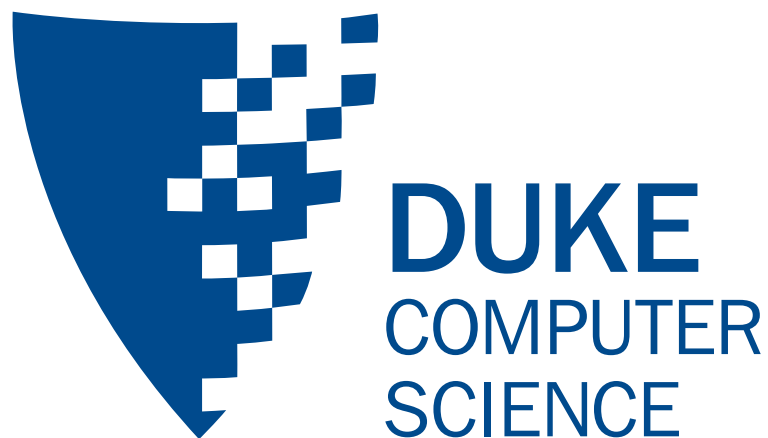


Unsupervised Learning

George Konidaris
gdk@cs.duke.edu



Spring 2016

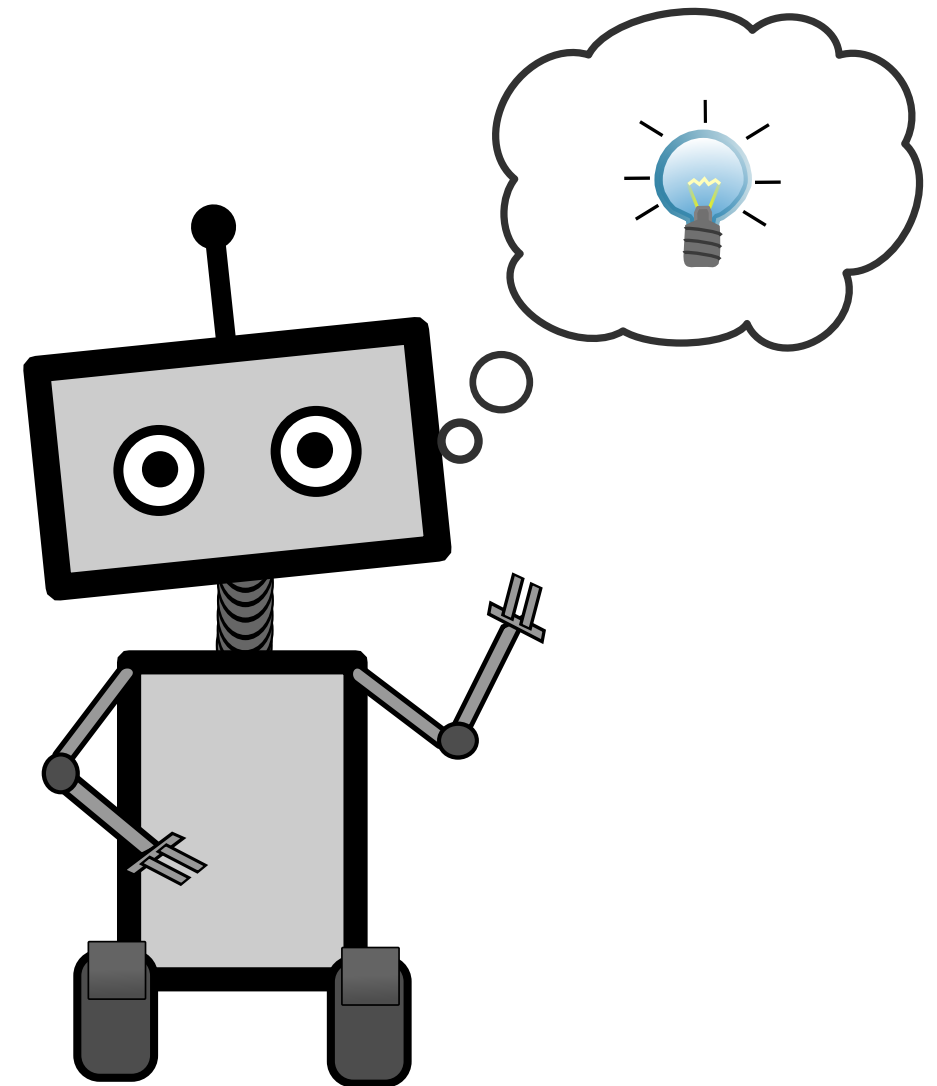
Machine Learning

Subfield of AI concerned with *learning from data*.

Broadly, using:

- *Experience*
- To Improve *Performance*
- On Some *Task*

(Tom Mitchell, 1997)



Unsupervised Learning

Input:

$$X = \{x_1, \dots, x_n\} \quad \text{inputs}$$

Try to understand the
structure of the data.

*E.g., how many types of cars?
How can they vary?*



Clustering

One particular type of unsupervised learning:

- Split the data into discrete clusters.
- Assign new data points to each cluster.
- Clusters can be thought of as *types*.

Formal definition

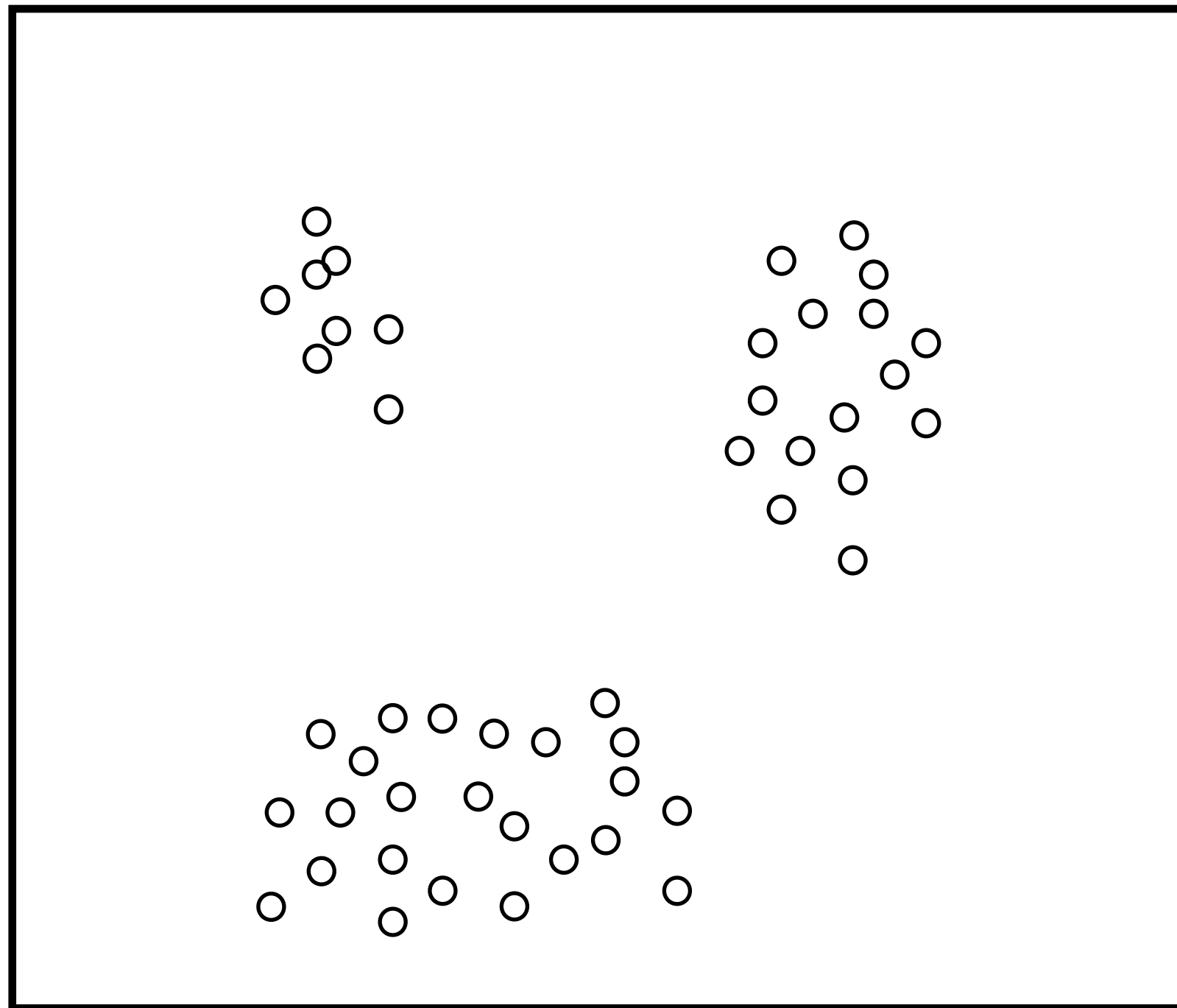
Given:

- Data points $X = \{x_1, \dots, x_n\}$,

Find:

- Number of clusters k
- Assignment function $f(x) = \{1, \dots, k\}$

Clustering

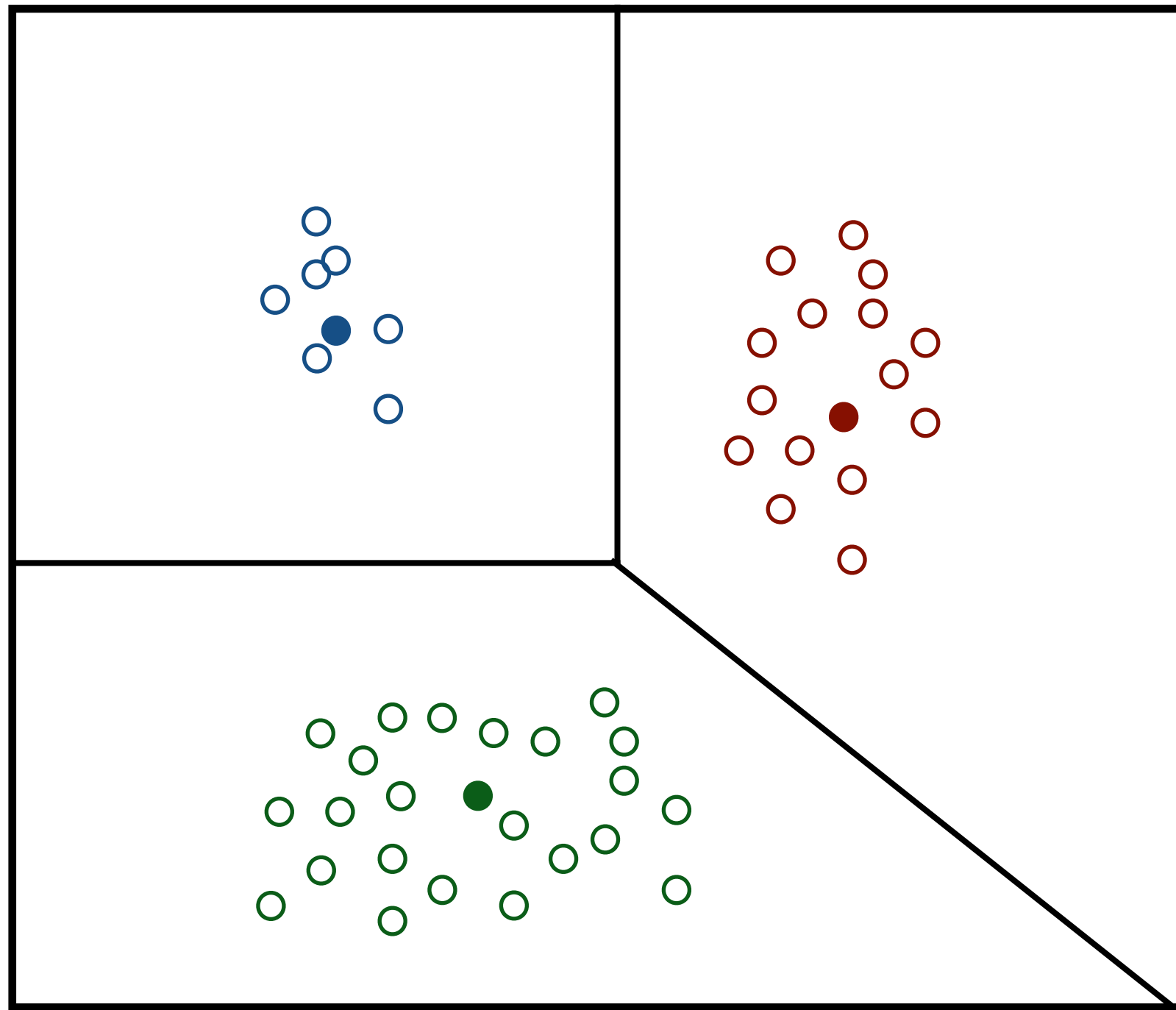


k-Means

One approach:

- Pick k
- Place k points (“means”) in the data
- Assign new point to i th cluster if nearest to i th “mean”.

k-Means



k-Means

Major question:

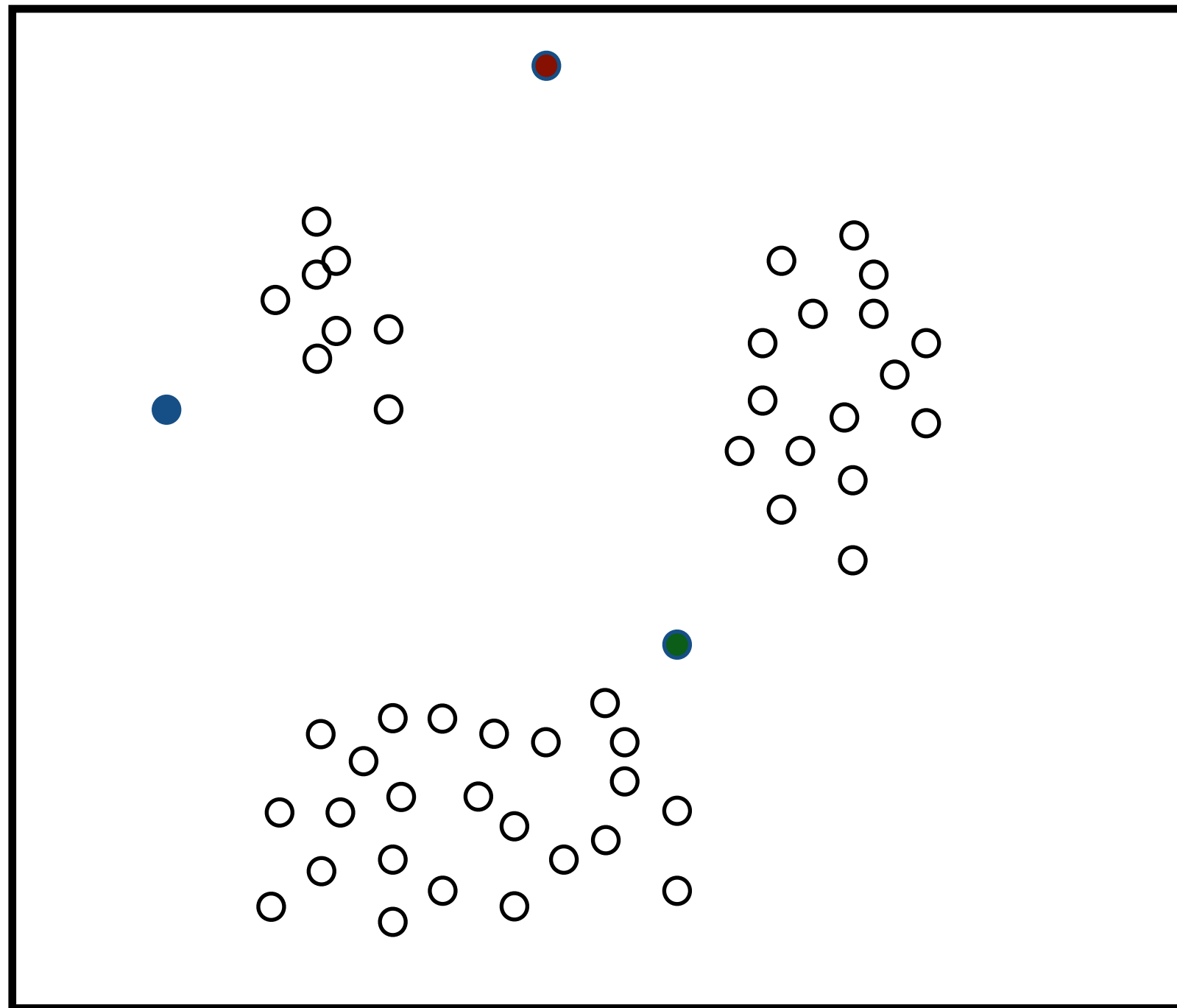
- *Where to put the “means”?*

Very simple algorithm:

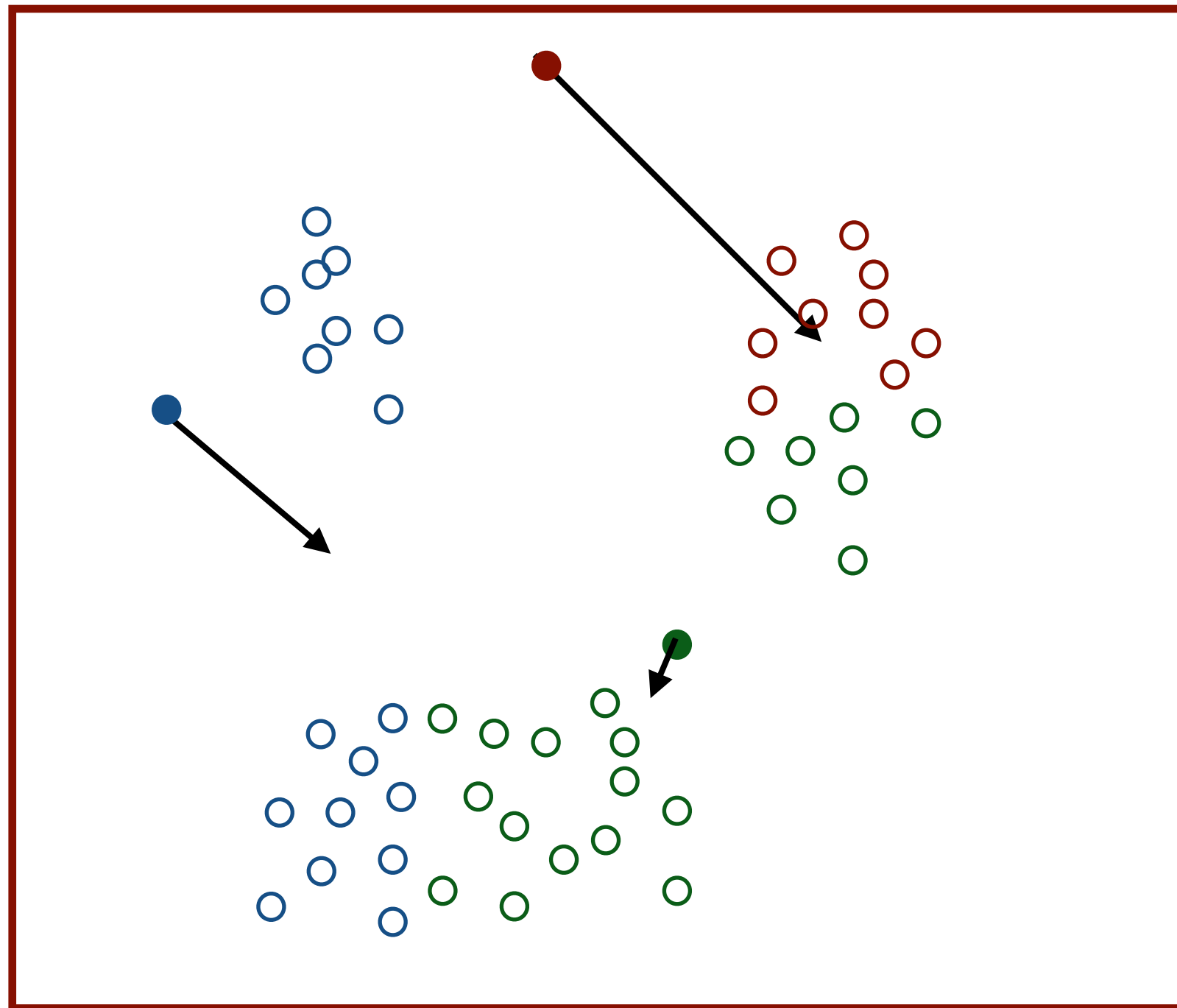
- Place k “means” $\{\mu_1, \dots, \mu_k\}$ at random.
- Assign all points in the data to each “mean”
 $f(x_j) = i$ such that $d(x_j, \mu_i) \leq d(x_j, \mu_l) \forall l \neq i$
- Move “mean” to mean of assigned data.

$$\mu_i = \sum_{v \in C_i} \frac{x_v}{|C_i|}$$

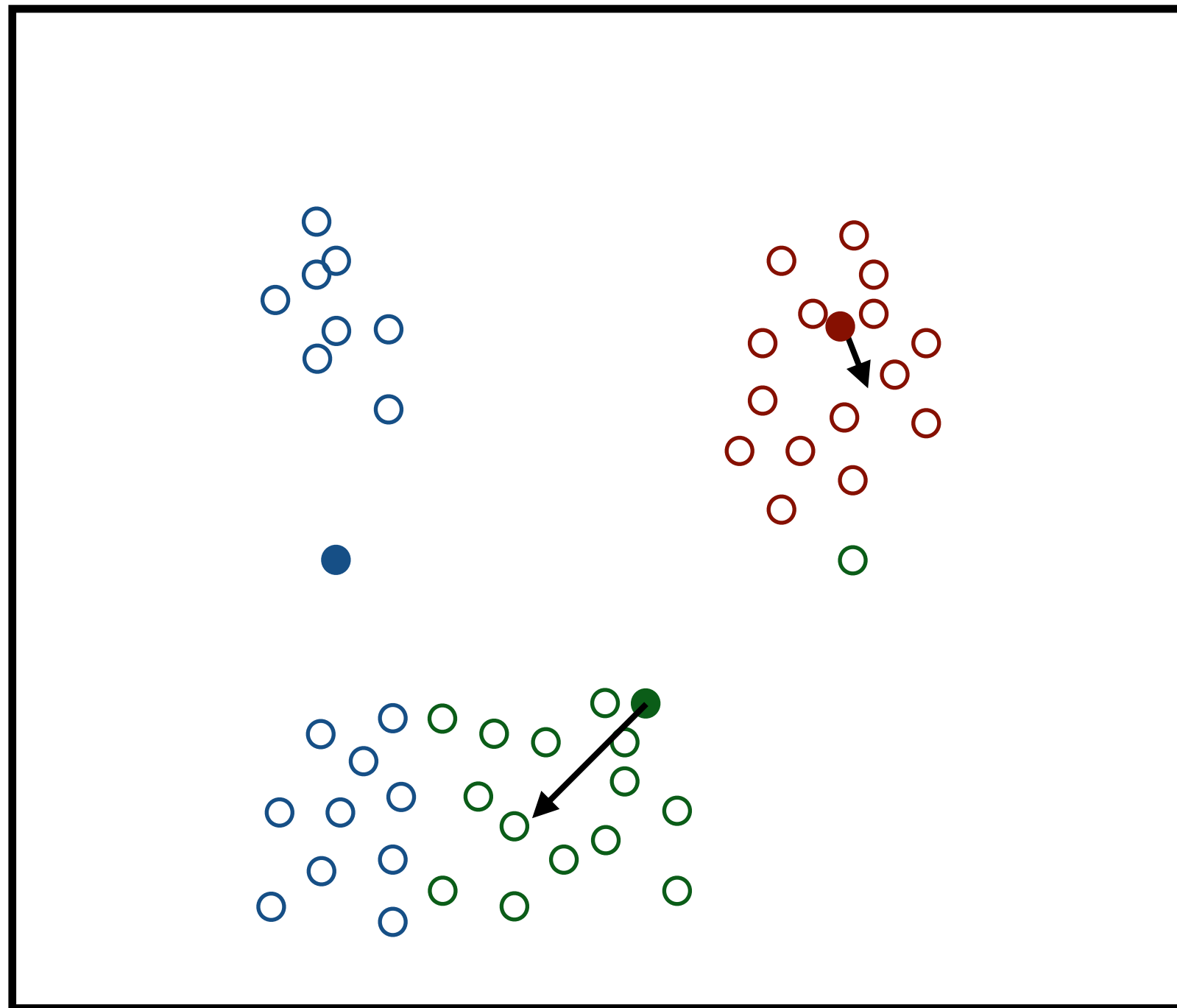
k-Means



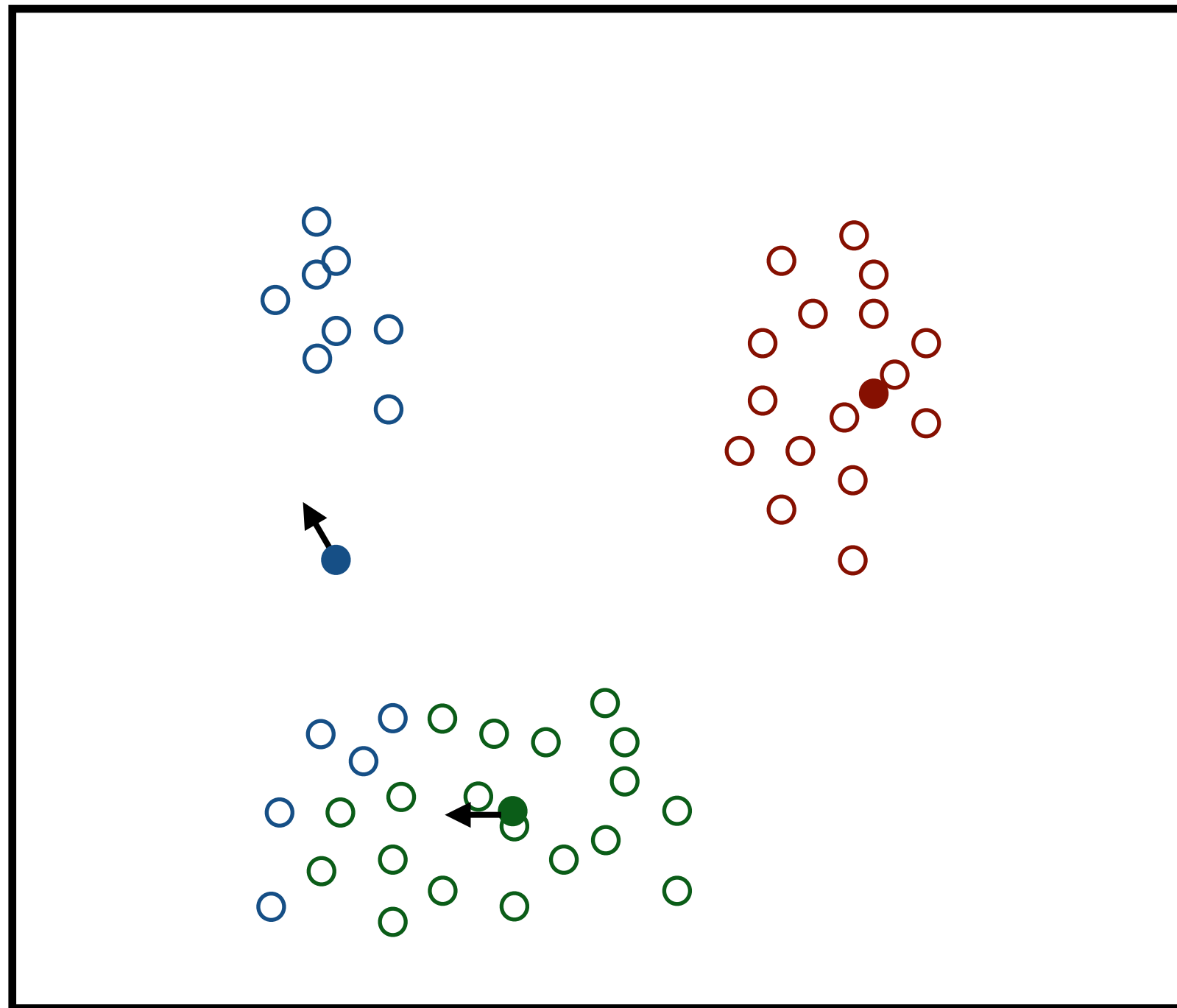
k-Means



k-Means



k-Means



k-Means

Remaining questions ...

How to choose k ?

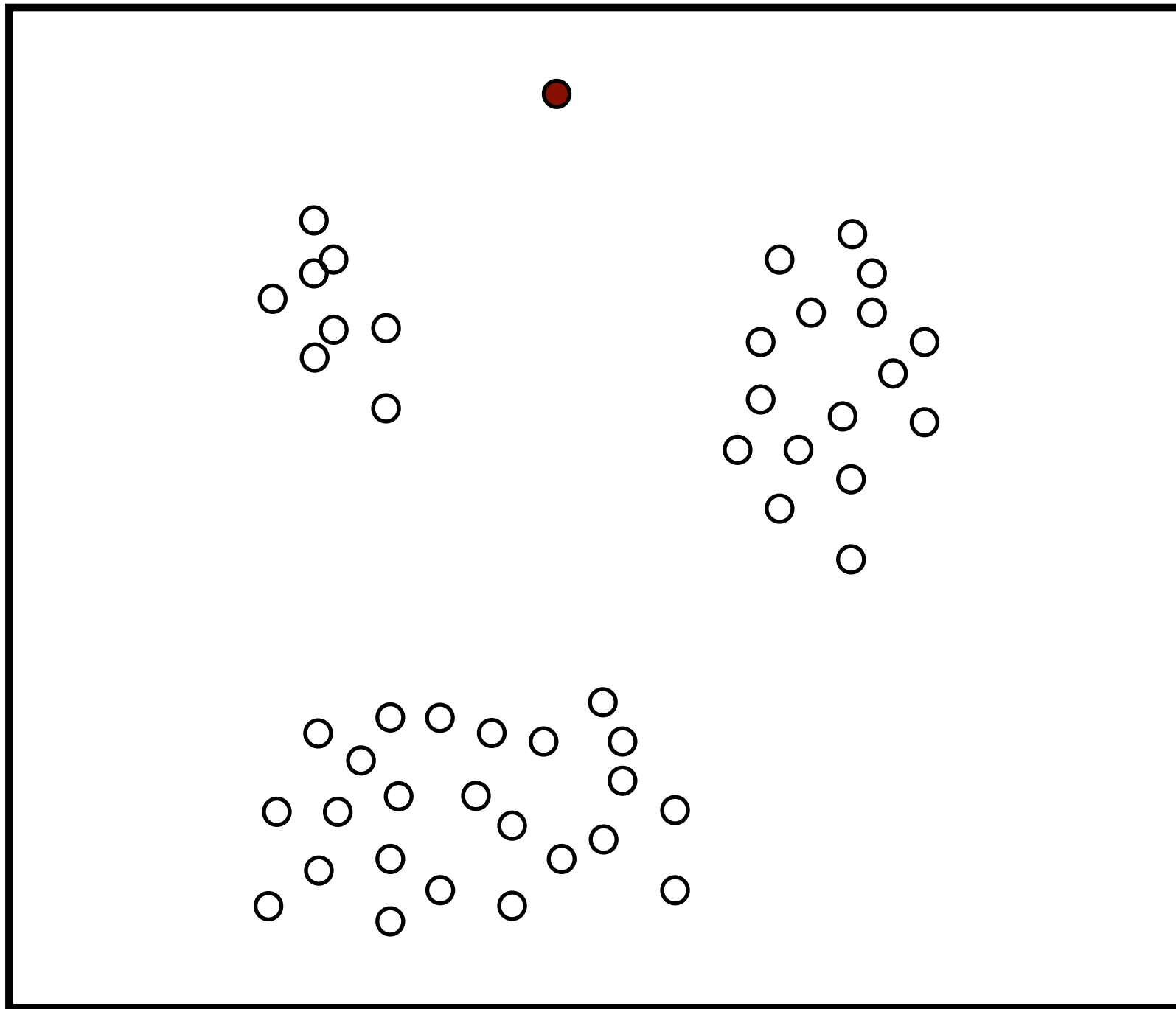
What about bad initializations?

Broadly:

- Use a quality metric.
- Search through k .
- Random restart initial position.

Density Estimation

Clustering: can answer *which cluster*, but not *does this belong?*



Density Estimation

Estimate the *distribution the data is drawn from*.

This allows us to evaluate the probability that a new point is drawn from the same distribution as the old data.

Formal definition

Given:

- Data points $X = \{x_1, \dots, x_n\}$,

Find:

- PDF $P(X)$

GMM

Simple approach:

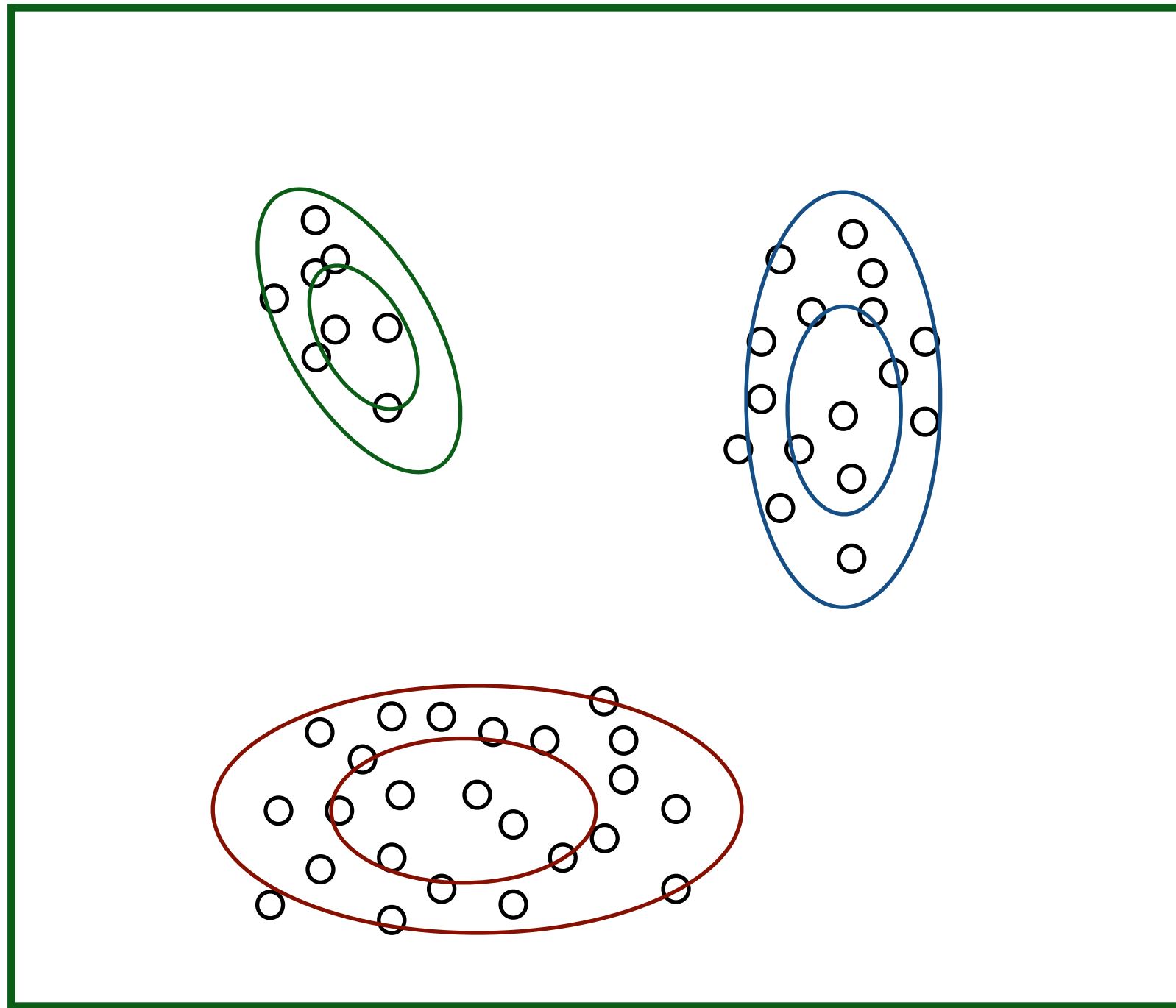
- Model the data as a mixture of Gaussians.

Each Gaussian has its own mean and variance.

Each has its own *weight* (sum to 1).

Weighted sum of Gaussians still a PDF.

GMM



GMM

Algorithm - broadly as before:

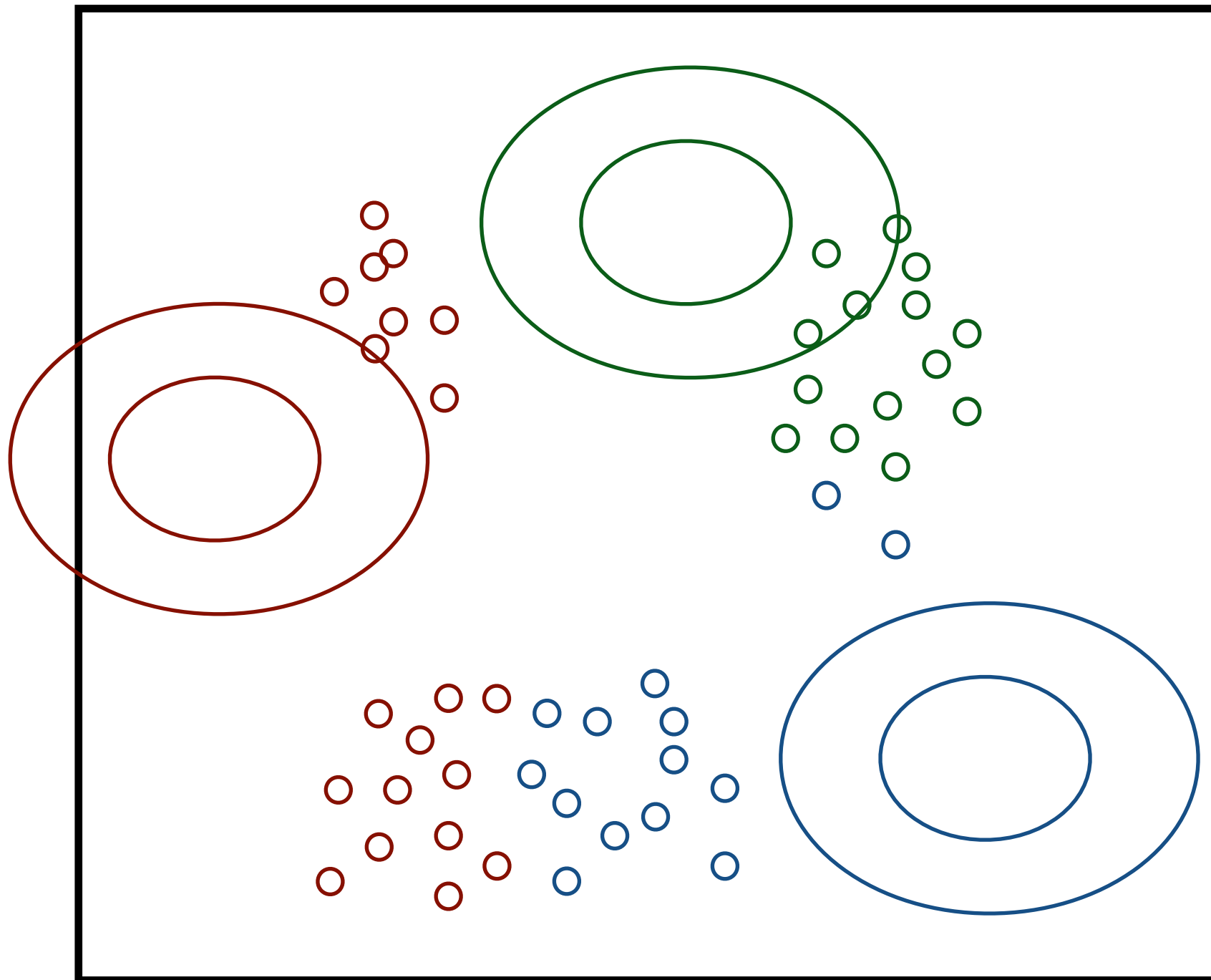
- Place k “means” $\{\mu_1, \dots, \mu_k\}$ at random.
- Set variances to be high.
- Assign all points to highest probability distribution.

$$C_i = \{x_v | N(x_v | \mu_i, \sigma_i^2) > N(x_v | \mu_j, \sigma_j^2), \forall j\}$$

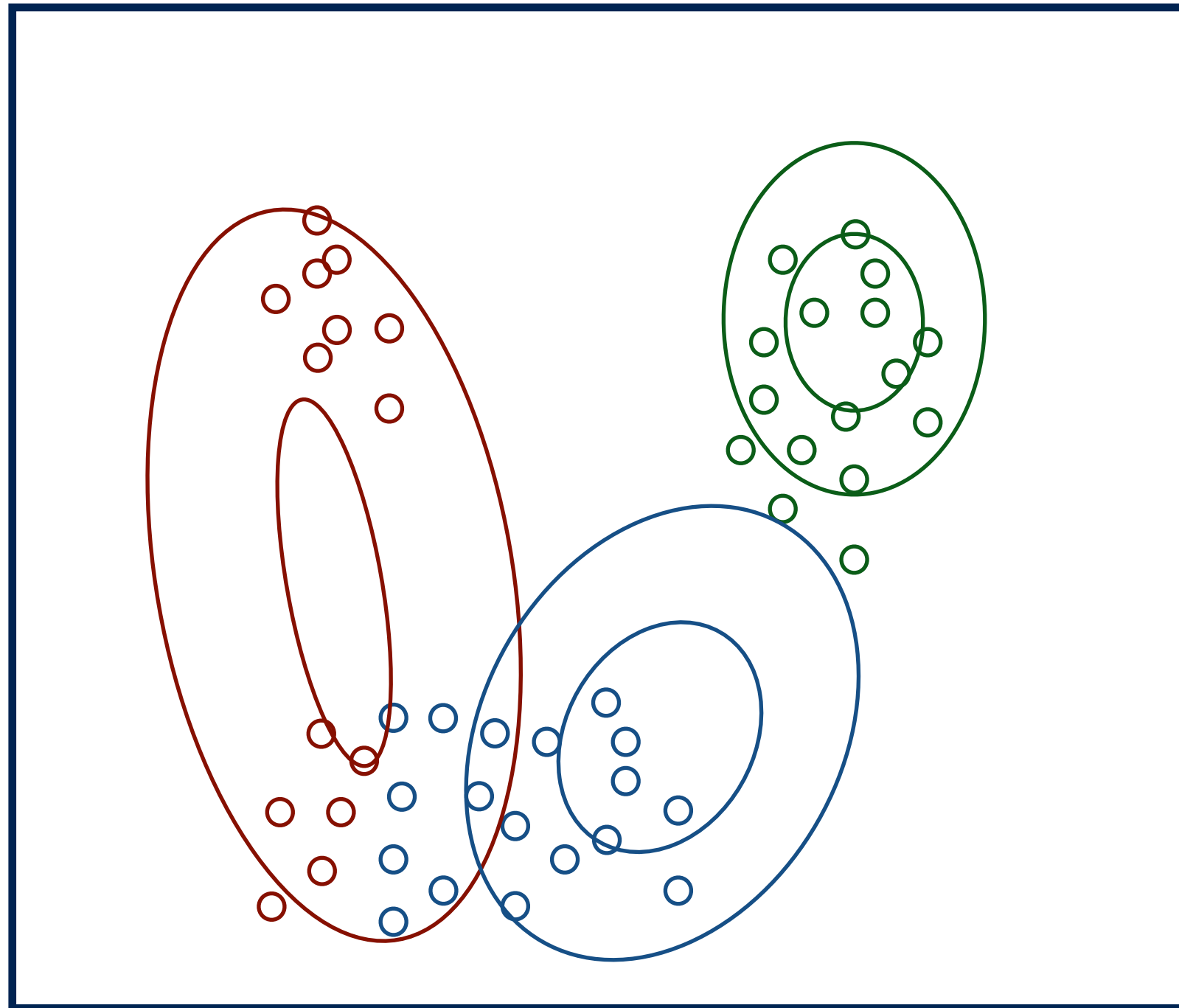
- Set mean, variance to match assigned data.

$$\mu_i = \sum_{v \in C_i} \frac{x_v}{|C_i|} \quad \sigma_i^2 = \text{variance}(C_i) \quad w_i = \frac{|C_i|}{\sum_j |C_j|}$$

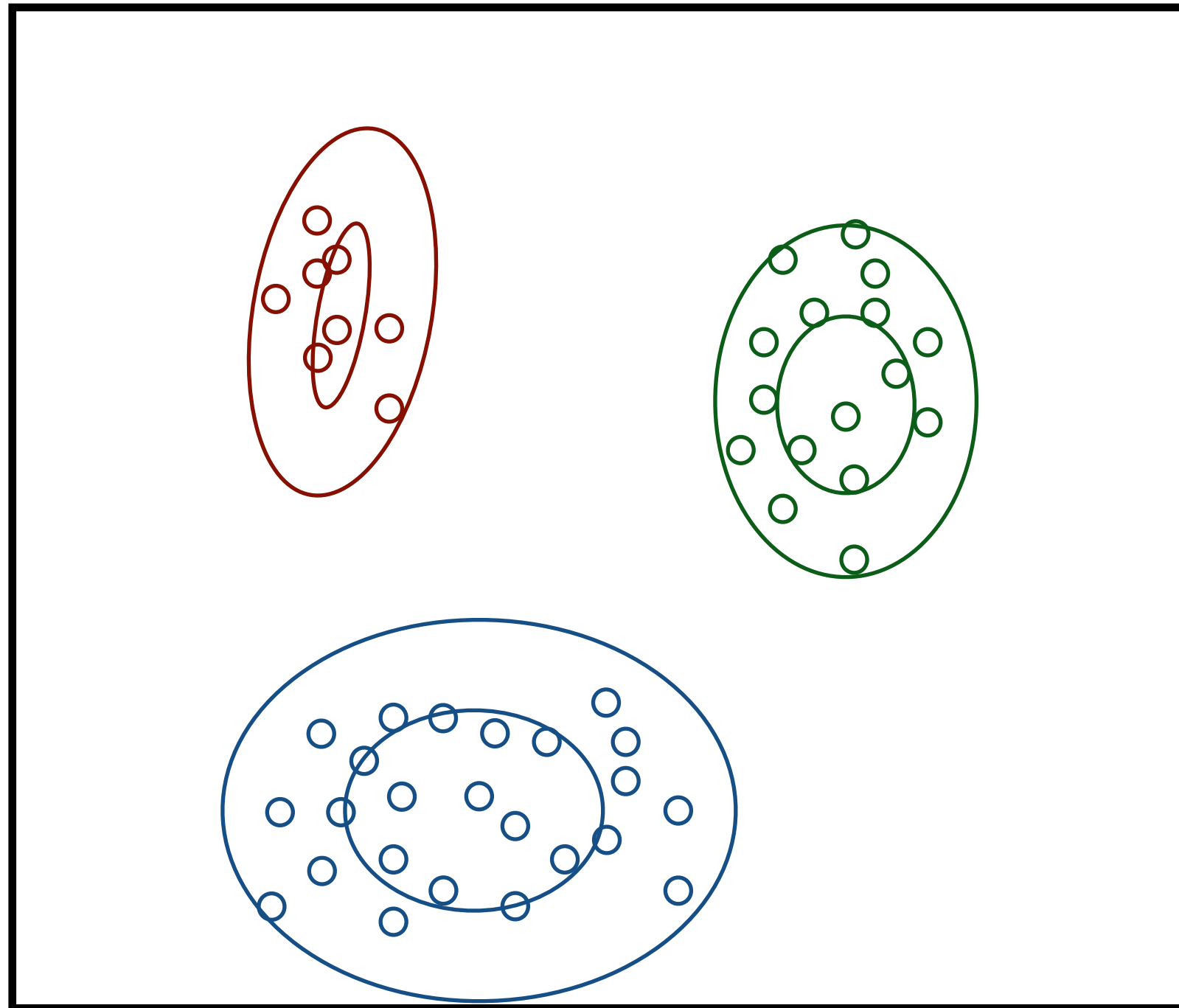
GMM



GMM



GMM



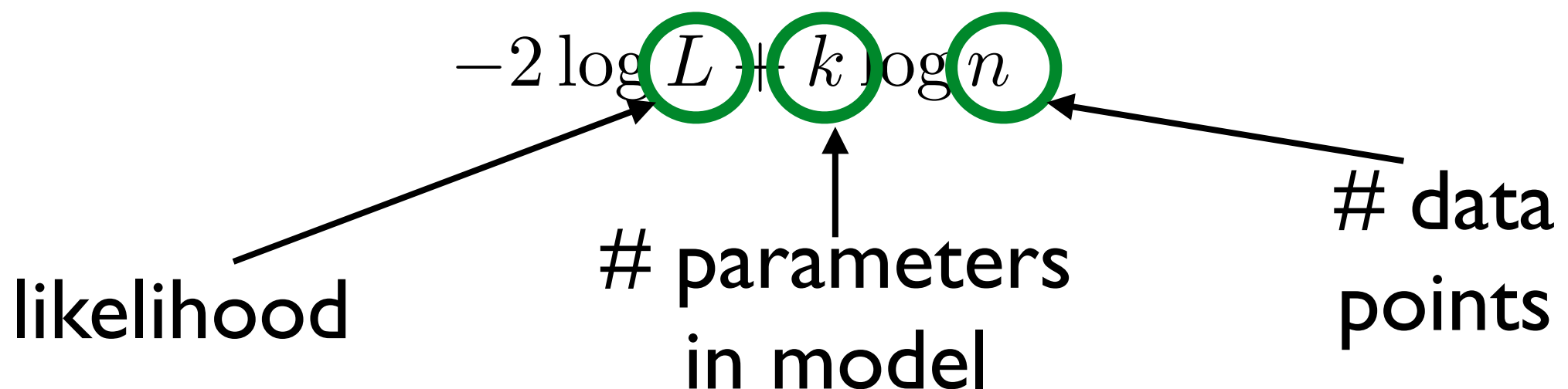
GMM

Major issue:

- How to decide between two GMMs?
- How to choose k ?

General statistical question: model selection.
Several good answers for this.

Simple example: **Bayesian information criterion (BIC)**.
Trades off model complexity (k) with fit (likelihood).

$$-2 \log L + k \log n$$


The diagram shows the BIC formula $-2 \log L + k \log n$ with three terms circled in green. Arrows point from descriptive text to each term: 'likelihood' points to L , '# parameters in model' points to k , and '# data points' points to n .

likelihood

parameters in model

data points

Dimensionality Reduction

$$X = \{x_1, \dots, x_n\}$$

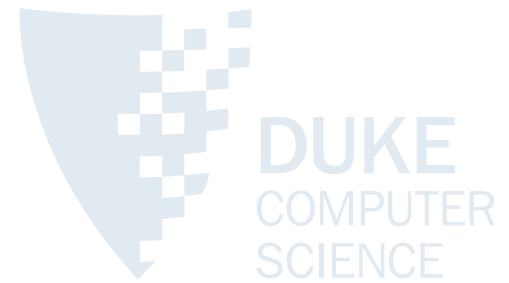
If n is high, data can be hard to deal with.

- High-dimensional decision boundary.
- Need more data.
- But data is often not really high-dimensional.

Dimensionality reduction:

- Reduce or compress the data
- Try not to lose too much!
- Find intrinsic dimensionality

Dimensionality Reduction



For example, imagine if x_1 and x_2 are meaningful features, and $x_3 \dots x_n$ are random noise.

What happens to k-nearest neighbors?

What happens to a decision tree?

What happens to the perceptron algorithm?

What happens if you want to do clustering?

Dimensionality Reduction

Often can be phrased as a projection:

$$f : X \rightarrow X'$$

where:

- $|X'| \ll |X|$
- our goal: retain as much *variance* as possible.

Variance captures what *varies within the data*.

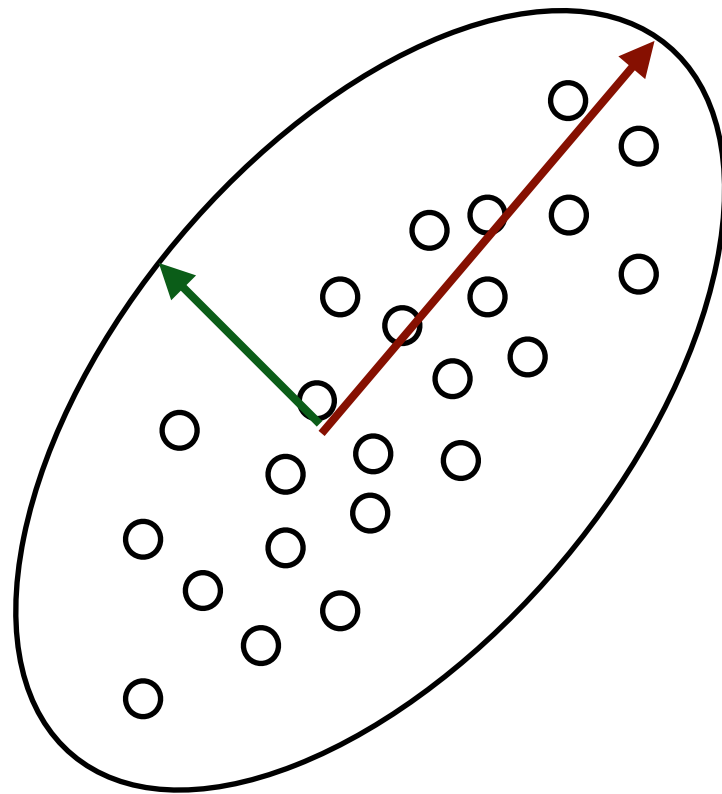
PCA

Principle Components Analysis.

Project data into a new space:

- Dimensions are linearly uncorrelated.
- We have a measure of importance for each dimension.

PCA



PCA

- Gather data X_1, \dots, X_m .
- Adjust data to be zero-mean:

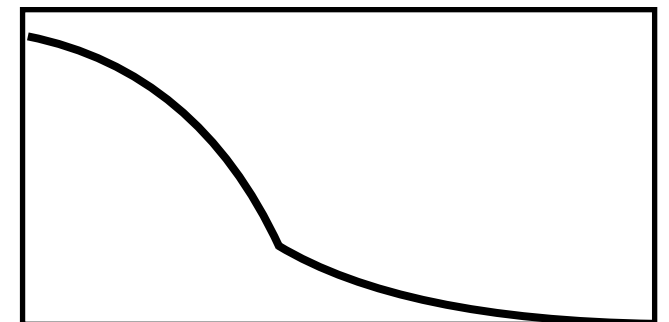
$$\bar{X}_i = X_i - \sum_j \frac{X_j}{m}$$

- Compute covariance matrix C .
- Compute unit eigenvectors V_i and eigenvalues v_i of C .

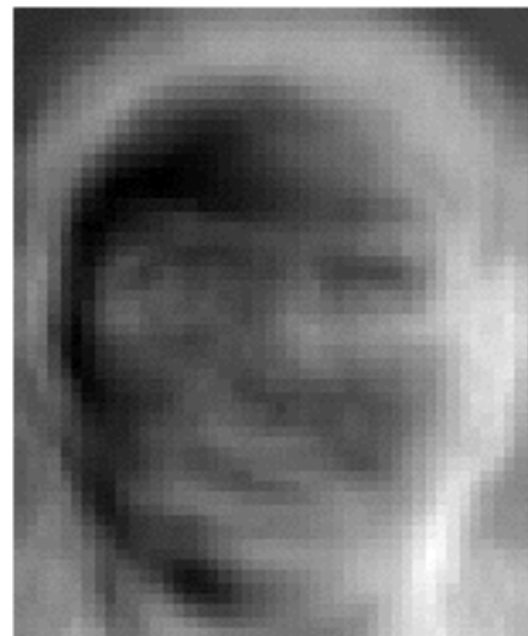
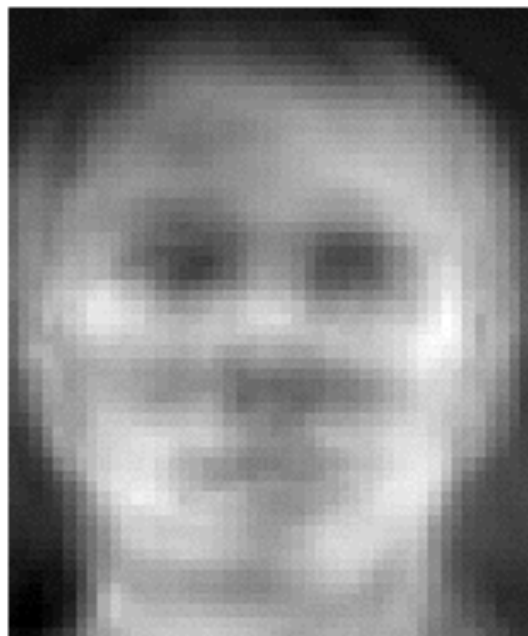
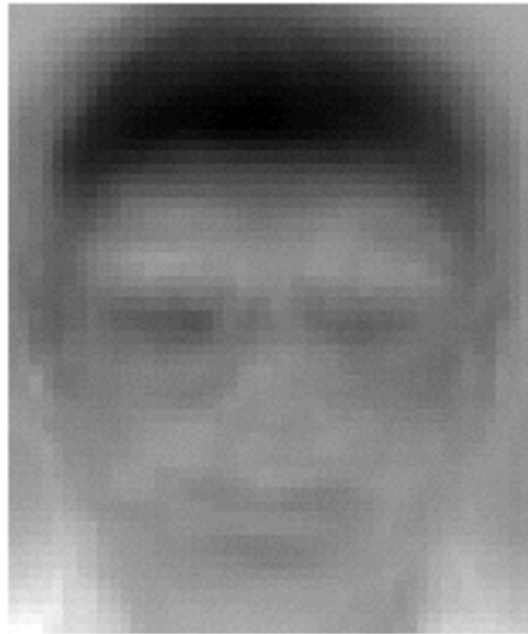
Each V_i is a direction, and each v_i is its importance - the amount of the data's variance it accounts for.

New data points:

$$\hat{X}_i = [V_1, \dots, V_p] X_i$$



Eigenfaces

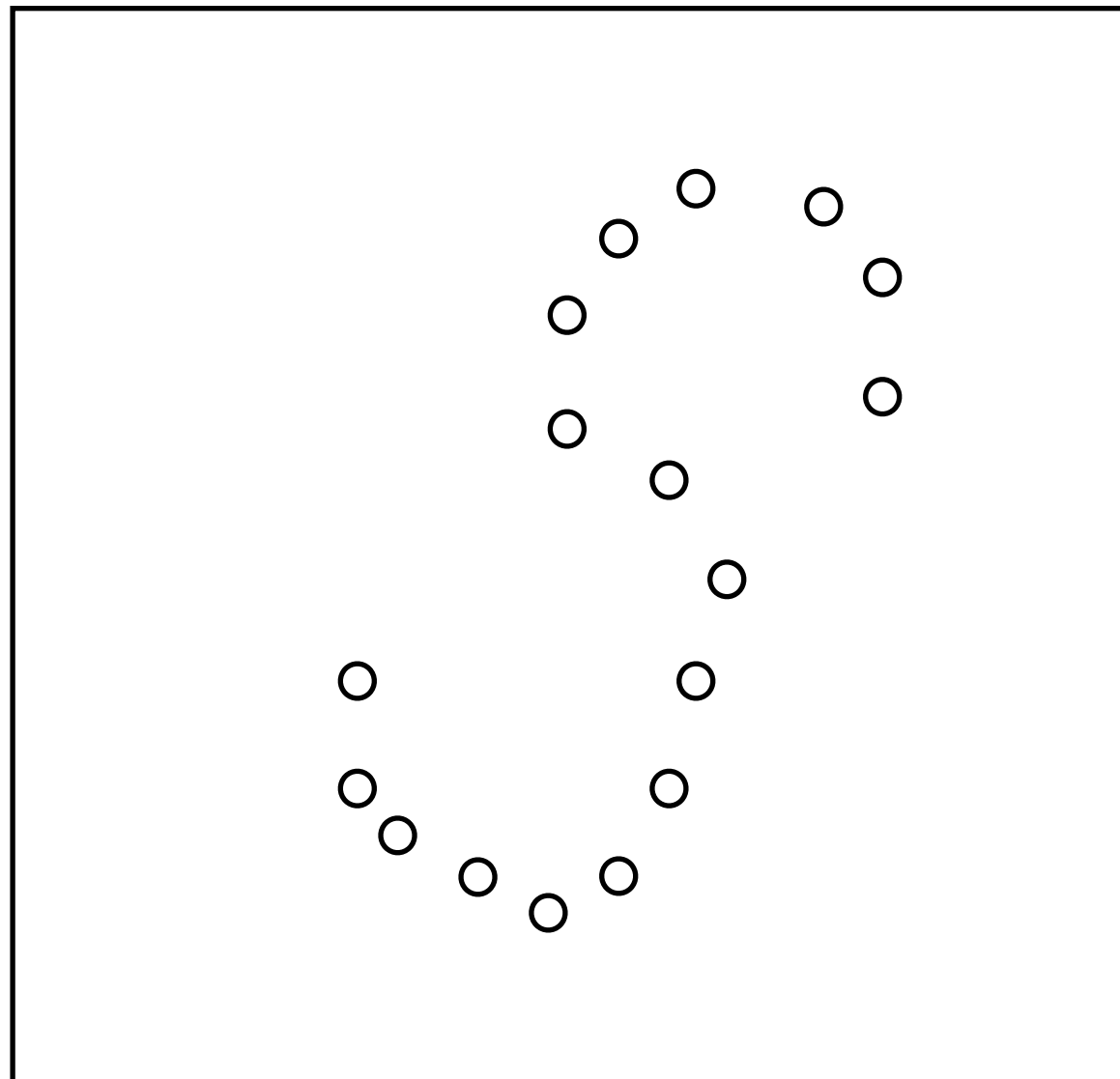


(courtesy ORL database)

ISOMAP

Another approach:

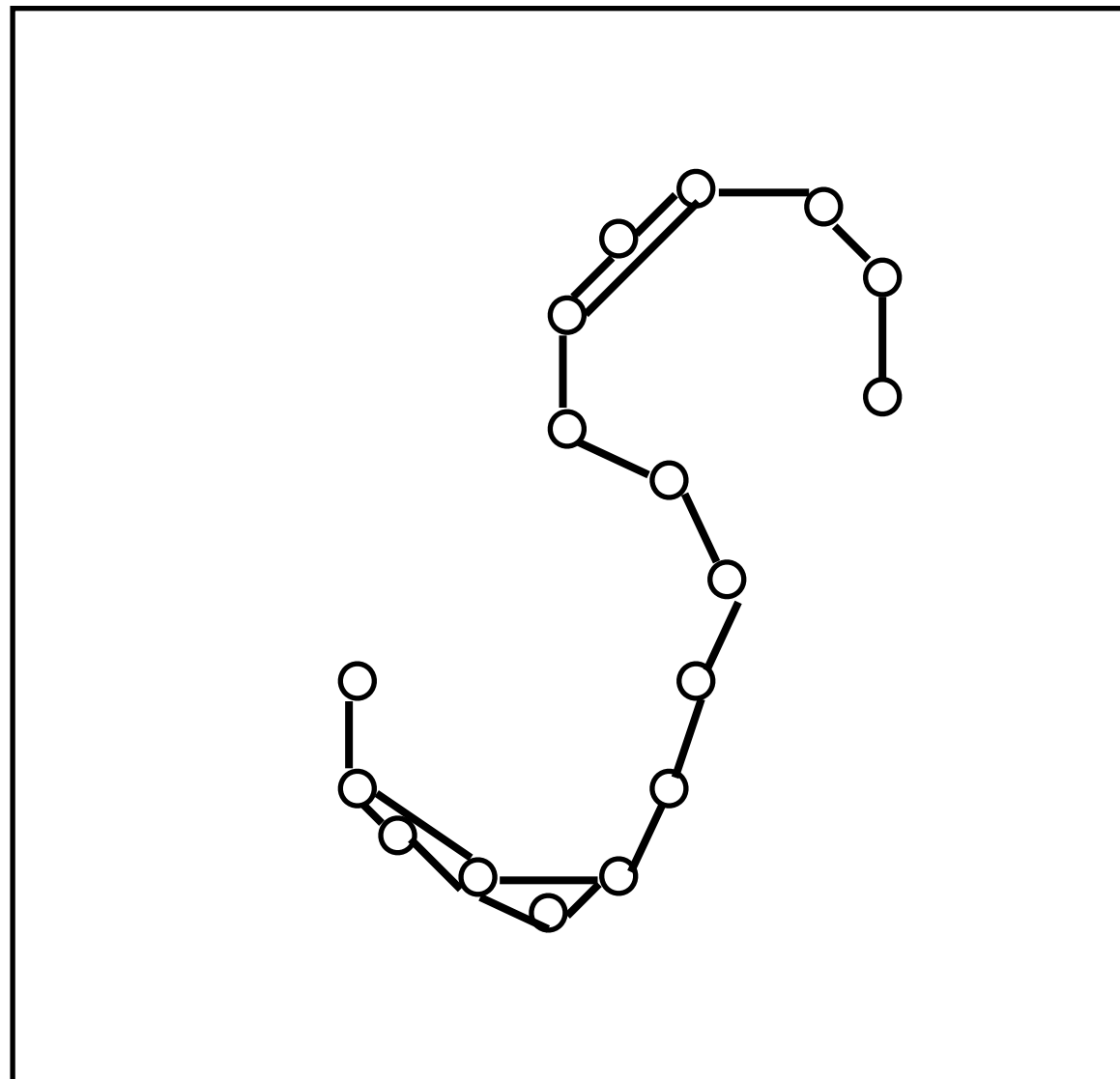
- Estimate intrinsic geometric dimensionality of data.
- Recover natural distance metric



ISOMAP

Core idea: distance metric *locally Euclidean*

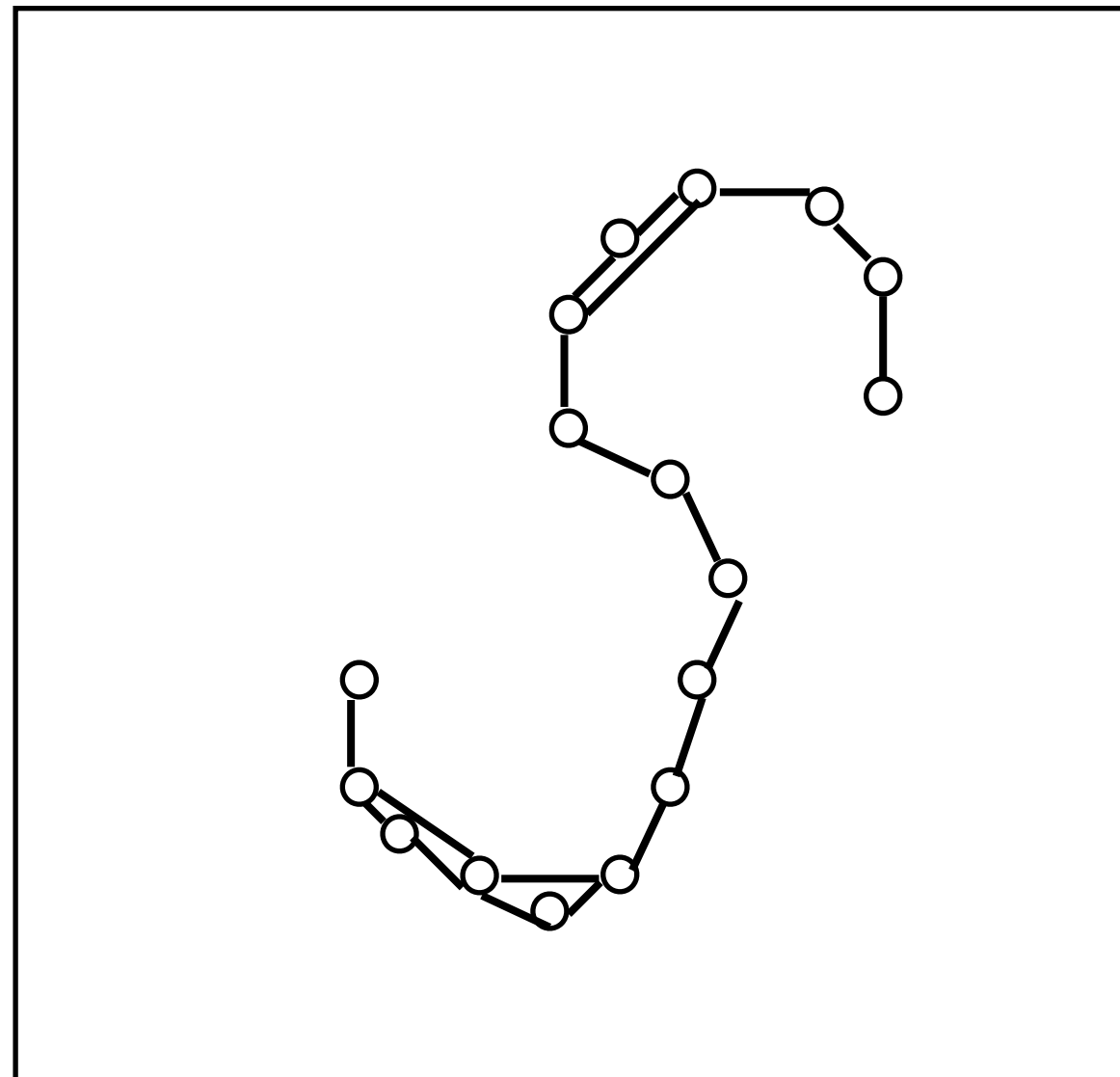
- Small radius r , connect each point to neighbors
- Weight based on Euclidean distance



ISOMAP

Solve all-points shortest pairs:

- Transforms local distance to global distance.
- Compute embedding.



ISOMAP

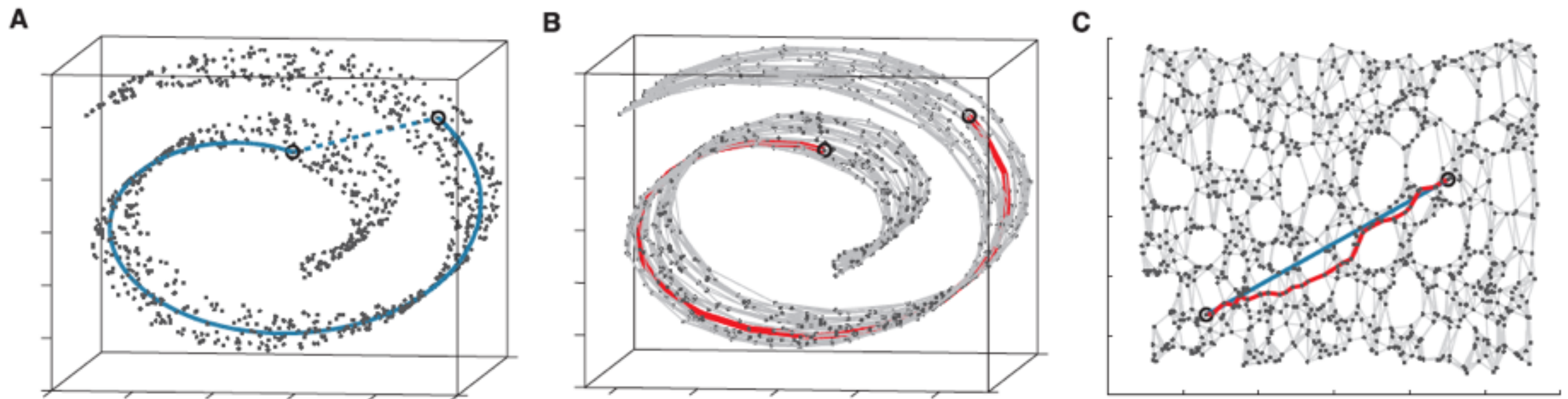


Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (A) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (B) The neighborhood graph G constructed in step one of Isomap (with $K = 7$ and $N =$

1000 data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in G . (C) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).

From Tenenbaum, de Silva, and Langford, *Science* 290:2319-2323, December 2000.

Application: Novelty Detection

Intrusion detection - when is a user behaving *unusually*?

First proposed by Prof. Dorothy Denning in 1986.
(1995 ACM Fellow)

