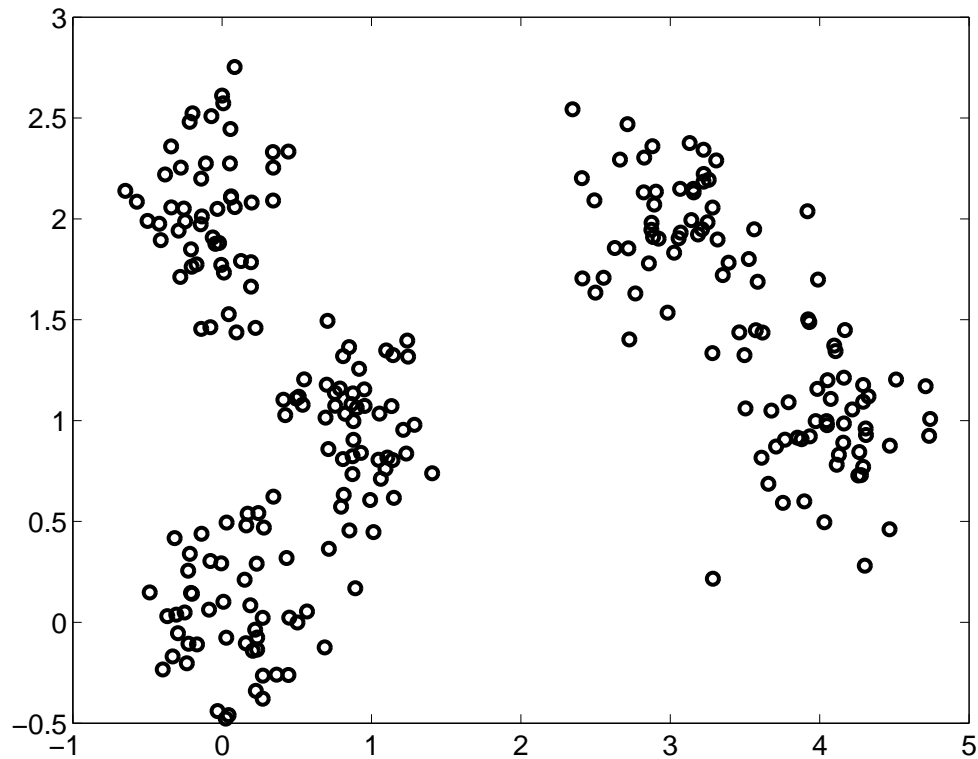


# K-means clustering cont'd



- K-means clustering corresponds to estimating a Gaussian mixture model with EM provided that the covariance matrices are fixed,  $\Sigma_j = \sigma^2 I$ , for all  $j = 1, \dots, m$ , and  $\sigma^2$  is small.

# Spectral clustering: motivation





## Spectral clustering: outline

- Spectral clustering (as described here) relies on a random walk over the points

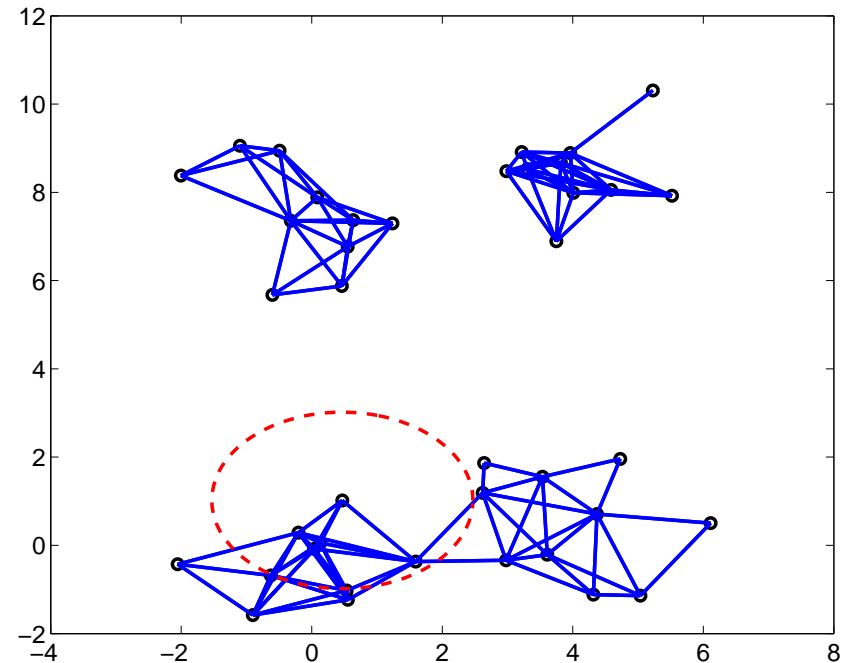
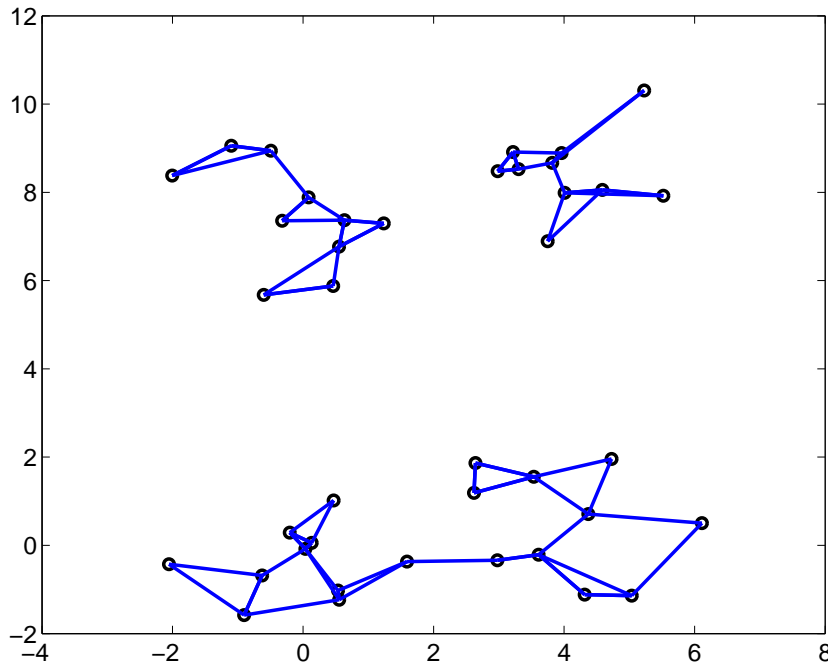
We find the random walk via the following steps

1. construct a neighborhood graph
  2. assign weights to the edges in the graph
  3. define a transition probability matrix based on the weights
- The points are clustered on the basis of the eigenvectors of the resulting transition probability matrix



# Step 1: neighborhood graph

- We can connect each point to its  $k$ -nearest neighbors, or connect each point to all neighbors within distance  $\epsilon$

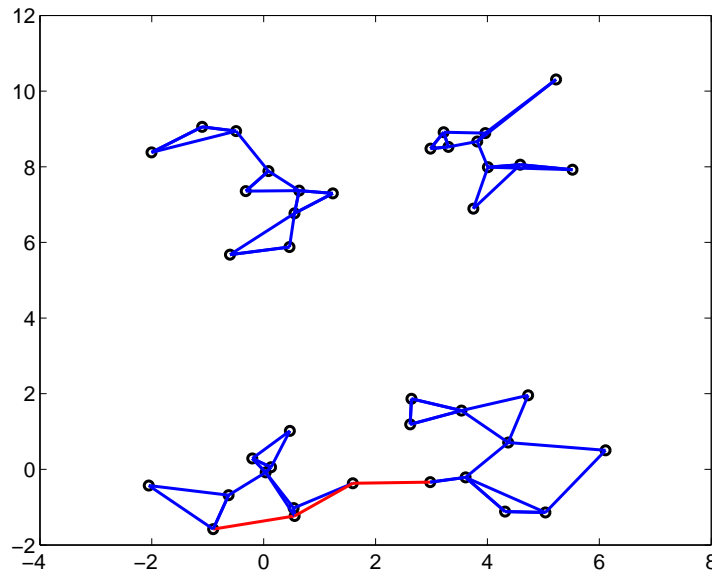


## Step 2: edge weights

- We assign symmetric non-negative edge weights  $W_{ij}$ :

$$W_{ij} = \exp\{-\beta\|\mathbf{x}_i - \mathbf{x}_j\|\}, \text{ if } i \text{ and } j \text{ connected}$$

$$W_{ij} = 0, \text{ otherwise}$$



Note: we do not use a squared distance in the exponent so that a weight for a path is computed analogously to the edge weights

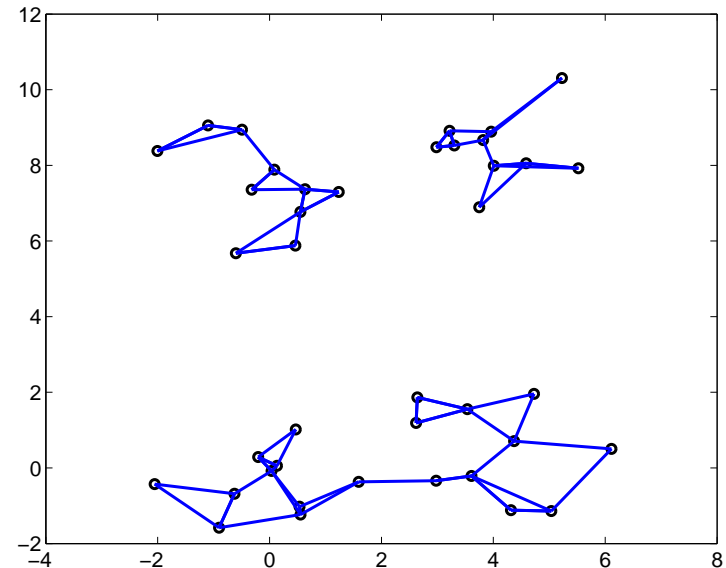
## Step 3: transition probability matrix

- Finally, we define a Markov random walk over the neighborhood graph by constructing a transition probability matrix from the edge weights

$$P_{ij} = \frac{W_{ij}}{W_{i\cdot}}, \quad \text{where } W_{i\cdot} = \sum_j W_{ij}$$

and  $\sum_j P_{ij} = 1$  for all  $i$ .

The random walk proceeds by successively selecting points according to  $j \sim P_{ij}$ , where  $i$  specifies the current location



## Random walk: properties

- If we start from  $i_0$ , the distribution of points  $i_t$  that we end up in after  $t$  steps is given by

$$i_1 \sim P_{i_0 i_1},$$

$$i_2 \sim \sum_{i_1} P_{i_0, i_1} P_{i_1 i_2} = [P^2]_{i_0 i_2},$$

$$i_3 \sim \sum_{i_1} \sum_{i_2} P_{i_0, i_1} P_{i_1 i_2} P_{i_2 i_3} = [P^3]_{i_0 i_3},$$

...

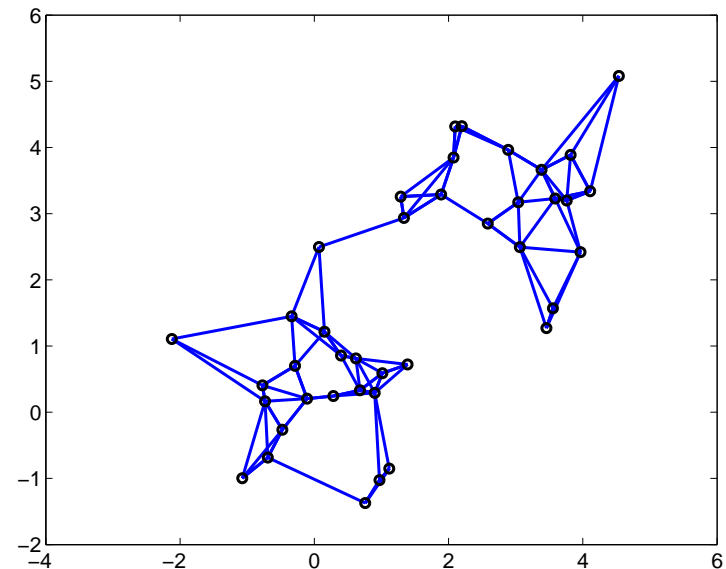
$$i_t \sim [P^t]_{i_0 i_t}$$

where  $P^t = PP \dots P$  ( $t$  matrix products) and  $[\cdot]_{ij}$  denotes the  $i, j$  component of the matrix.

# Random walk and clustering

- The distribution of points we end up in after  $t$  random steps converge as  $t$  increases. If the graph is connected (and ergodic), the resulting distribution becomes independent of the starting point

Even for large  $t$ , the transition probabilities  $[P^t]_{ij}$  have a slightly higher probability of transitioning within “clusters” than across; we want to recover this effect from eigenvalues/vectors





# Eigenvalues/vectors and spectral clustering

- Let  $W$  be the matrix with components  $W_{ij}$  and  $D$  a diagonal matrix such that  $D_{ii} = \sum_j W_{ij}$ . Then

$$P = D^{-1}W$$

- To find out how  $P^t$  behaves for large  $t$  it is useful to examine the eigen-decomposition of the following symmetric matrix

$$D^{-\frac{1}{2}}W D^{-\frac{1}{2}} = \lambda_1 \mathbf{z}_1 \mathbf{z}_1^T + \lambda_2 \mathbf{z}_2 \mathbf{z}_2^T + \dots + \lambda_n \mathbf{z}_n \mathbf{z}_n^T$$

where the ordering is such that  $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$ .

## Eigenvalues/vectors cont'd

- The symmetric matrix is related to  $P^t$  since

$$(D^{-\frac{1}{2}}WD^{-\frac{1}{2}}) \dots (D^{-\frac{1}{2}}WD^{-\frac{1}{2}}) = D^{\frac{1}{2}}(P \dots P)D^{-\frac{1}{2}}$$

This allows us to write the  $t$  step transition probability matrix in terms of the eigenvalues/vectors of the symmetric matrix

$$\begin{aligned} P^t &= D^{-\frac{1}{2}} \left( D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \right)^t D^{\frac{1}{2}} \\ &= D^{-\frac{1}{2}} \left( \lambda_1^t \mathbf{z}_1 \mathbf{z}_1^T + \lambda_2^t \mathbf{z}_2 \mathbf{z}_2^T + \dots + \lambda_n^t \mathbf{z}_n \mathbf{z}_n^T \right) D^{\frac{1}{2}} \end{aligned}$$

where  $\lambda_1 = 1$  and

$$P^\infty = D^{-\frac{1}{2}} \left( \mathbf{z}_1 \mathbf{z}_1^T \right) D^{\frac{1}{2}}$$



# Eigenvalues/vectors and spectral clustering

- We are interested in the largest correction to the asymptotic limit

$$P^t \approx P^\infty + D^{-\frac{1}{2}} \left( \lambda_2^t \mathbf{z}_2 \mathbf{z}_2^T \right) D^{\frac{1}{2}}$$

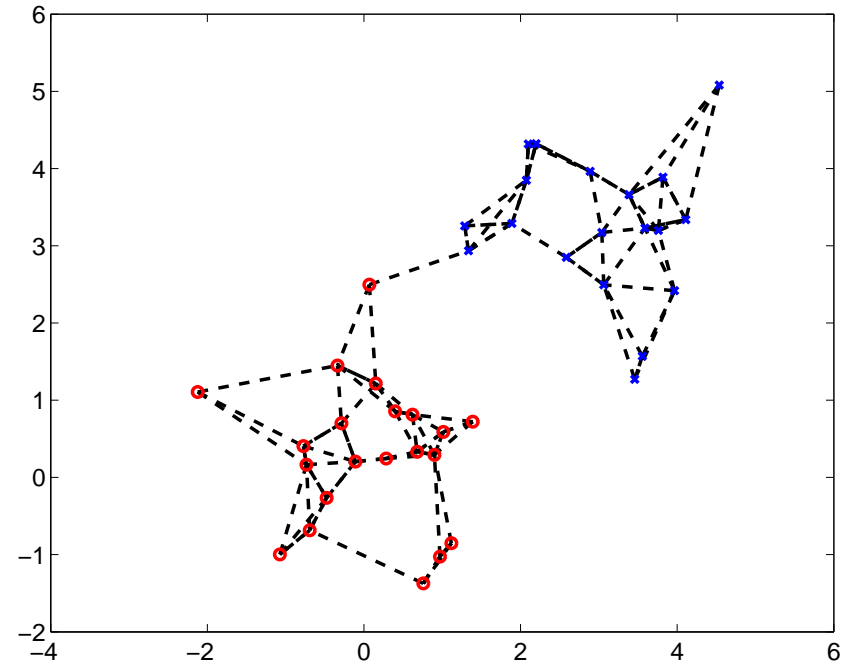
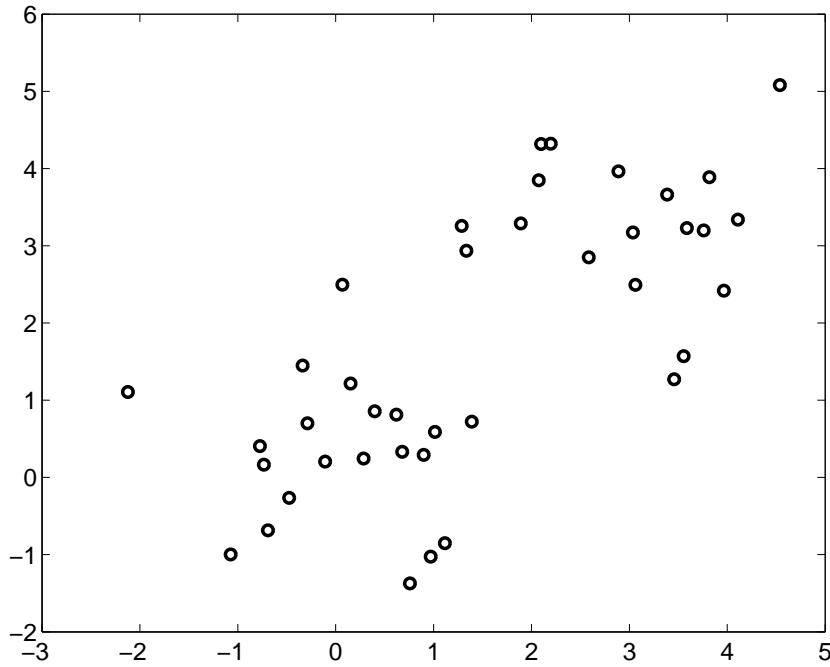
Note:  $[\mathbf{z}_2 \mathbf{z}_2^T]_{ij} = z_{2i} z_{2j}$  and thus the largest correction term increases the probability of transitions between points that share the same sign of  $z_{2i}$  and decreases transitions across points with different signs

- Binary spectral clustering: we divide the points into clusters based on the sign of the elements of  $\mathbf{z}_2$

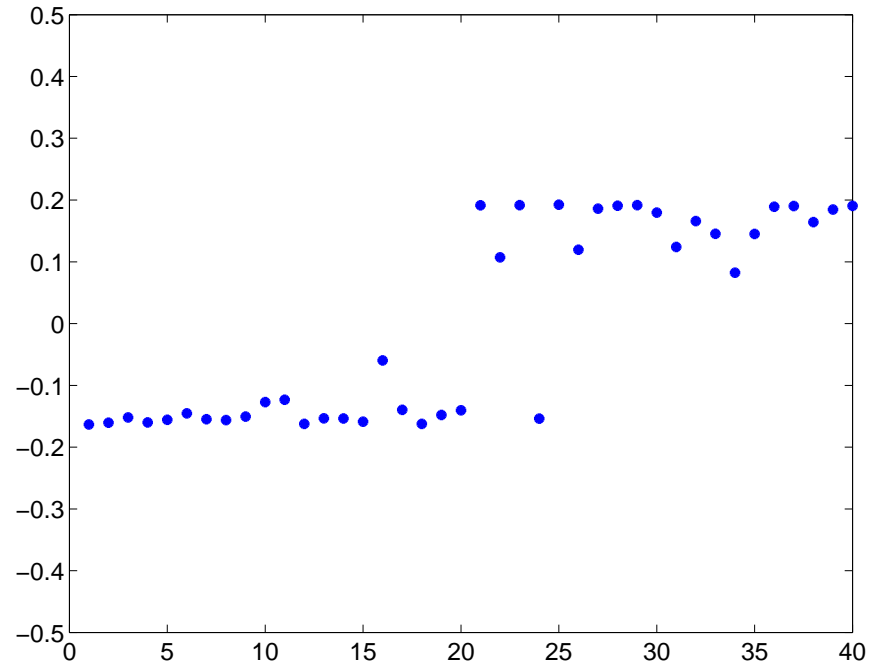
$$z_{2j} > 0 \Rightarrow \text{cluster 1, otherwise cluster 0}$$



# Spectral clustering: example



# Spectral clustering: example cont'd



Components of the eigenvector corresponding to the second largest eigenvalue