PHYS 139/239: Machine Learning in Physics

Lecture 13: Unsupervised learning, clustering, & autoencoders

Javier Duarte — February 21, 2023

Types of learning

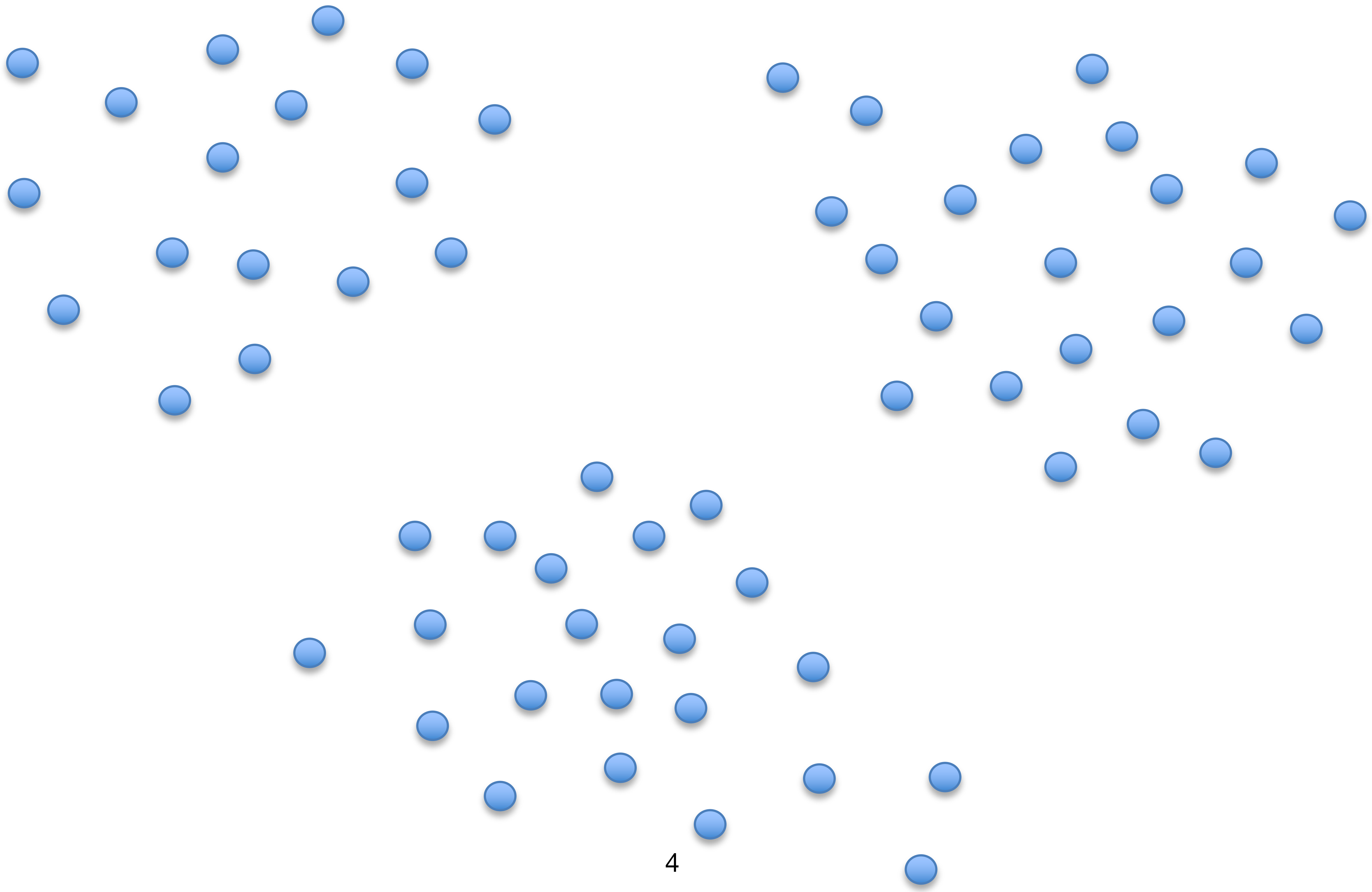
- Supervised learning: labels known for each data sample
- Unsupervised learning: only features known; no labels!
- Weakly supervised learning: features paired with noisy labels
- Semi-supervised learning: features paired with partial (incomplete) labels
- …

Unsupervised learning

- Clustering
- Dimensionality reduction

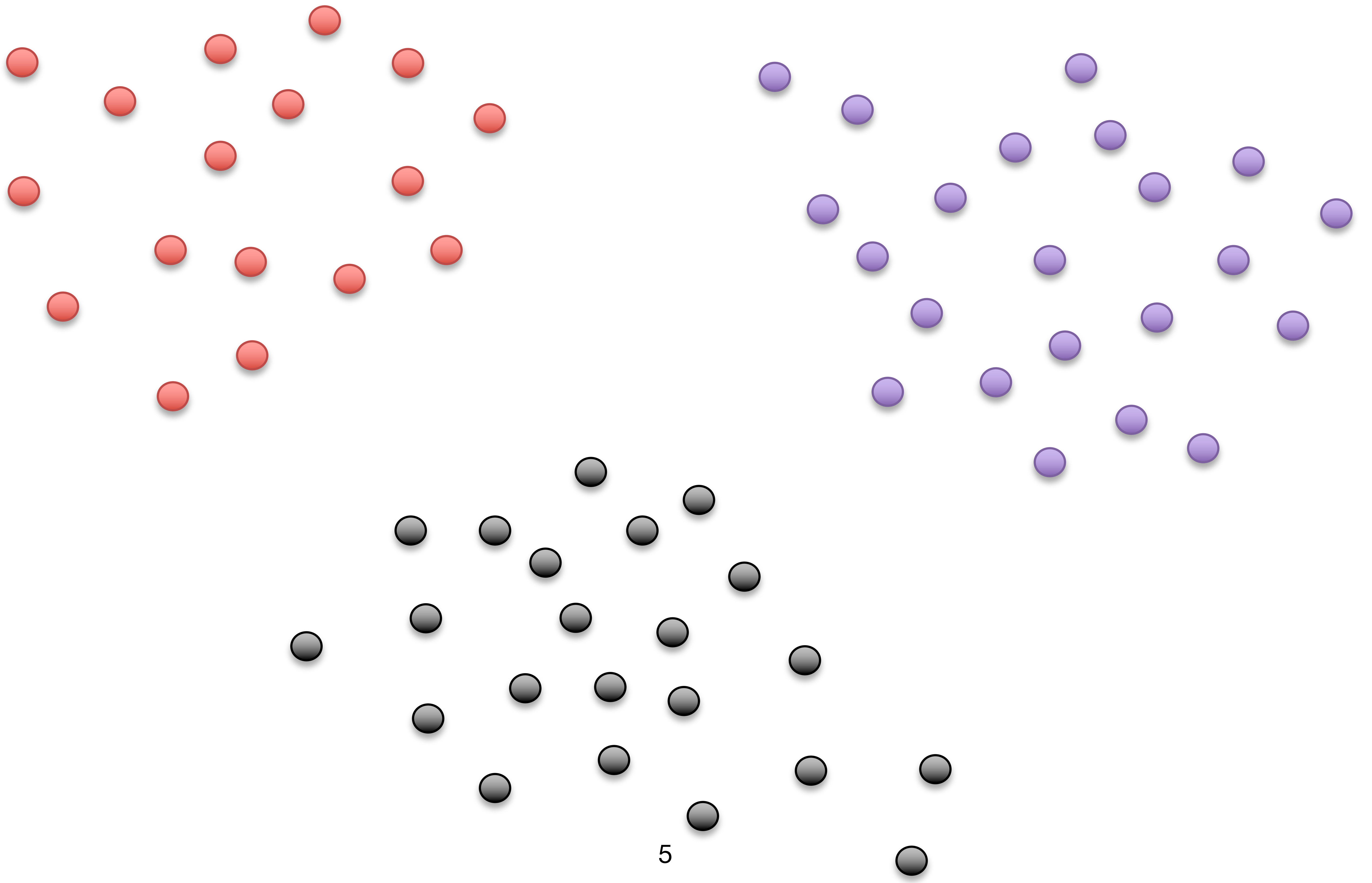
What is clustering?

- Clustering is the process of grouping data points into “clusters”
- High intra-cluster similarity
- Low inter-cluster similarity



What is clustering?

- Clustering is the process of grouping data points into “clusters”
- High intra-cluster similarity
- Low inter-cluster similarity



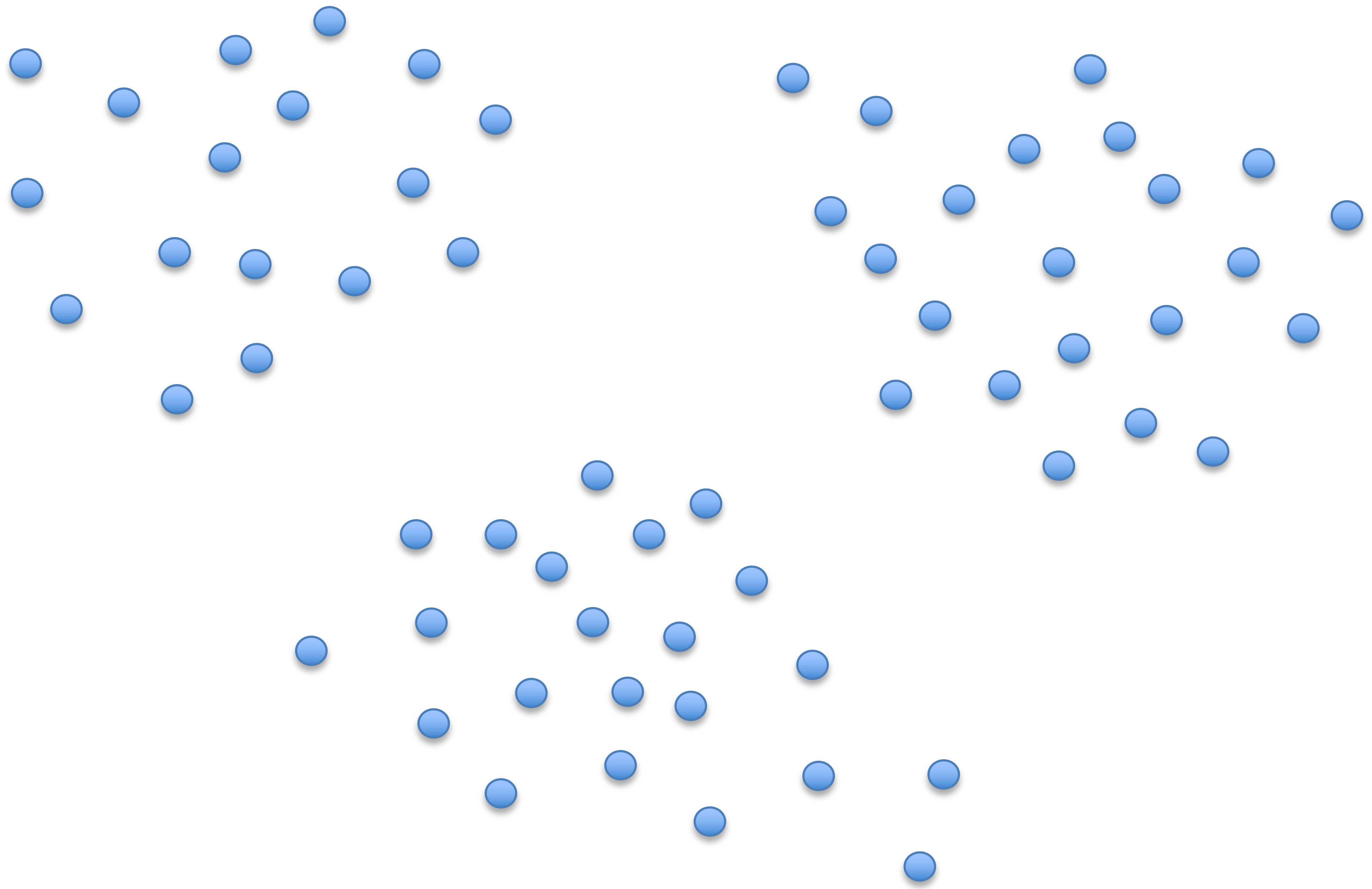
Unsupervised learning

- Given: unlabeled data $S = \{x_i\}_{i=1}^{N}$
  - Only input features
  - No labels
- Goal: find hidden structure/patterns
  - e.g. hidden structure is a clustering of data
  - Generative model of data $P(x)$
    - Discussed further in guest lecture
  - Low dimensional summary of the data

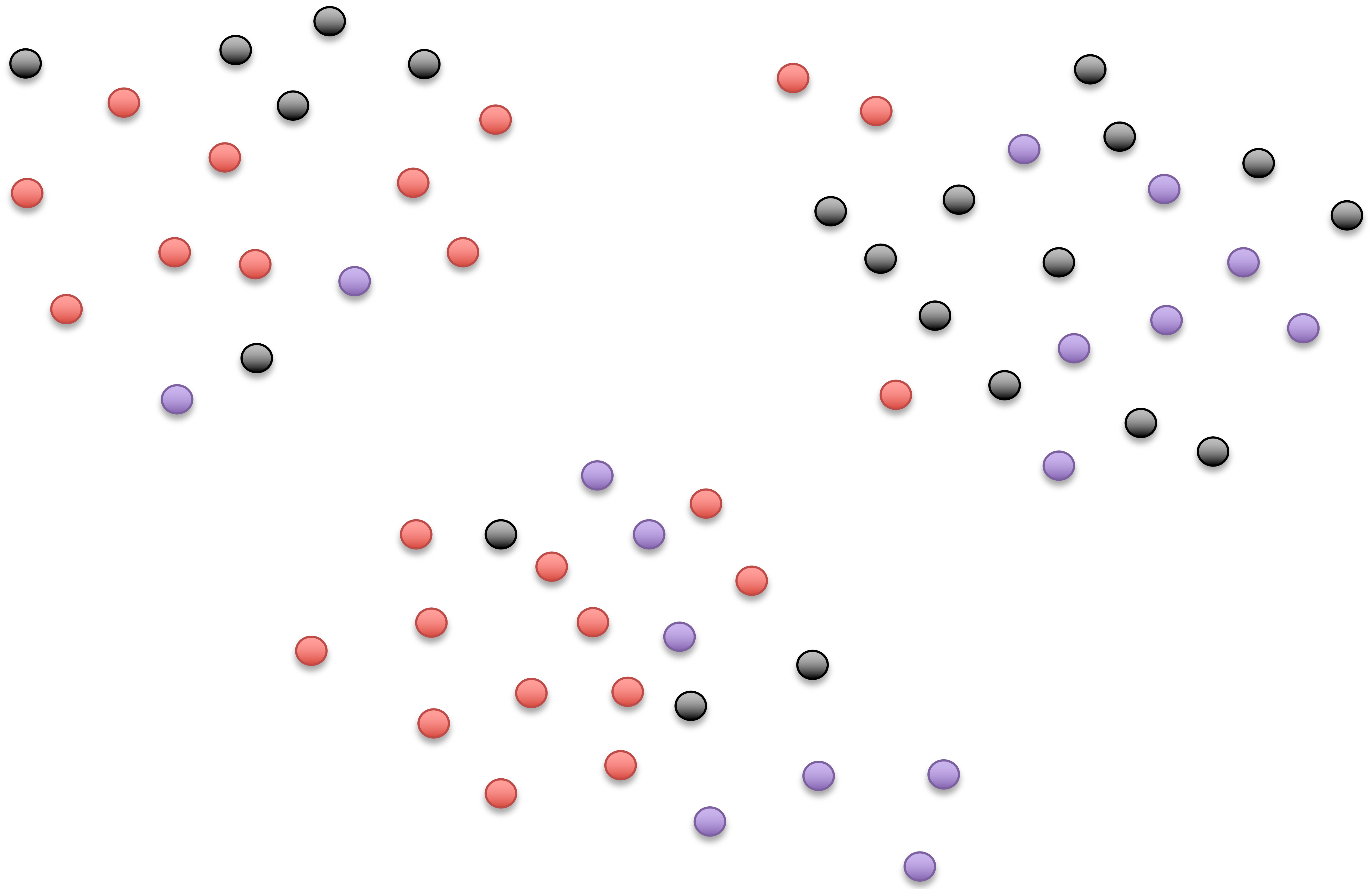
Why is clustering useful?

- Clustering is a “summary” of the data
  - Can just inspect cluster centers
  - Or inspect a few data points per cluster
- Compact preprocessing of data before supervised training

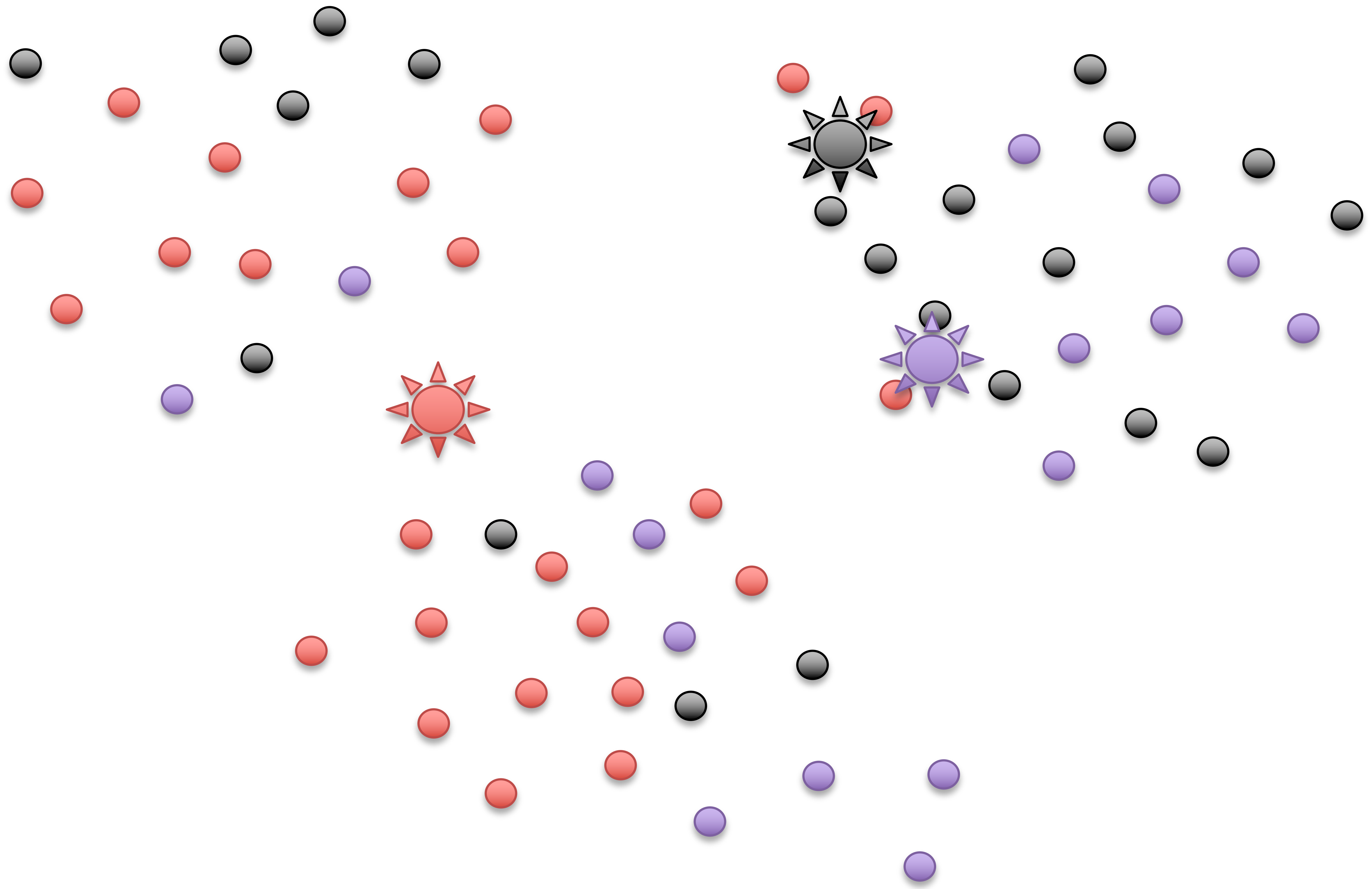
Centroid-based (K-means) clustering



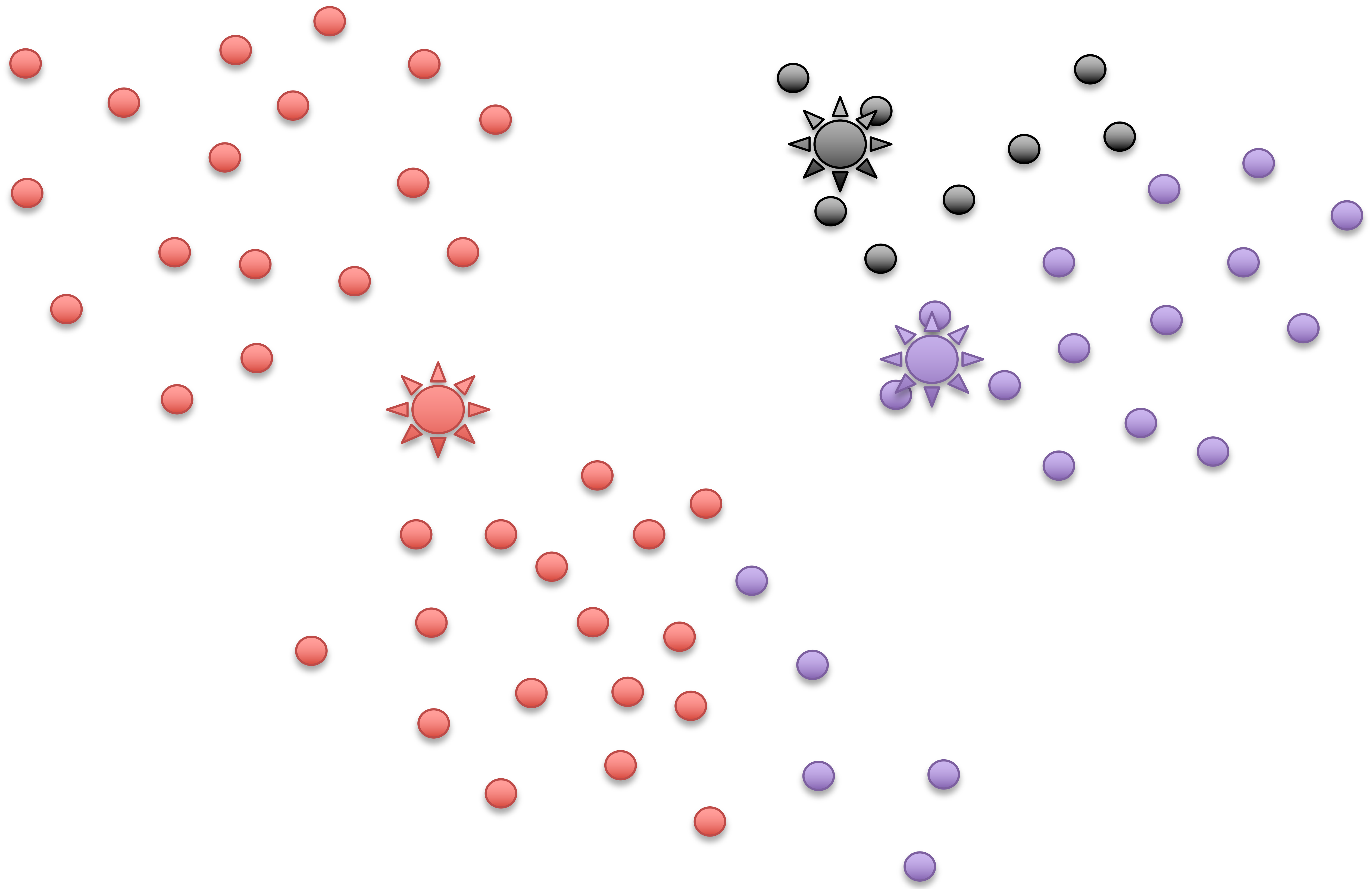
Centroid-based (K-means) clustering



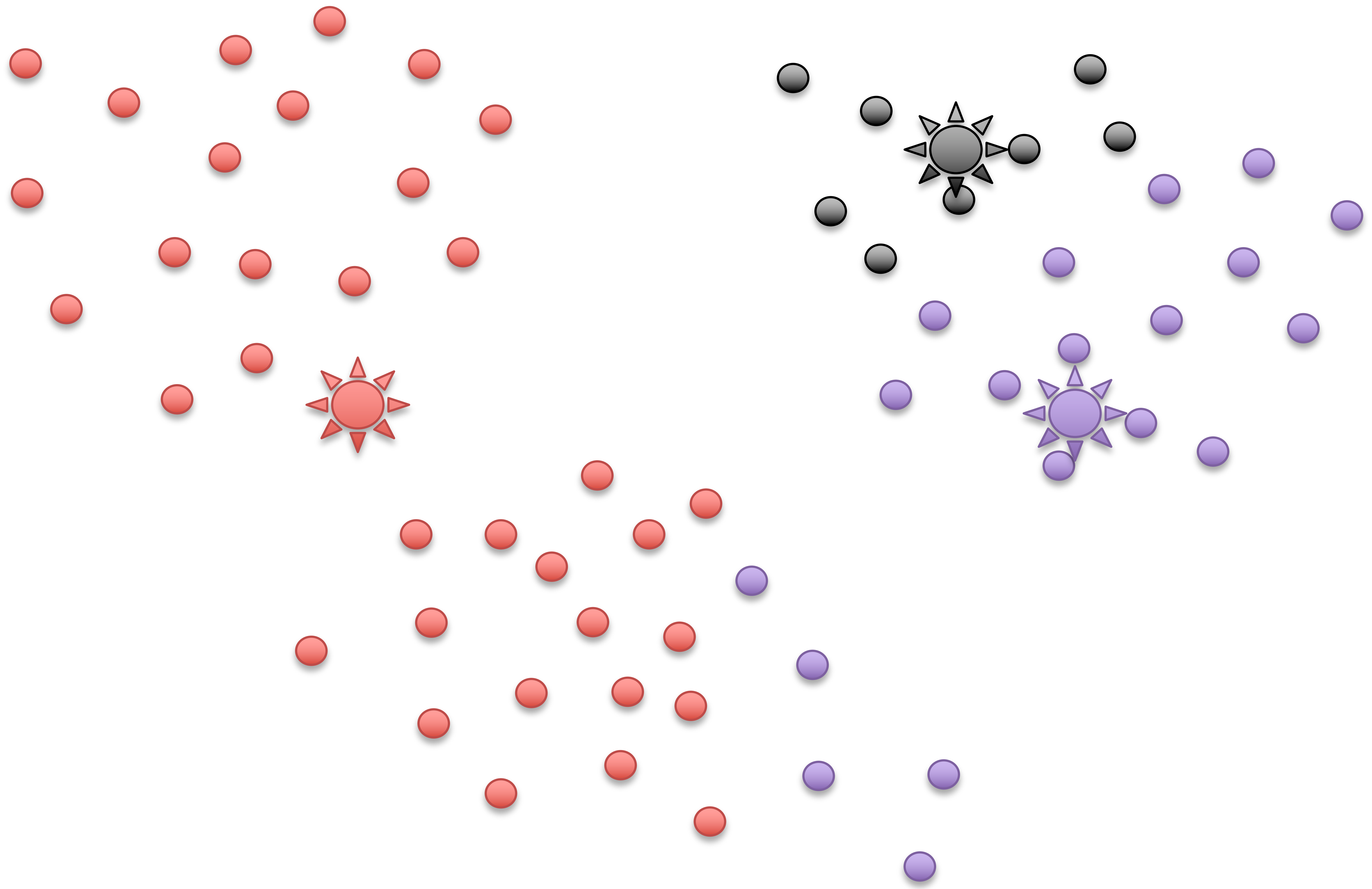
Centroid-based (K-means) clustering



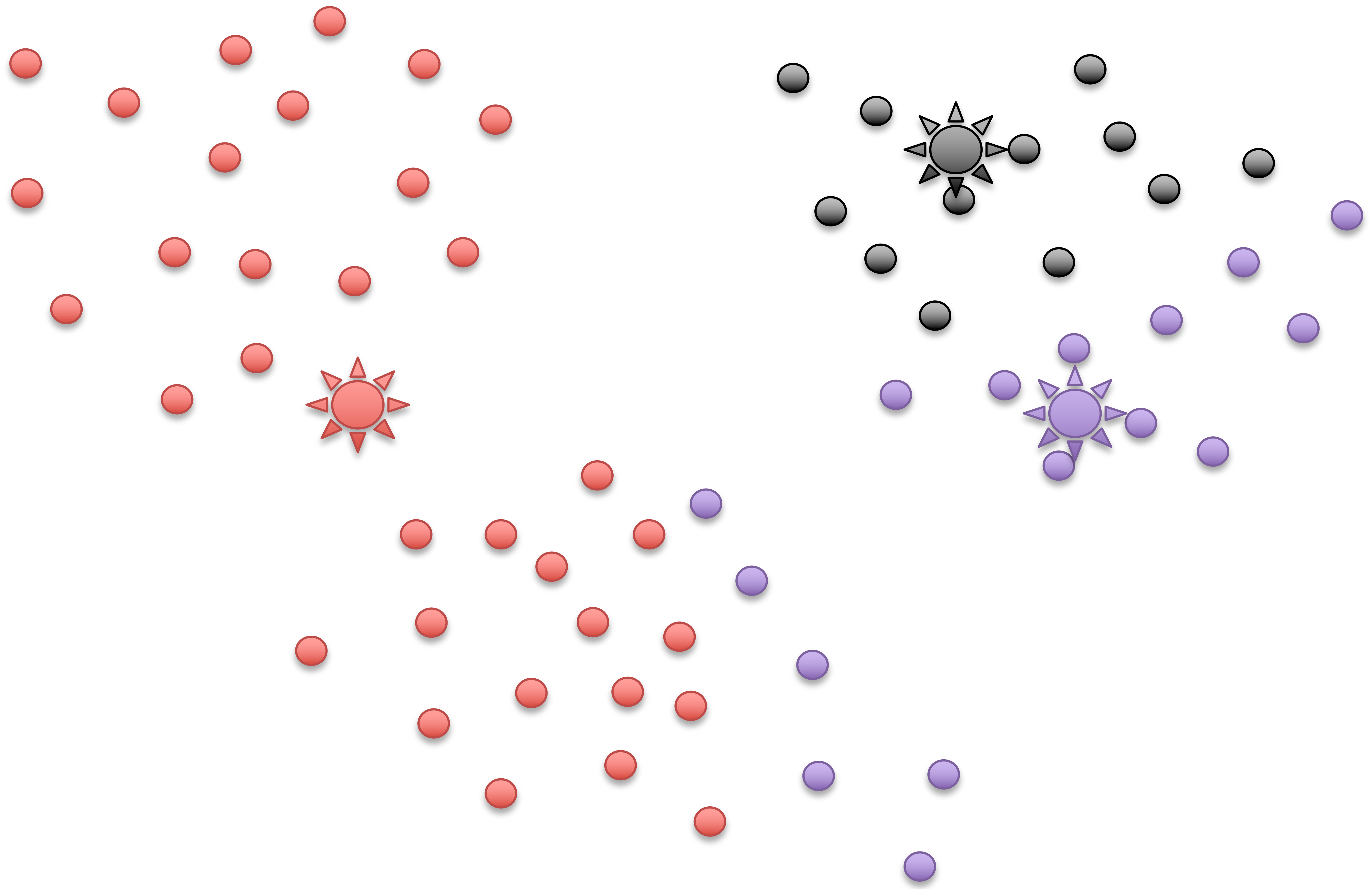
Centroid-based (K-means) clustering



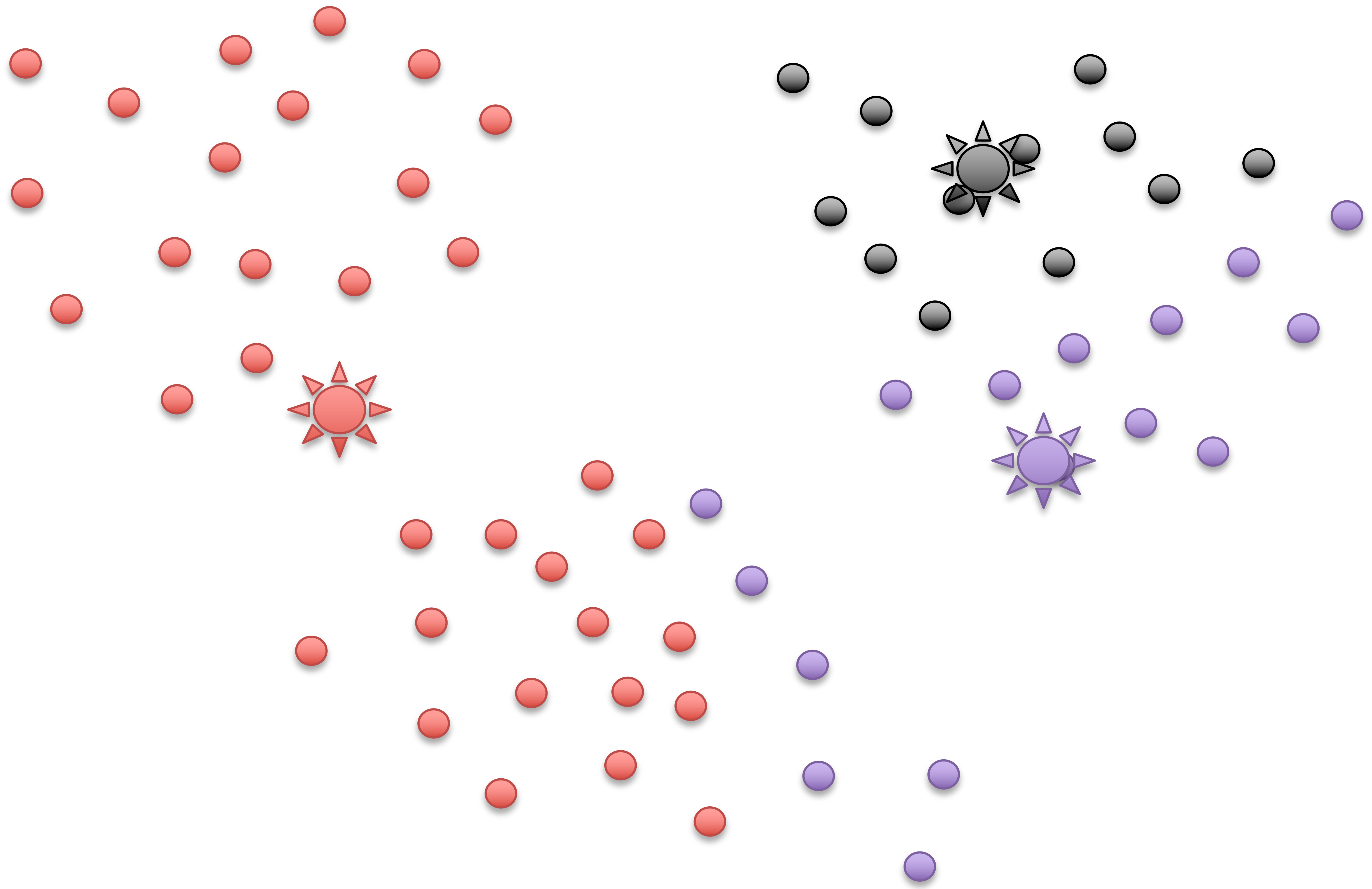
Centroid-based (K-means) clustering



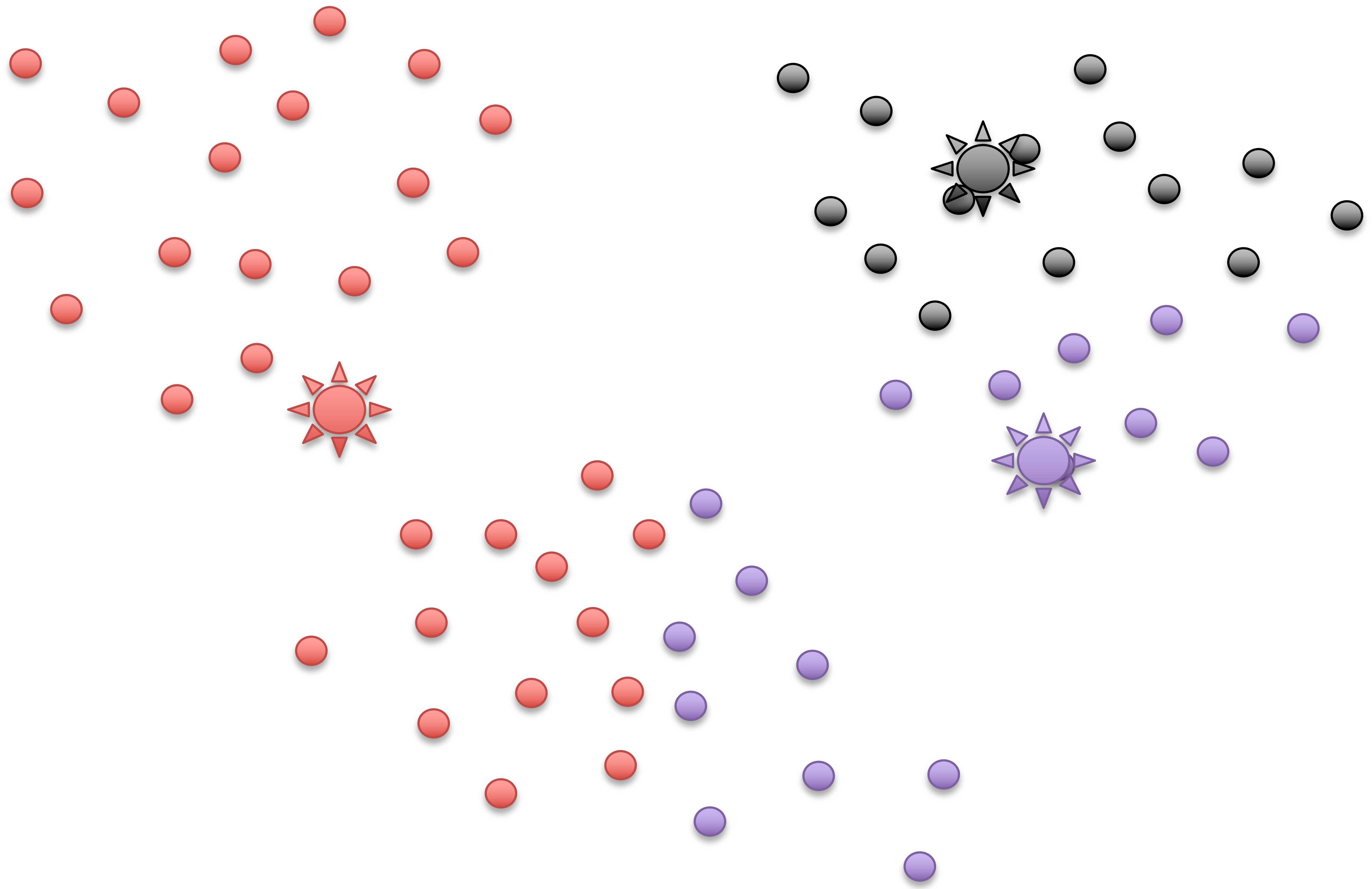
Centroid-based (K-means) clustering



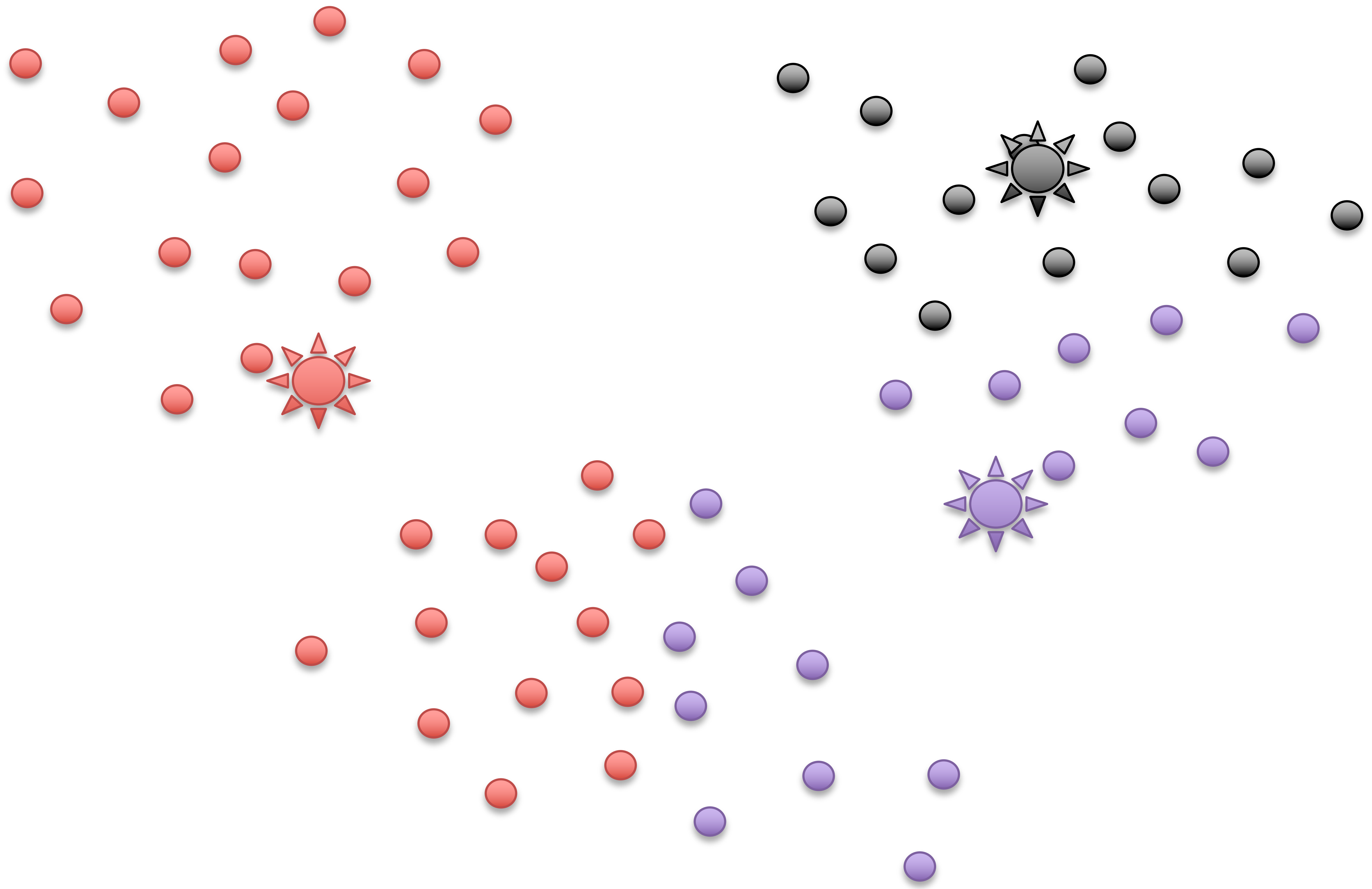
Centroid-based (K-means) clustering



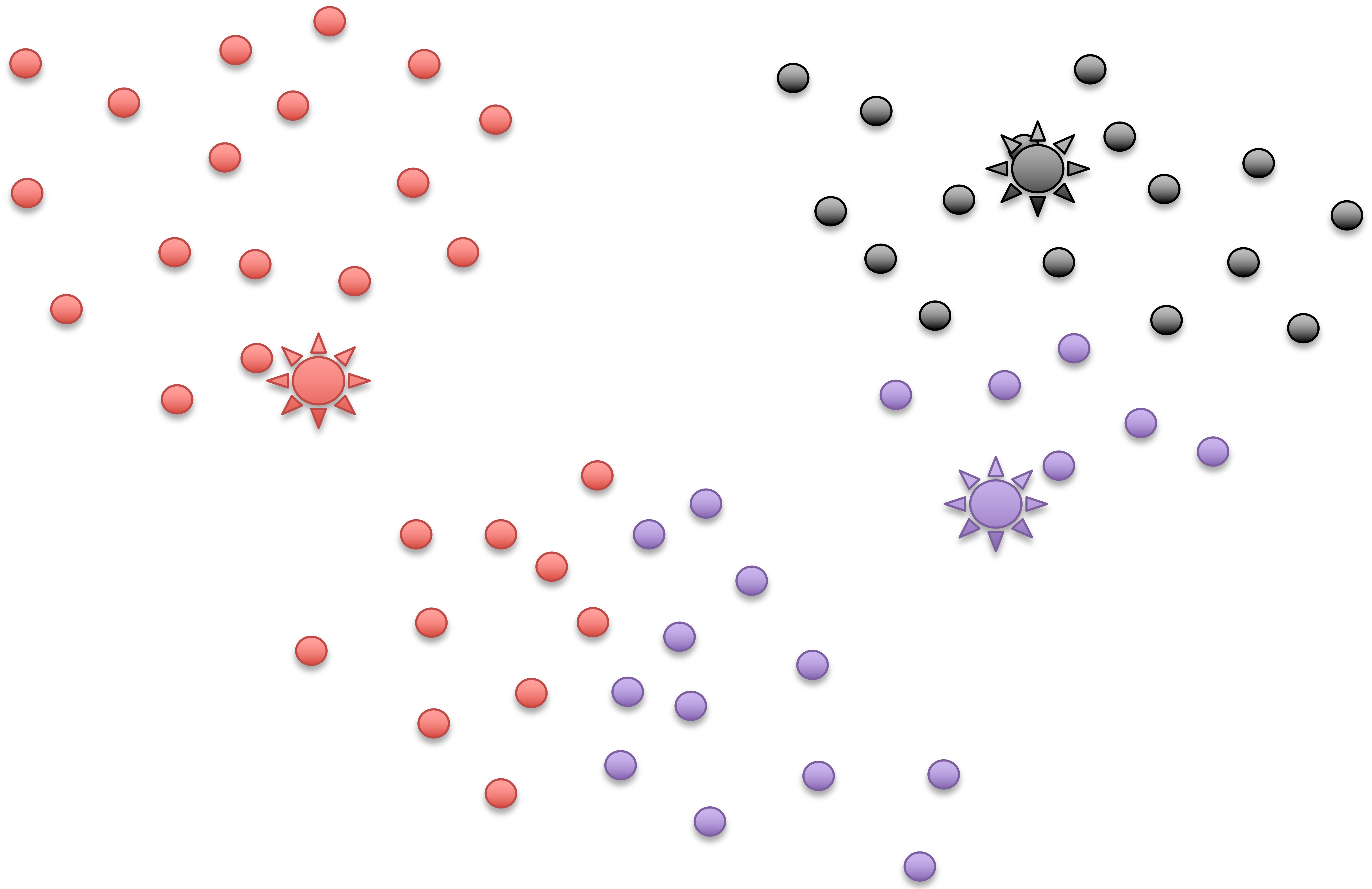
Centroid-based (K-means) clustering



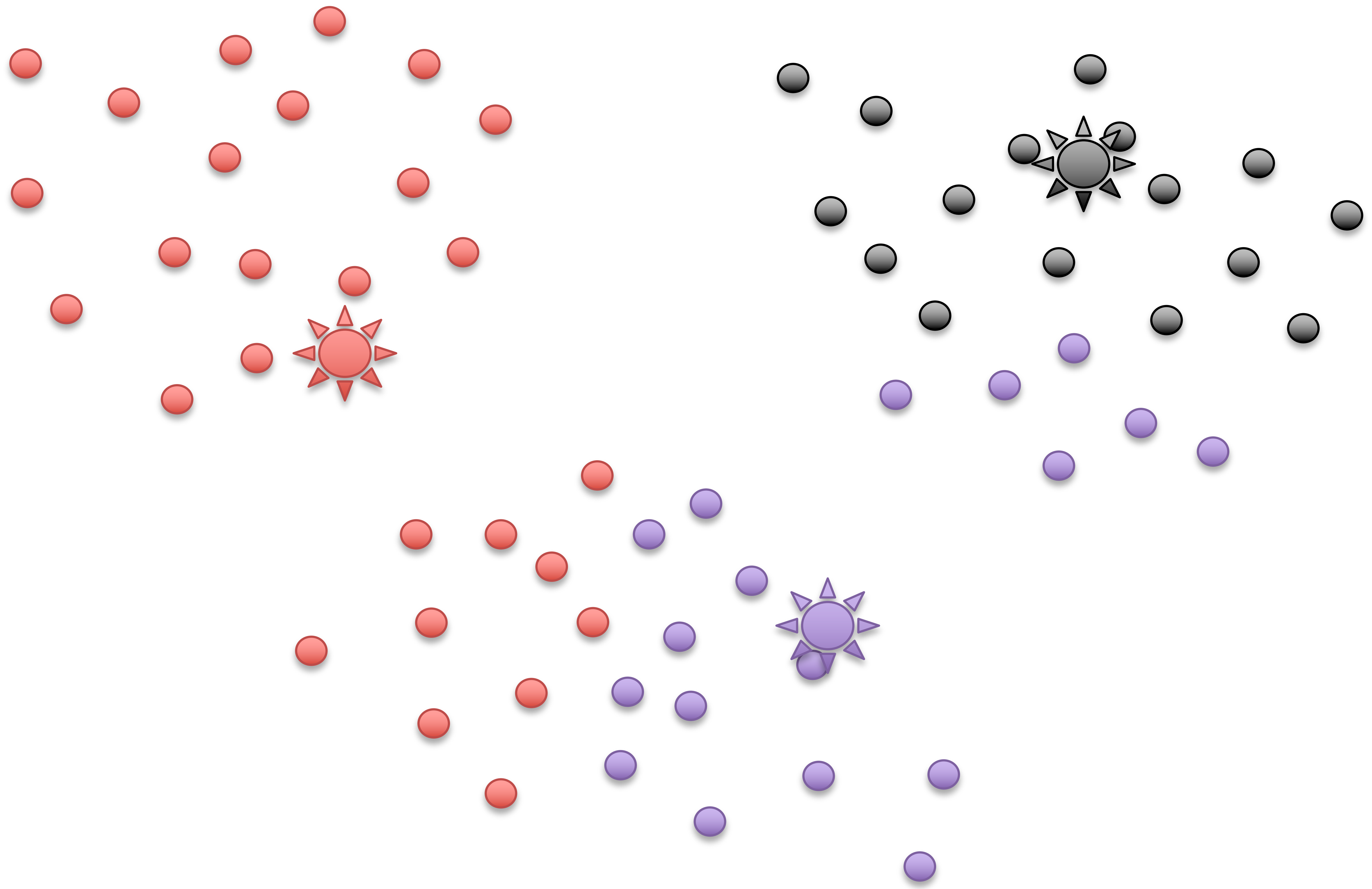
Centroid-based (K-means) clustering



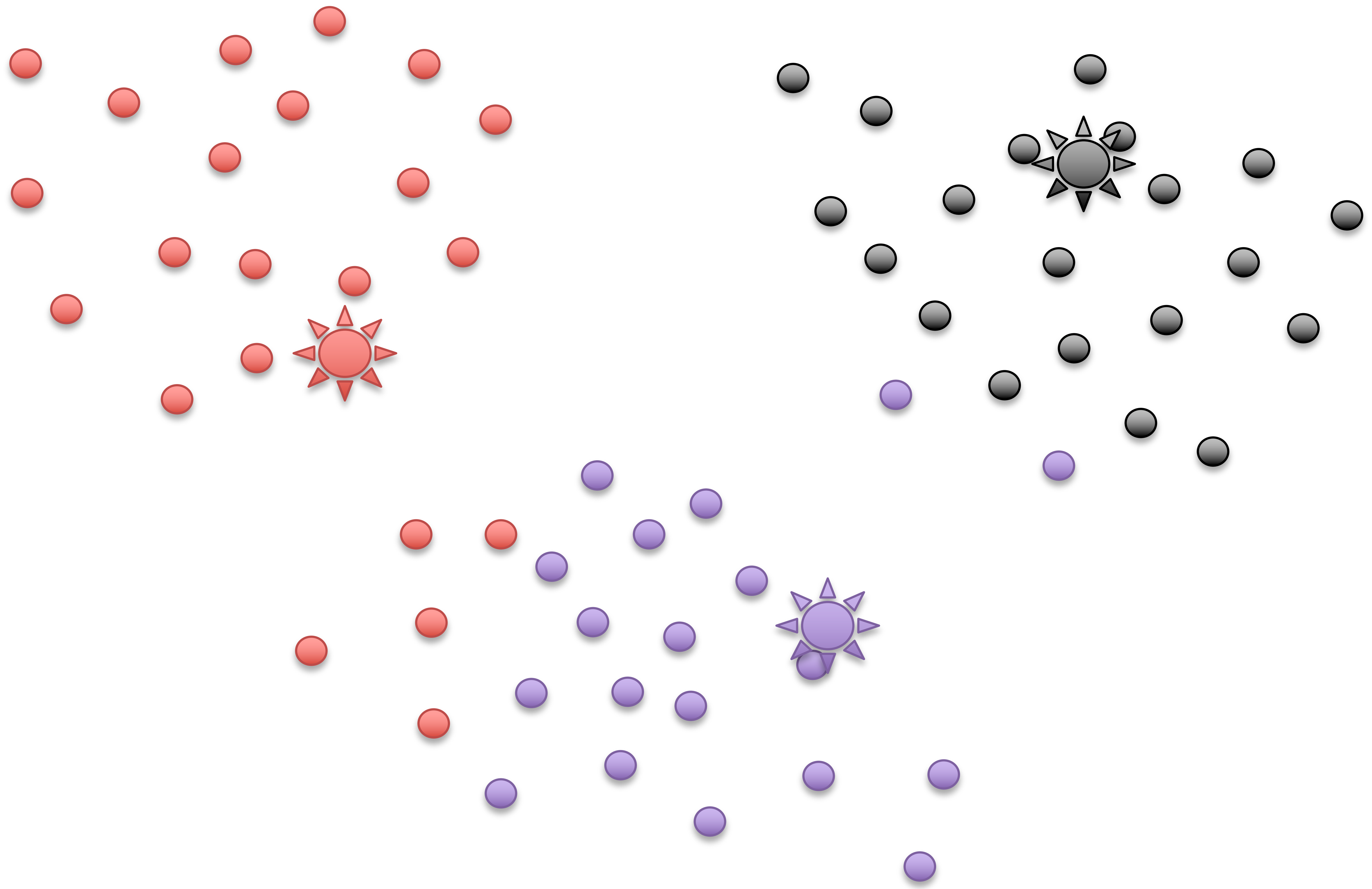
Centroid-based (K-means) clustering



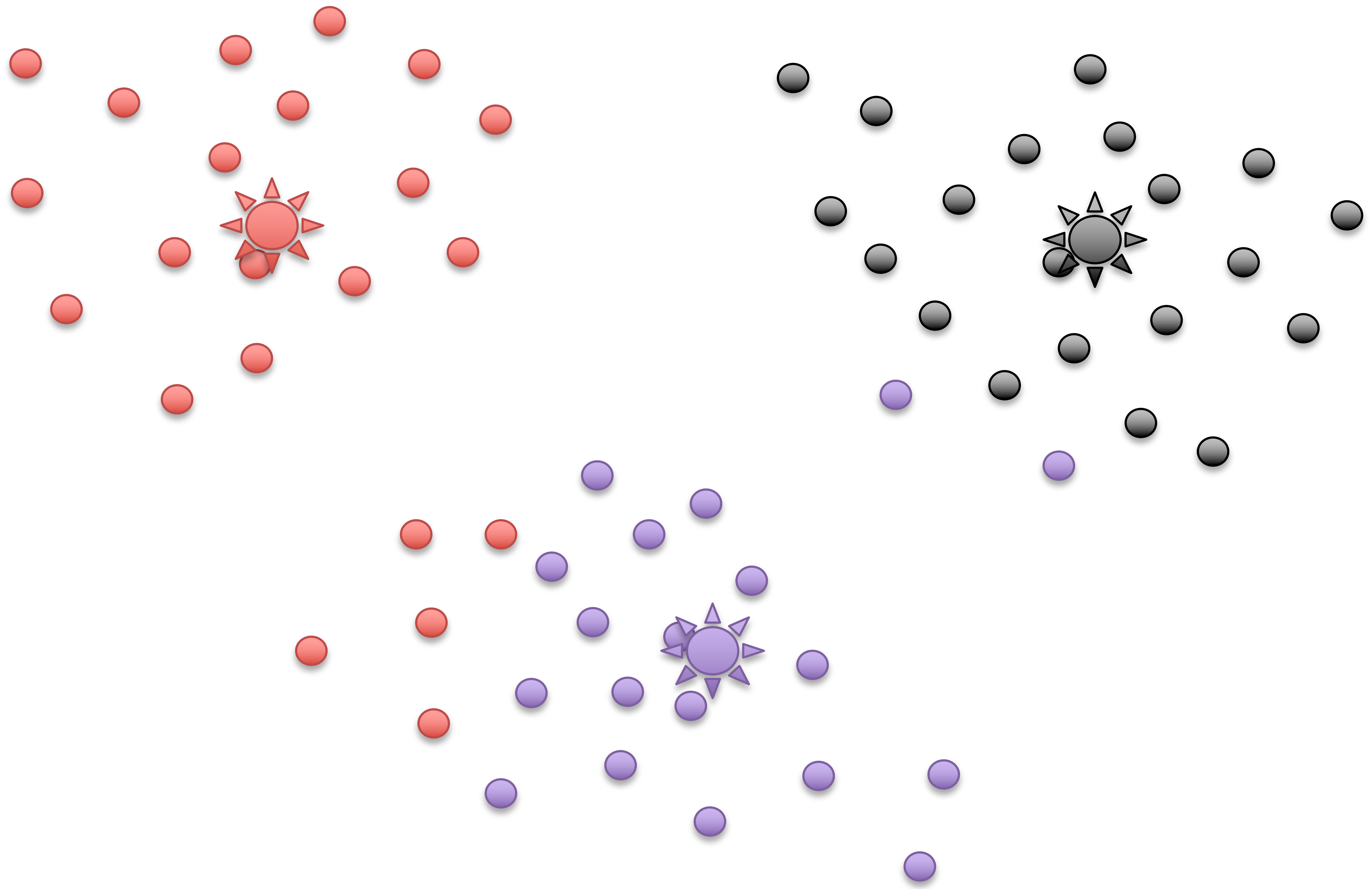
Centroid-based (K-means) clustering



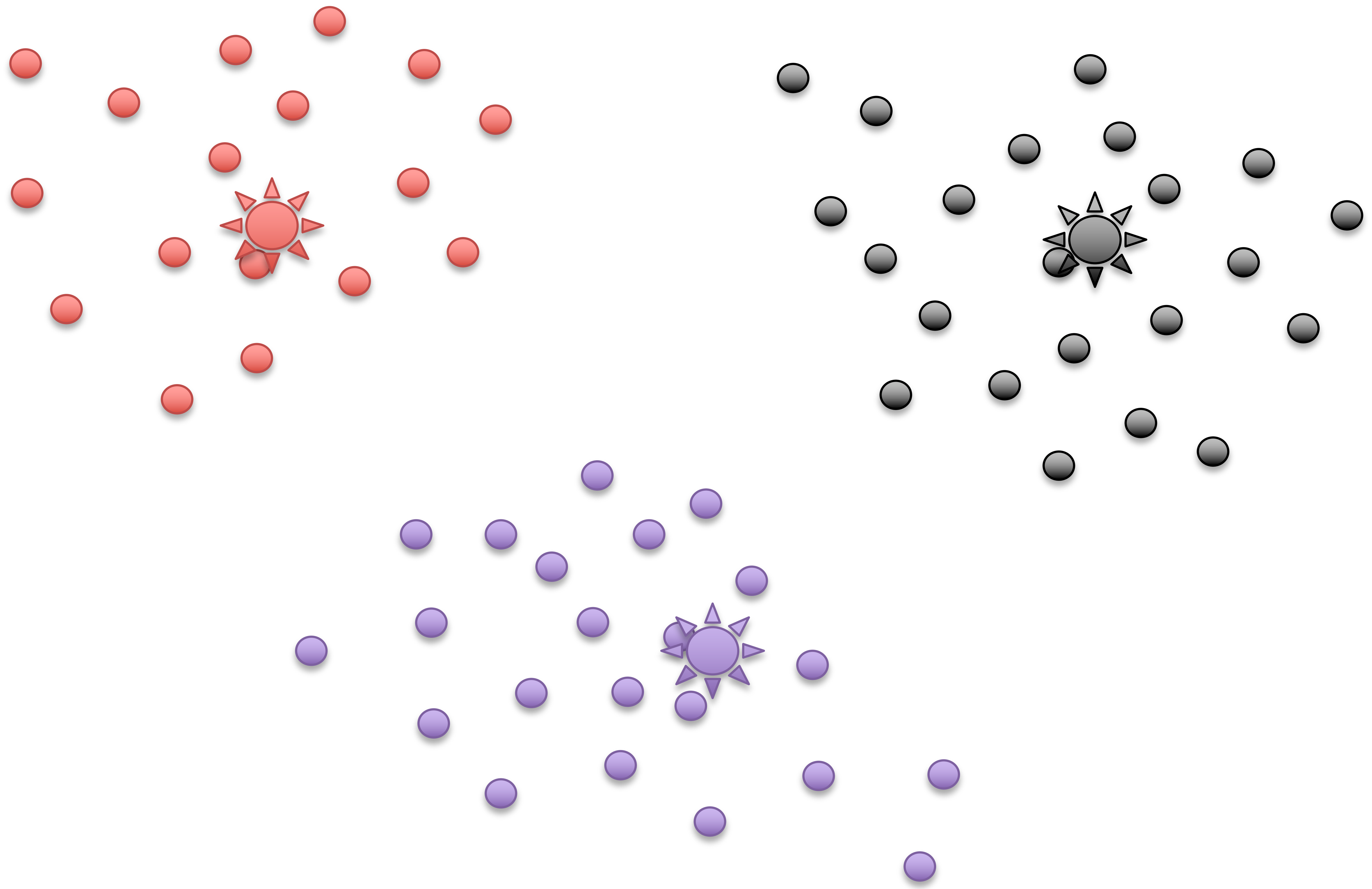
Centroid-based (K-means) clustering



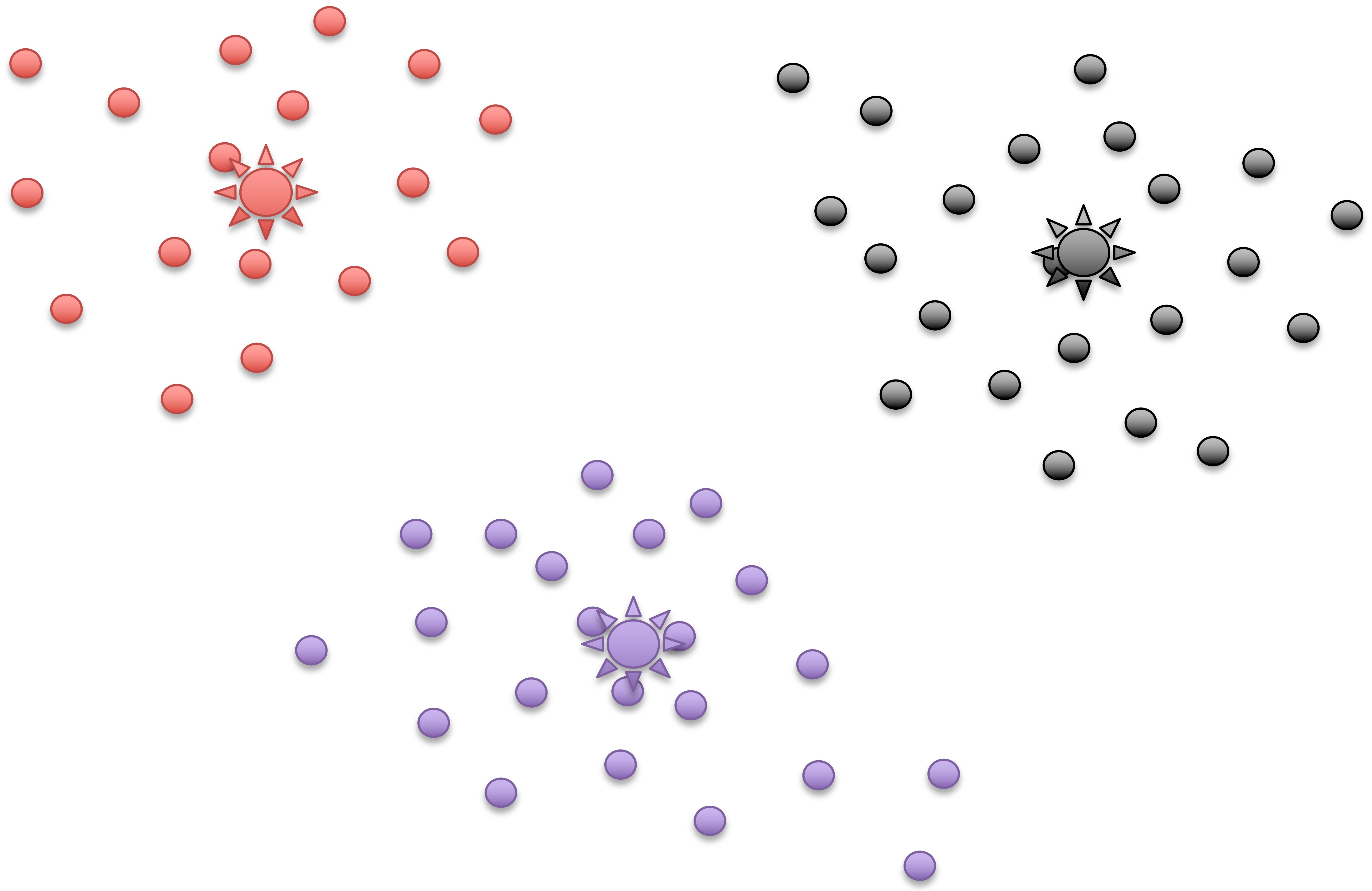
Centroid-based (K-means) clustering



Centroid-based (K-means) clustering



Centroid-based (K-means) clustering



K-means objective

- Given $S = \{x_i\}_{i=1}^N$, find

$$\underset{S=C_1\cup\cdots\cup C_k, \{c_1,\ldots,c_k\}}{\operatorname{argmin}} \sum_k \sum_{x\in C_k} \|x - c_k\|^2$$

$$\underset{S=C_1\cup\cdots\cup C_k}{\operatorname{argmin}} \sum_k |C_k| \operatorname{var}(C_k)$$

EM algorithm for K-means

$$\underset{S=C_1\cup\cdots\cup C_k,\{c_1,\ldots,c_k\}}{\operatorname{argmin}} \sum_k \sum_{x\in C_k} \|x - c_k\|^2$$

$$S = \{x_i\}_{i=1}^N$$

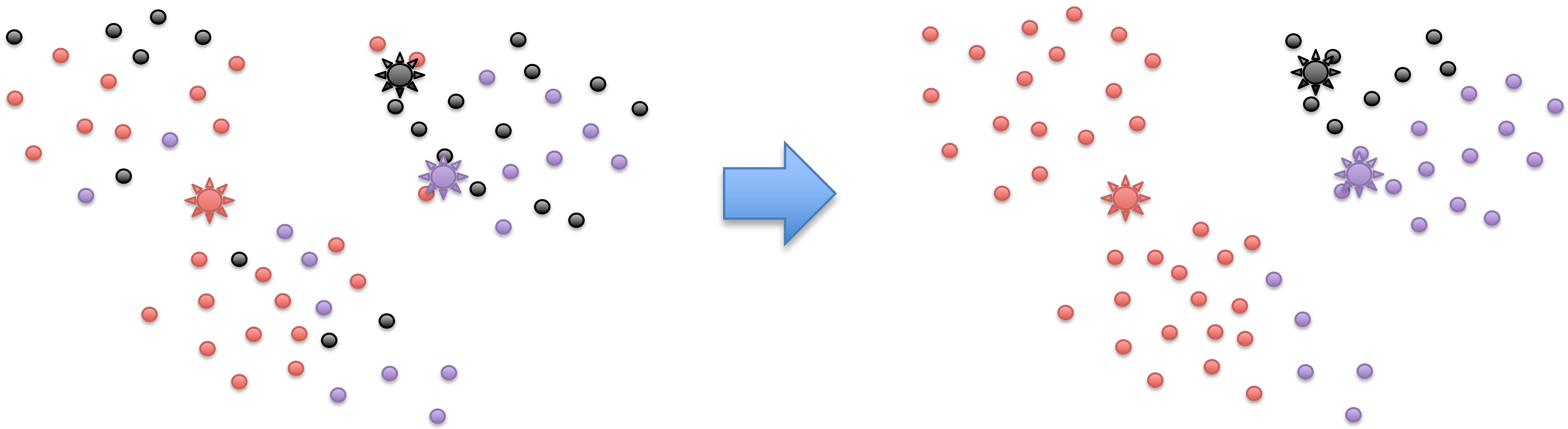
- Expectation/maximization (EM) algorithm
- E step:
  - Estimate clusters C_k
  - Estimate cluster membership
- M step:
  - Estimate cluster centers c_k
  - Estimate model parameters

E step

$$\operatorname*{argmin}_{S=C_1\cup\cdots\cup C_k,\{c_1,\ldots,c_k\}} \sum_k \sum_{x\in C_k} \|x - c_k\|^2$$

- For each x,
  - Assign to cluster C_k with smallest distance to c_k

$$S = \{x_i\}_{i=1}^N$$

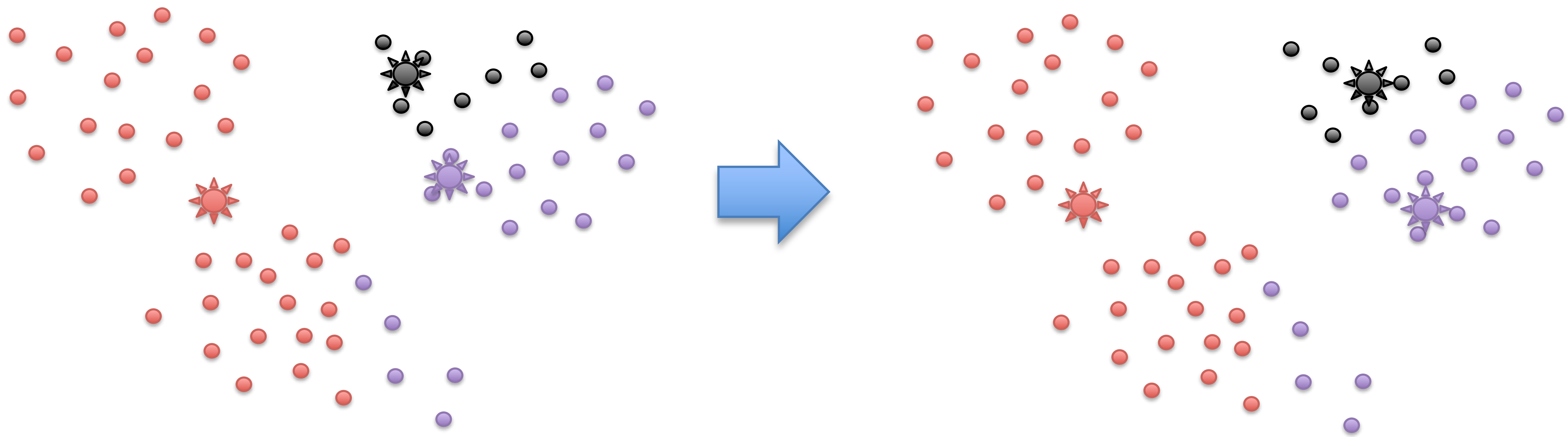


M step

- For each c_k,
  - Compute $c_k = \text{mean}(C_k)$

$$\underset{S=C_1\cup\cdots\cup C_k,\{c_1,\ldots,c_k\}}{\operatorname{argmin}} \sum_k \sum_{x\in C_k} \|x - c_k\|^2$$

$$S = \{x_i\}_{i=1}^N$$



More advanced clustering algorithms

- DBSCAN: density-based clustering with some advantages over K-means
  - Does not require one to specify the number of clusters in the data a priori
  - Can find arbitrarily-shaped clusters
  - Has a notion of noise, and is robust to outliers
  - Just two parameters and mostly insensitive to the ordering of the data points

https://scikit-learn.org/stable/modules/clustering.html

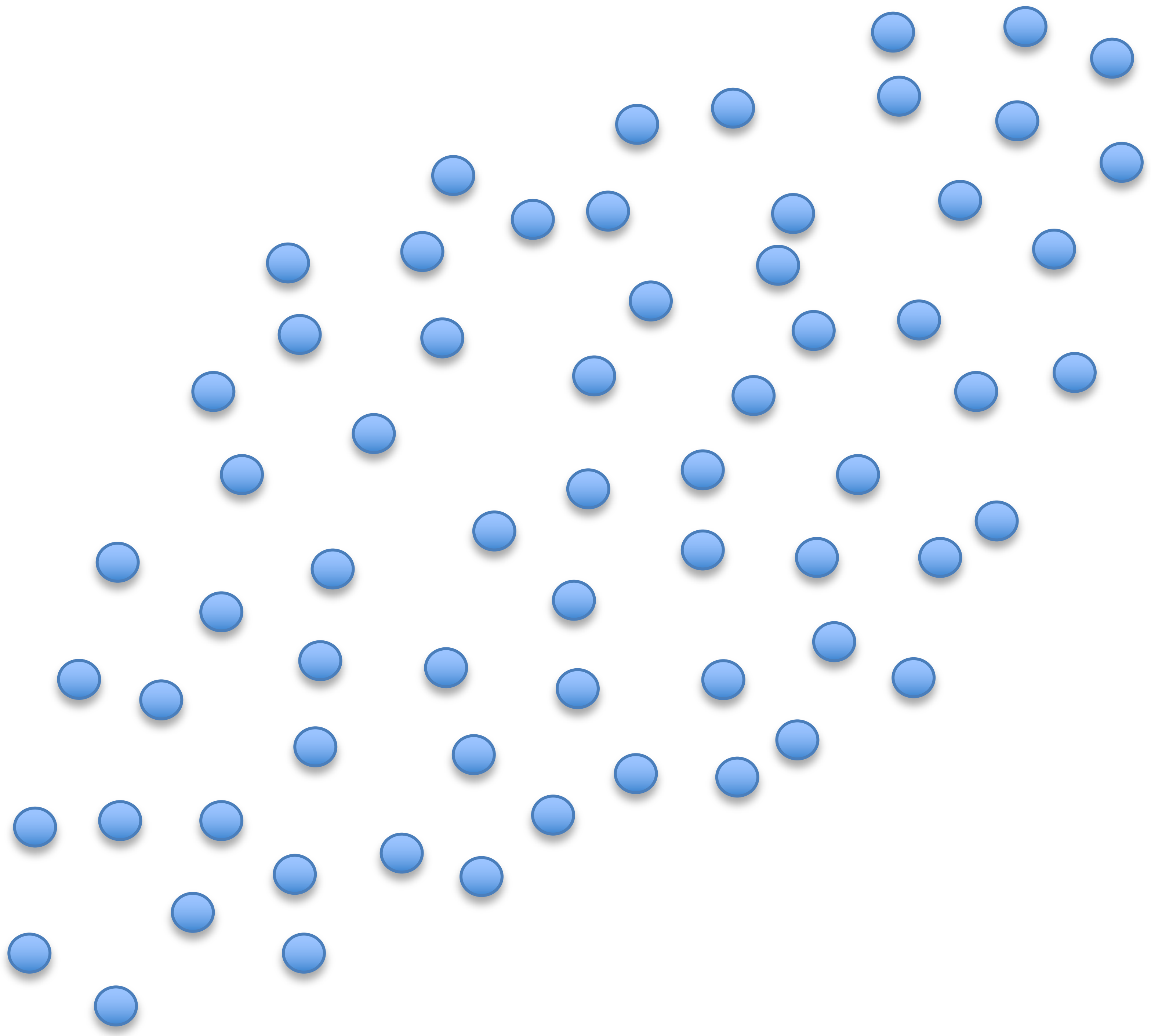
DBSCAN

k-means

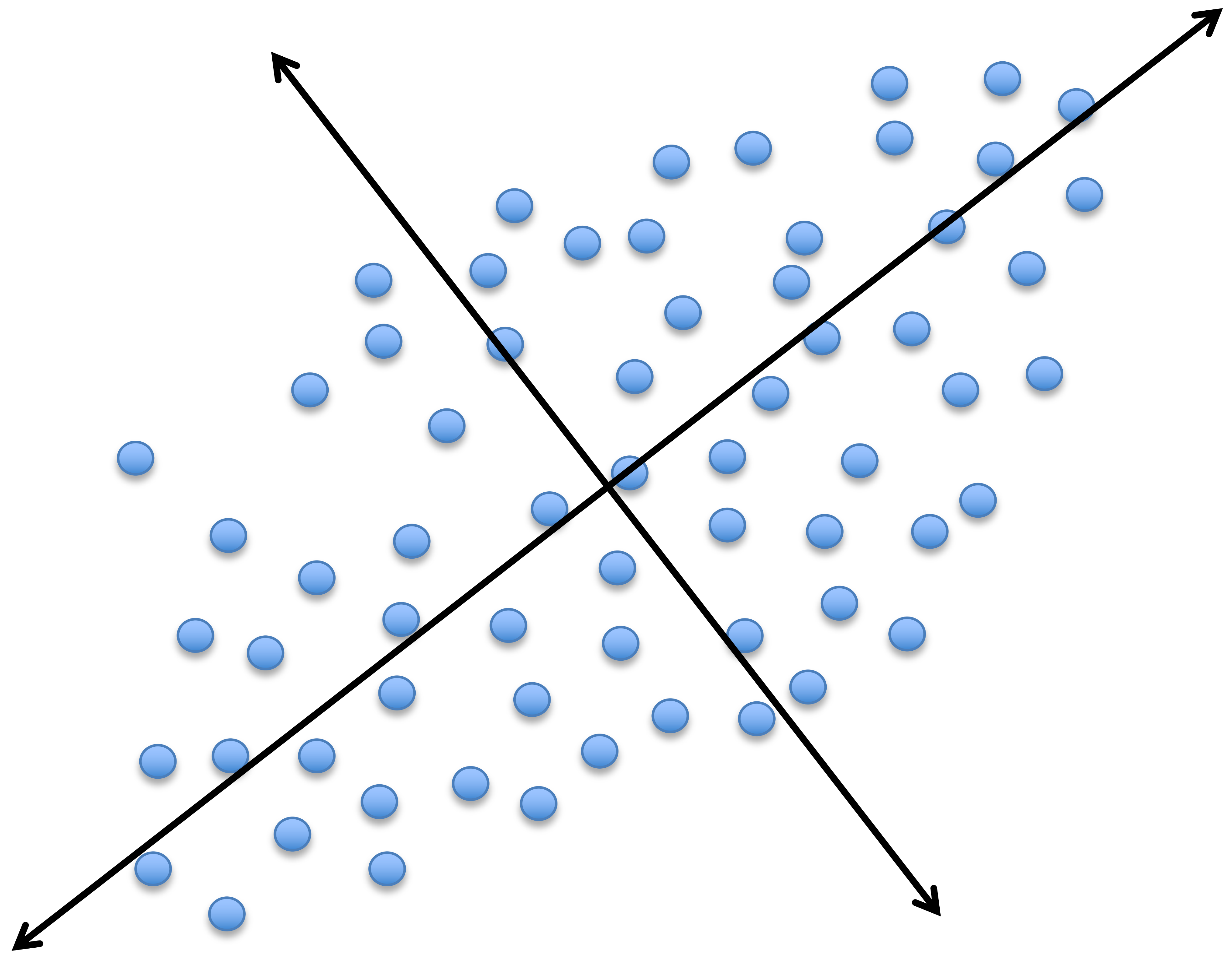
Recap: K-means

- Centroid-based clustering
  - Defines clusters using a notion of centrality
  - e.g. all items in the cluster must be close to each other
- Solve using EM algorithm
  - Also probabilistic variant (Gaussian mixture models)
- Useful when centrality assumption is good
- Other (density-based) variants include DBSCAN

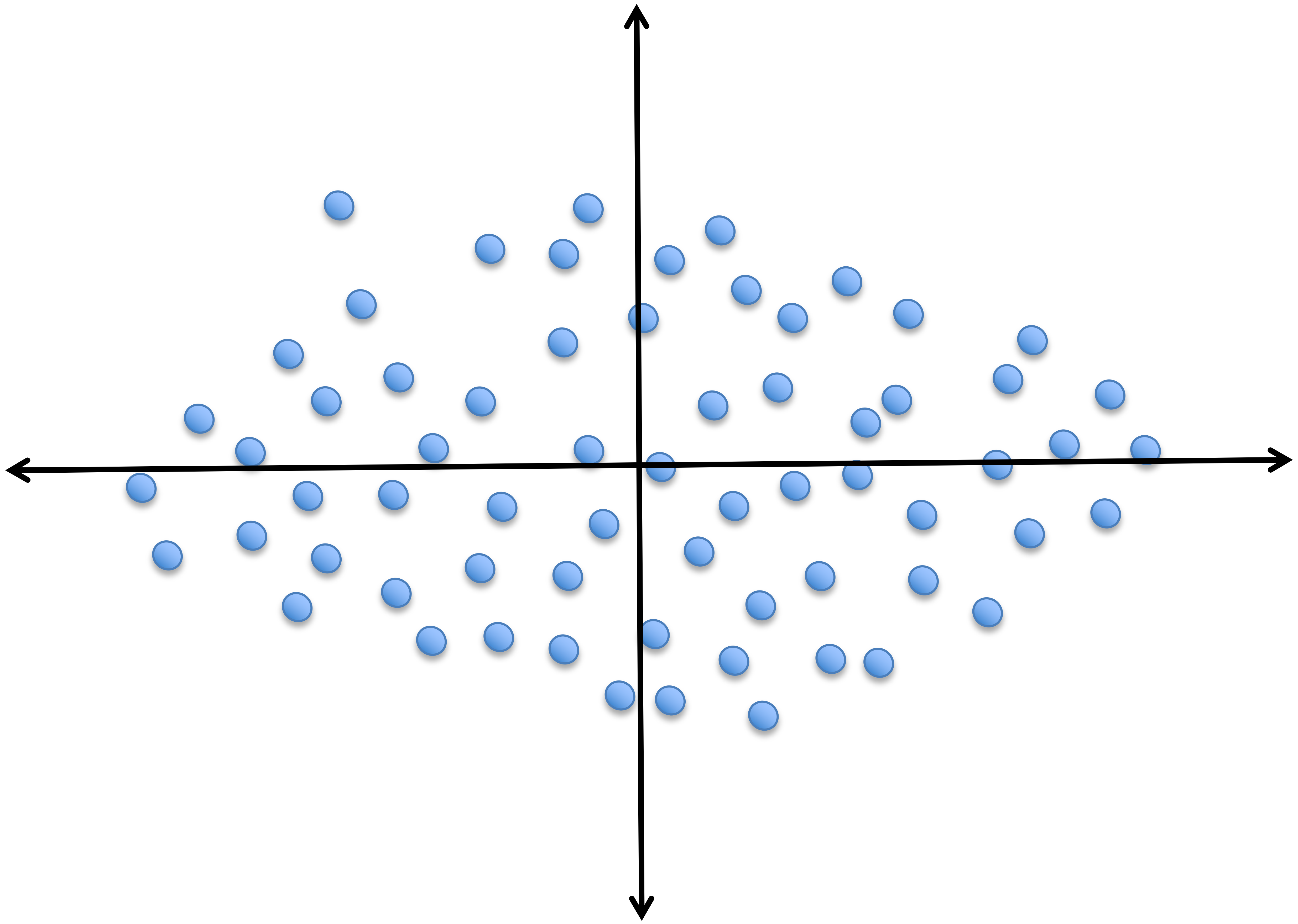
Limitations of clustering



Principal component analysis



Principal component analysis



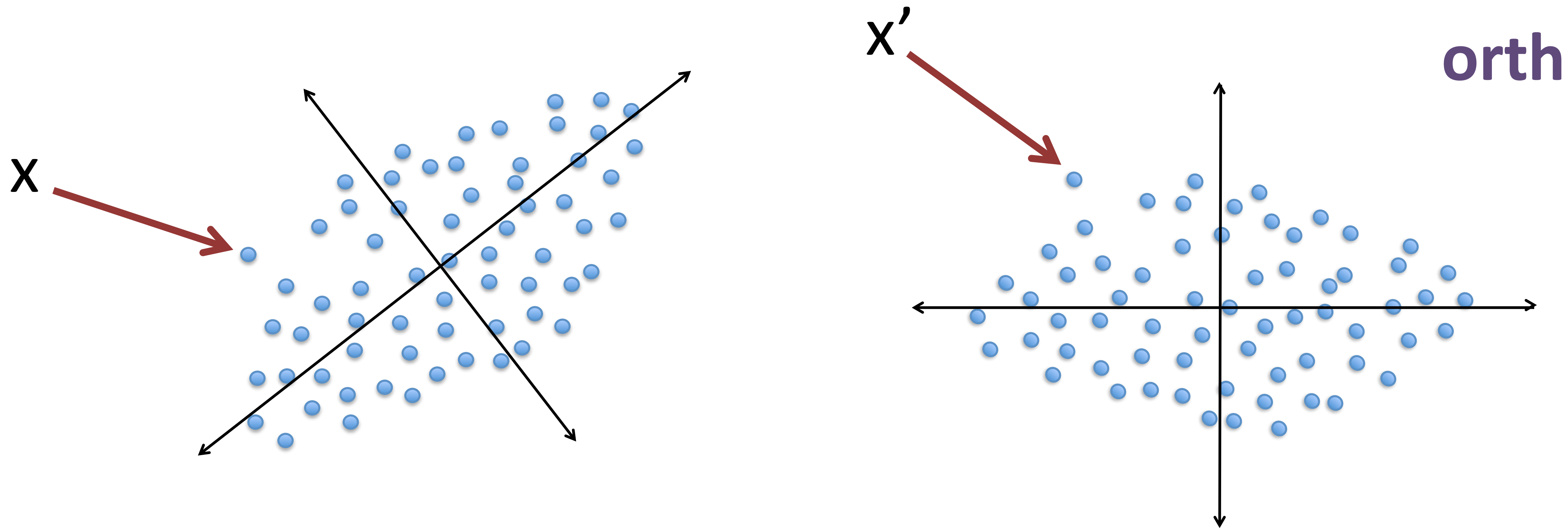
New Feature Representation!

Summarizing data

- Summarize data using smaller number of attributes
- Clustering: summarize data via clusters
  - K-means: summarize via cluster membership
- PCA: summarize via orthogonal projections
  - Define new feature representation
  - Rotation + projection

Orthogonal matrices

- A matrix U is orthogonal if $U^\intercal U = UU^\intercal = I$
  - For any column u: $u^\intercal u = 1$
  - For any two columns u, u': $u^\intercal u' = 0$
  - U is a rotation matrix, $U^\intercal$ is the inverse rotation
  - If $x' = U^\intercal x$, then $x = Ux'$



PCA finds a specific orthogonal U

Orthogonal matrix properties

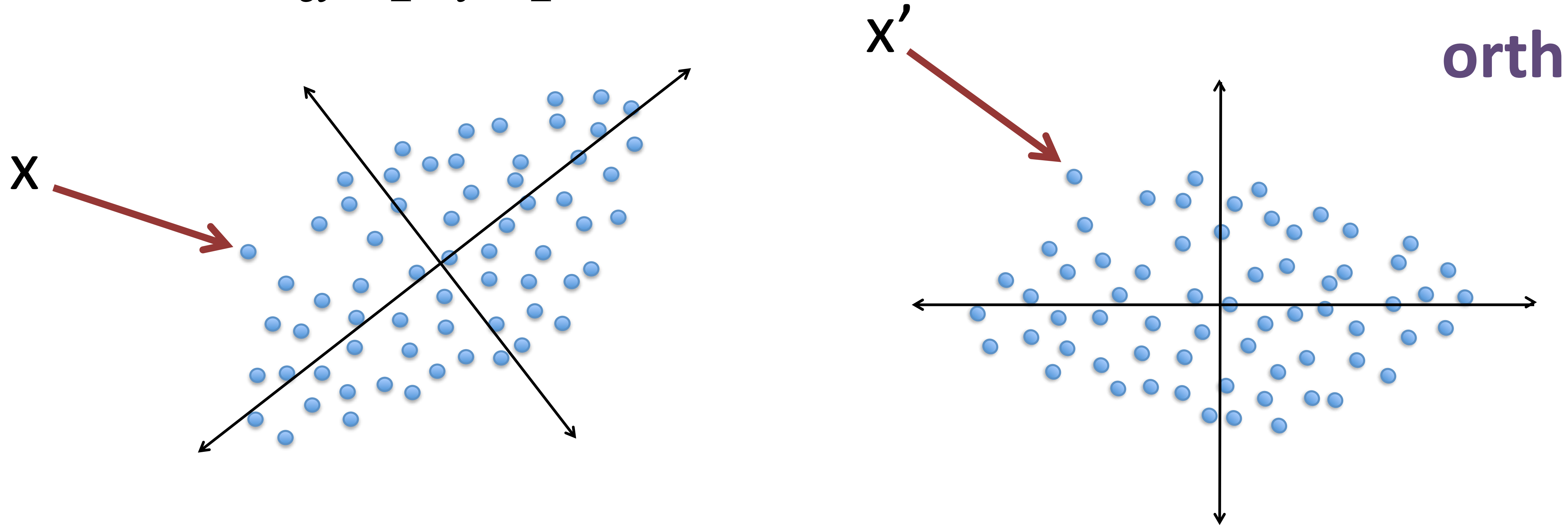
- Orthogonal transformation: $x' = U^{\intercal} x$

- Norm preserving:

$$(x')^{\intercal} x' = x^{\intercal} x$$

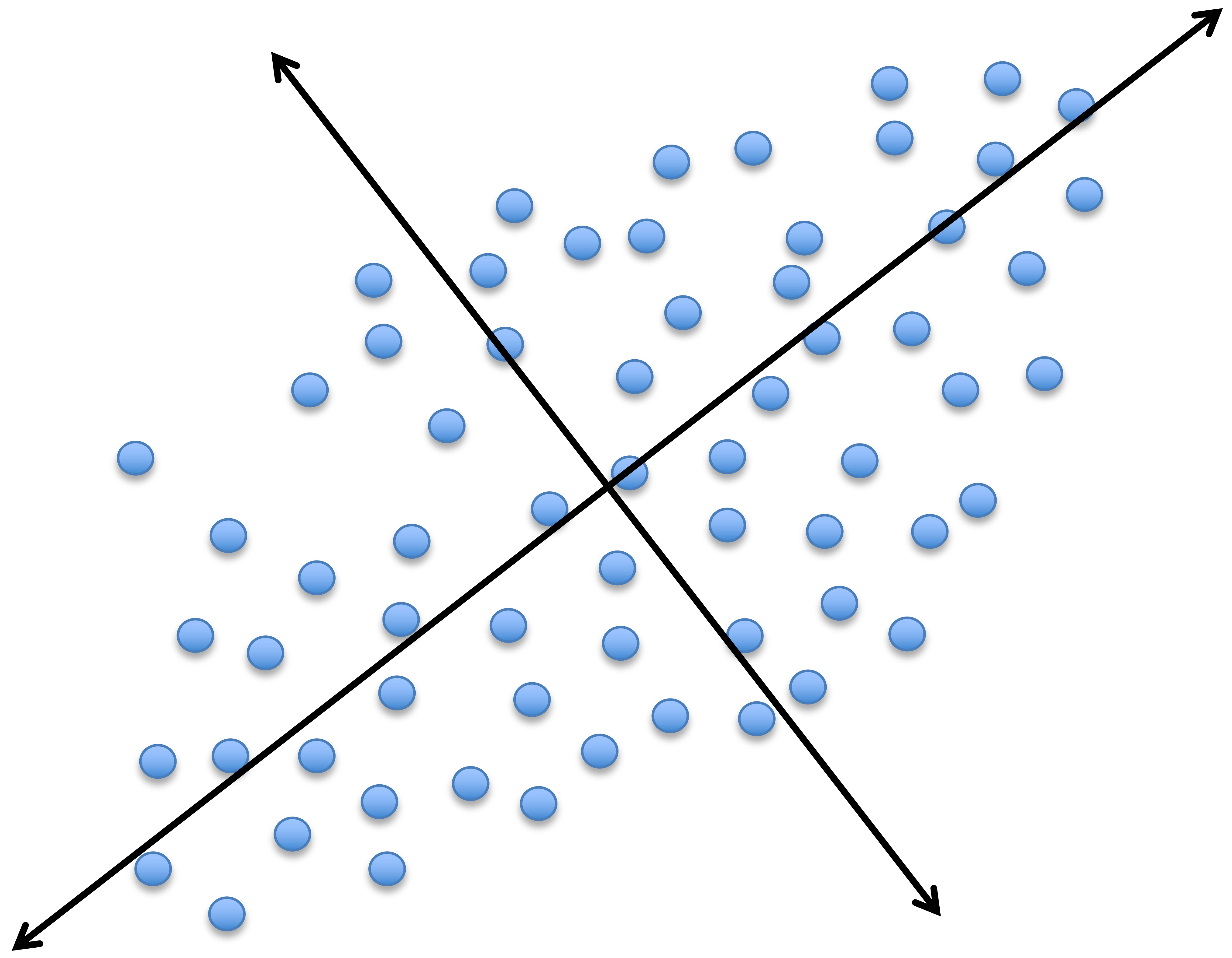
- Preserves total variance:

$$\sum_{d=1}^{D} \sum_{i=1}^{N} \left(x_i^{(d)}\right)^2 = \sum_{d=1}^{D} \sum_{i=1}^{N} \left(x_i^{\prime(d)}\right)^2$$ assuming zero mean



PCA finds a specific orthogonal U

Principal component analysis

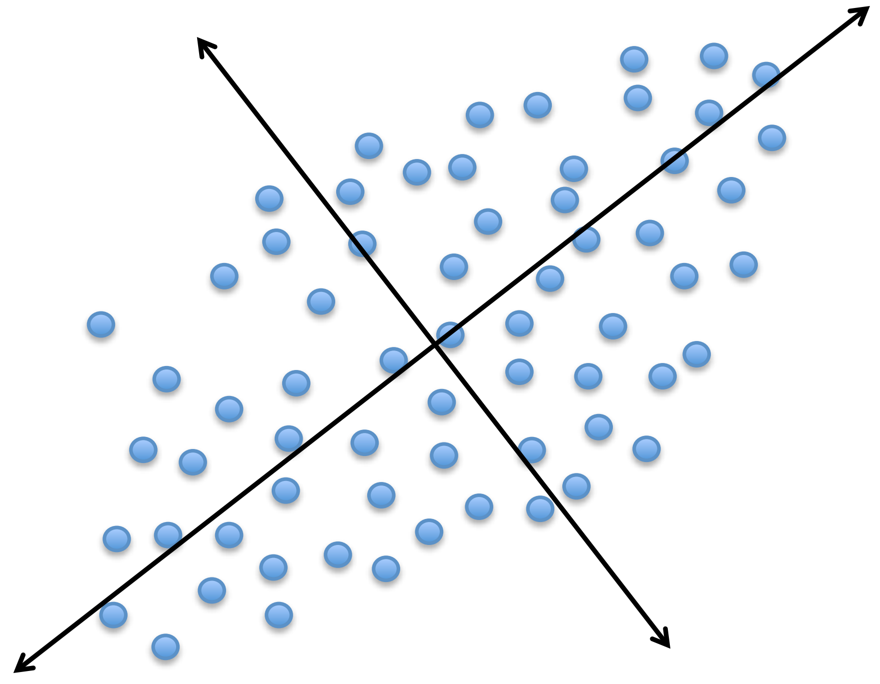


Principal component analysis

Summarize Using 1 Feature?

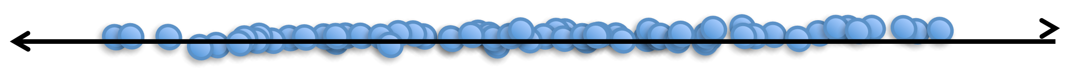
Principal component analysis

Summarize Using 1 Feature?



"projection"

$$u_1^T x$$



$$u_1 u_1^T x$$

"reconstruction"

Works with arbitrary subsets of columns of orthogonal U

E.g., $U' = [u_1, u_5, u_{20}]$

PCA definition

- Define X as the matrix of all data

$$X = [x_1, \ldots, x_N] \in \mathbb{R}^{D \times N}$$

- Mean-centered matrix

$$\overline{X} = X - [\bar{x}_1, \ldots, \bar{x}_N]$$

- PCA

$$\overline{X}\,\overline{X}^\intercal = U \Lambda U^\intercal$$

Symmetric

Orthogonal

Diagonal

PCA properties

- Assuming zero mean,

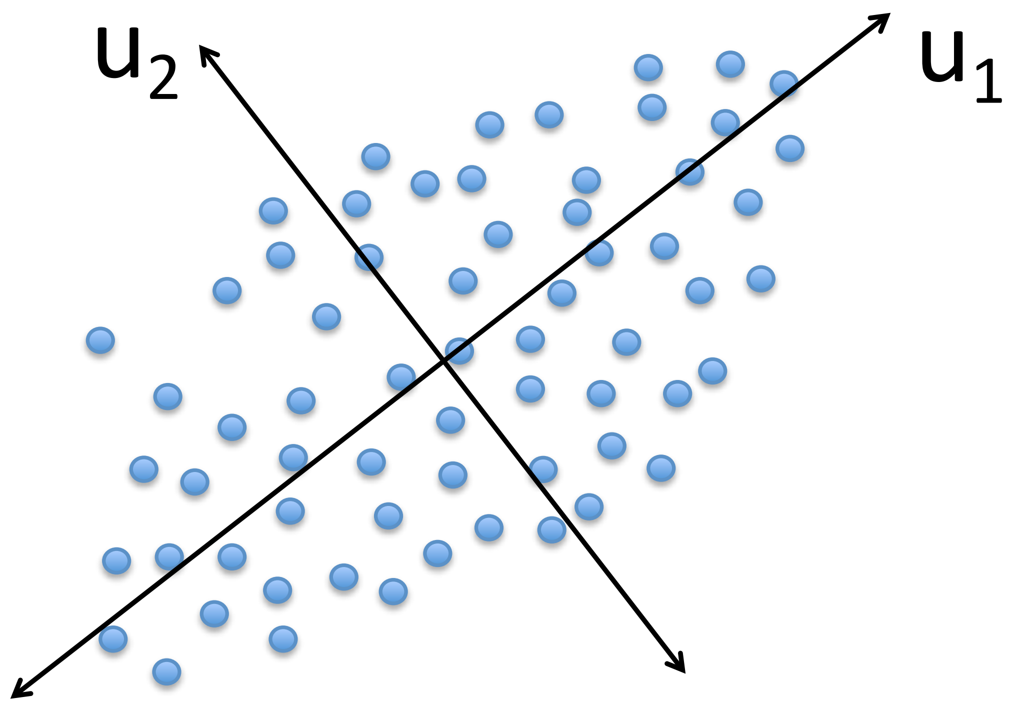
  $XX^\intercal = U\Lambda U^\intercal$

$$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \end{bmatrix}$$

- Each column of U if an eigenvector of $XX^\intercal$
- Each λ_i is an eigenvalue
  - $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D$

$$(XX^T)u_d = \lambda_d u_d$$

Interpretation



- Feature covariance matrix

$$\Sigma = XX^T = U\Lambda U^T$$

- $\Sigma_{dd'}$ is the covariance of features d and d' in the training data

- The first column u_1 is the single direction of greatest variance

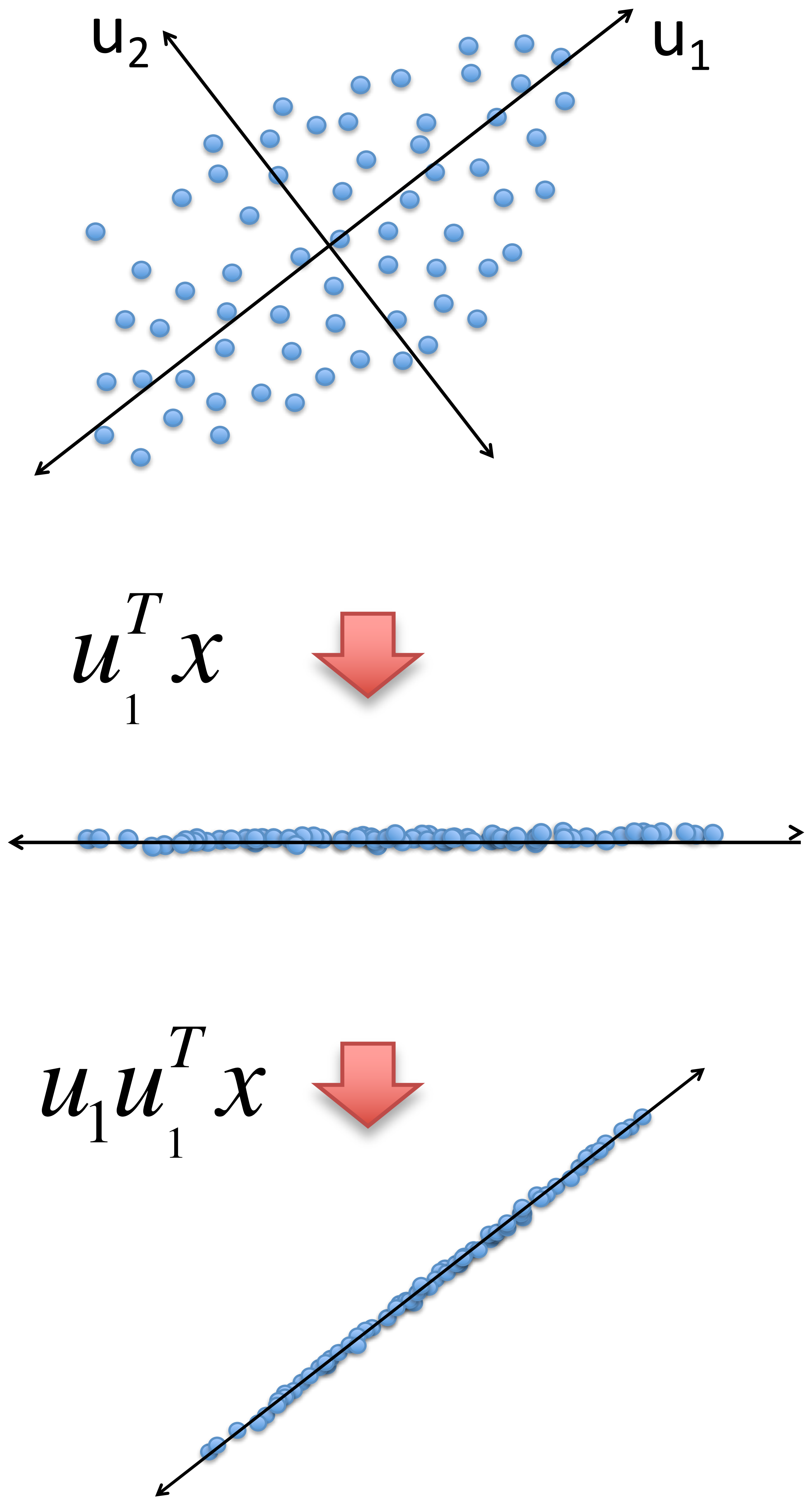
    - λ_1 is the total variance along u_1

$$\lambda_1 = \sum_{i=1}^{N} \left(u_1^T x_i\right)^2 = \sum_{i=1}^{N} \left(x_i^{'(1)}\right)^2$$

Interpretation (continued)

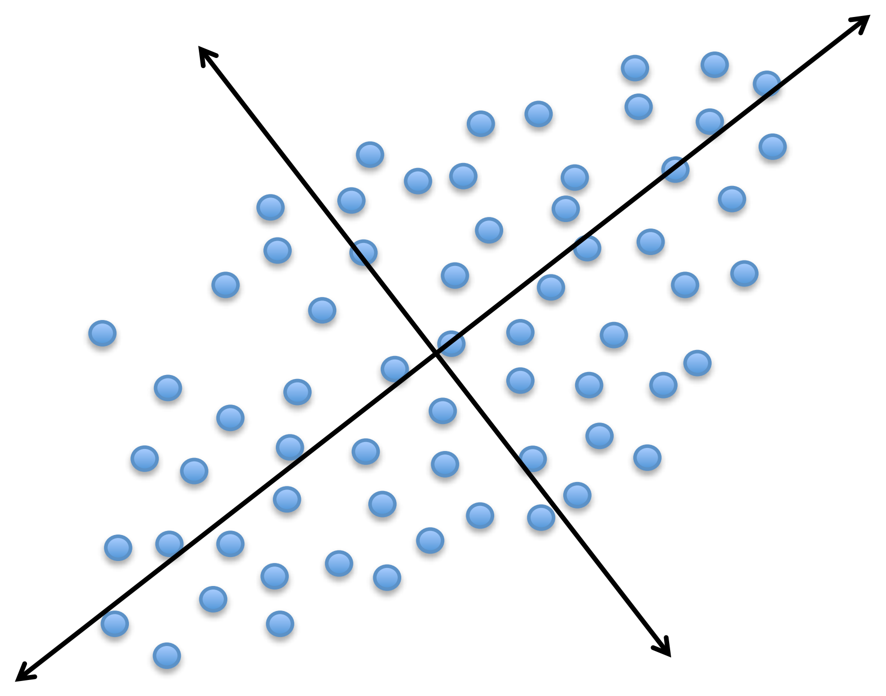
- The first column u_1 is the single direction that minimized the squared loss of reconstructing the original x features
  - It minimizes the amount of residual variation
- One can prove that:

$$u_1 = \underset{u:\ u^T u = 1}{\operatorname{argmin}} \sum_{i=1}^{N} \|x_i - u u^\top x_i\|^2$$



Interpretation (continued)

Find the u_1 that minimizes the residual squared norm:



-

$u_1 u_1^T x$

Solving PCA

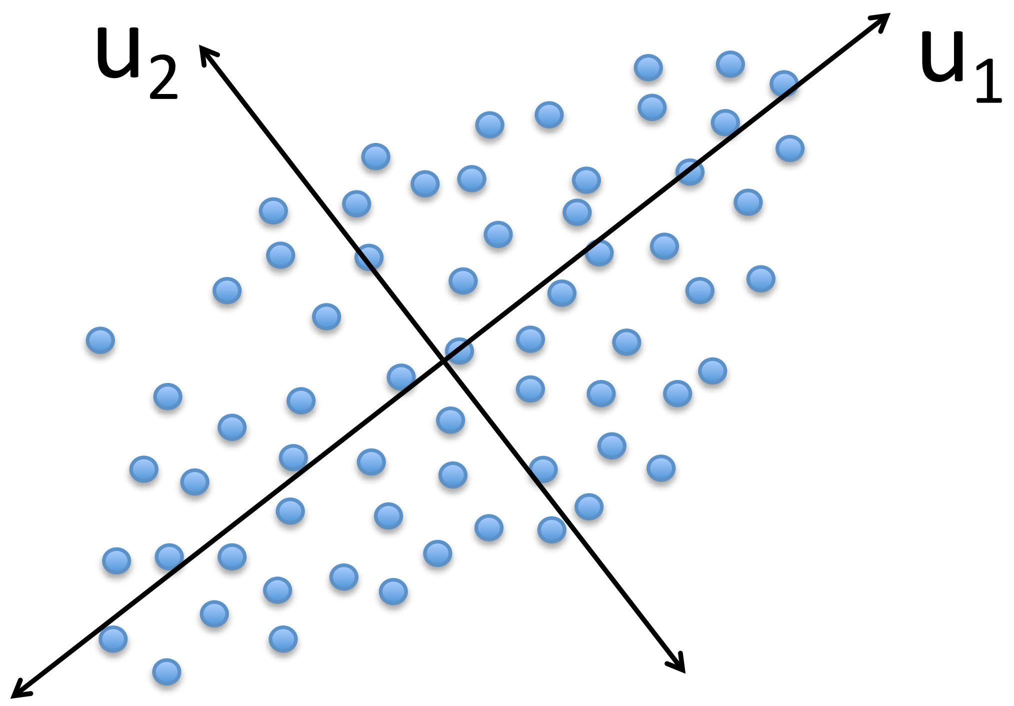
- Given $X = [x_1, \ldots, x_N] \in \mathbb{R}^{D \times N}$ (assuming zero mean)
- Initialize $X_1 = X$
- For $d = 1, \ldots, D$
  - Solve:
    $$u_d = \underset{u:\ u^T u = 1}{\operatorname{argmin}} \|X_d - uu^\intercal X_d\|^2$$
  - Update:
    $$X_{d+1} = X_d - uu^\intercal X_d$$

Dimensionality reduction with PCA

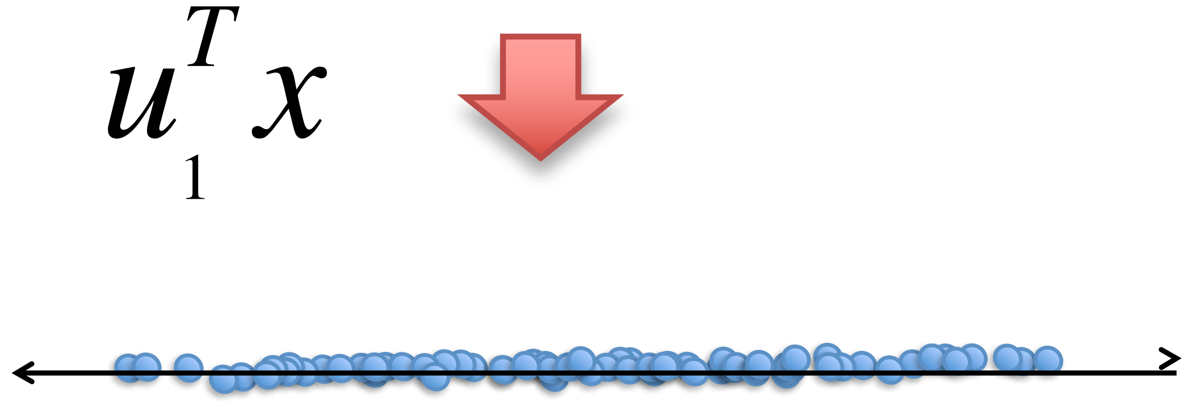
- PCA: $XX^\intercal = U\Lambda U^\intercal$
- The first K columns of U are guaranteed to be the K-dimensional subspace that captures the most variability of X
- Use first K columns of U to create a K -dimensional representation:

$$x' = U_{1:K}^\intercal x$$

- This creates a compact summary of the original dataset
  - e.g. $K = 50,\ D = 1{,}000{,}000$

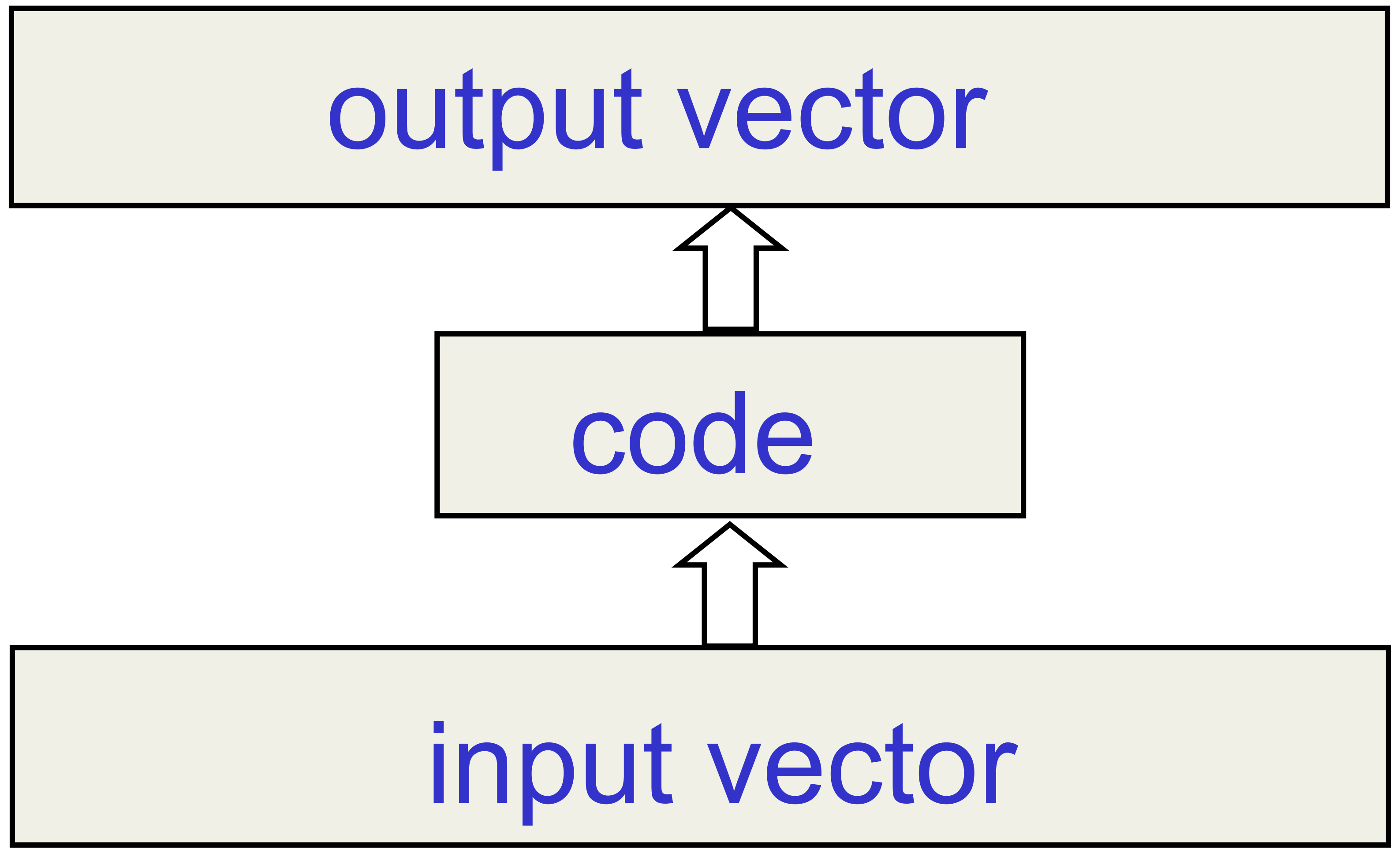


$u_1^T x$



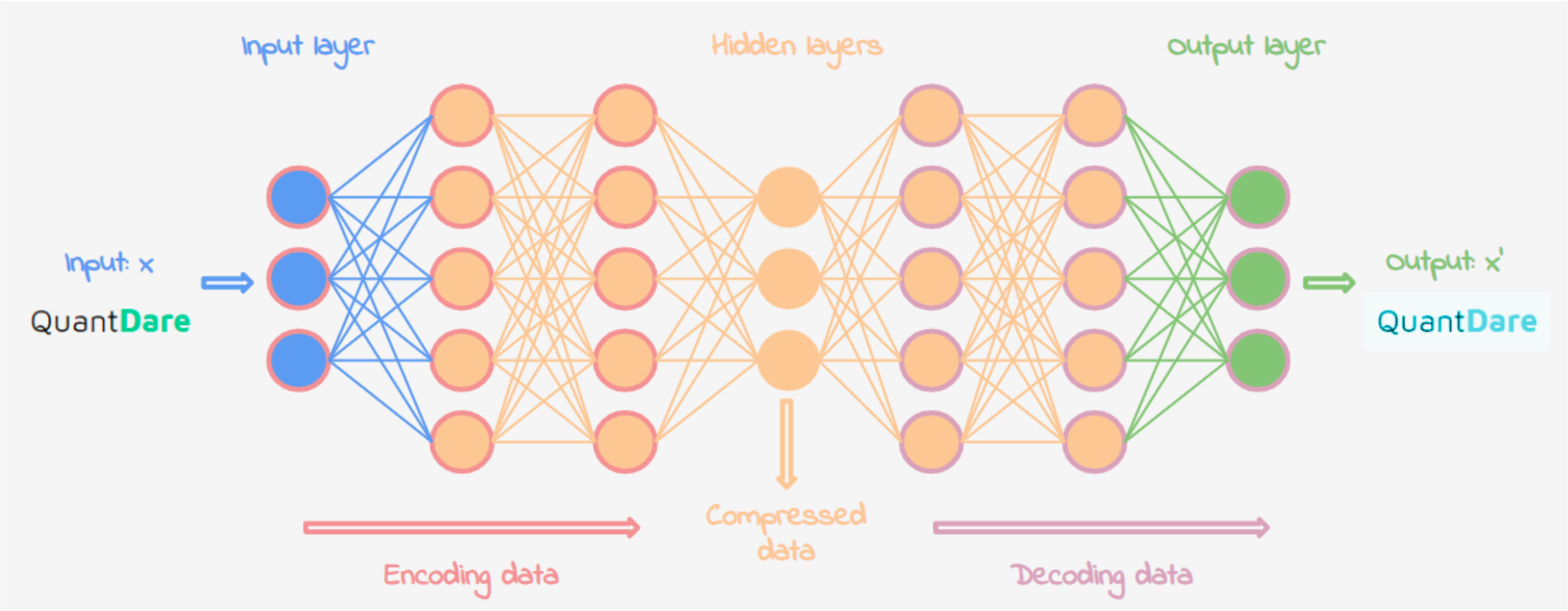
Backpropagation to implement PCA?

- Make the output the same as the input with a central bottleneck
- Activations of the hidden units form a "code"
- If the hidden and output layers are linear, hidden units are a linear function of the data and minimize the squared reconstruction error
    - This is exactly what PCA does!
- The K hidden units will span the same space as the first K components found by PCA
    - Their weight vectors may not be orthogonal
    - They will tend to have equal variances



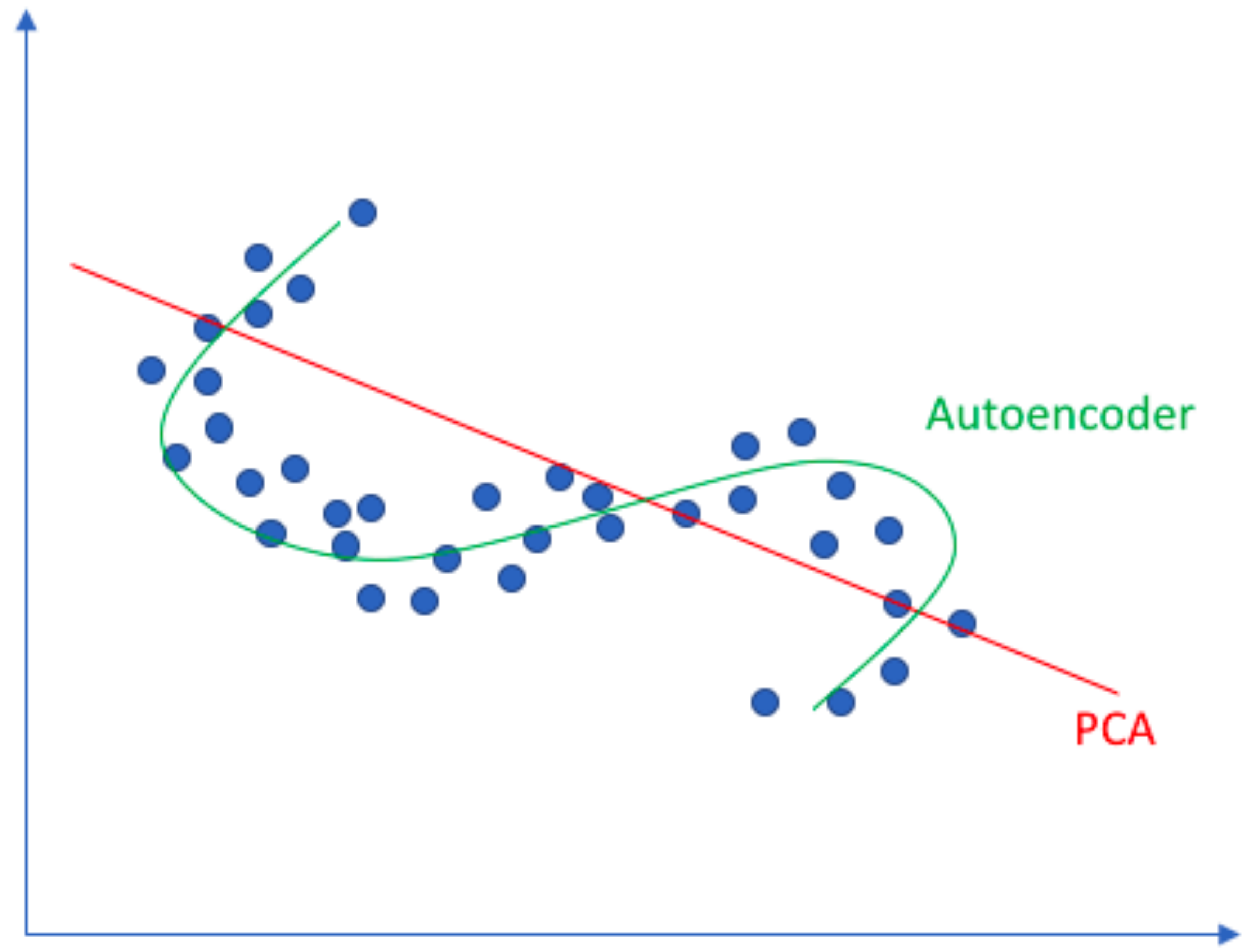
Autoencoders

- Feedforwrad neural networks with same input and output shapes
  - Goal: reconstruct original input by minimizing mean-squared error loss (or similar)



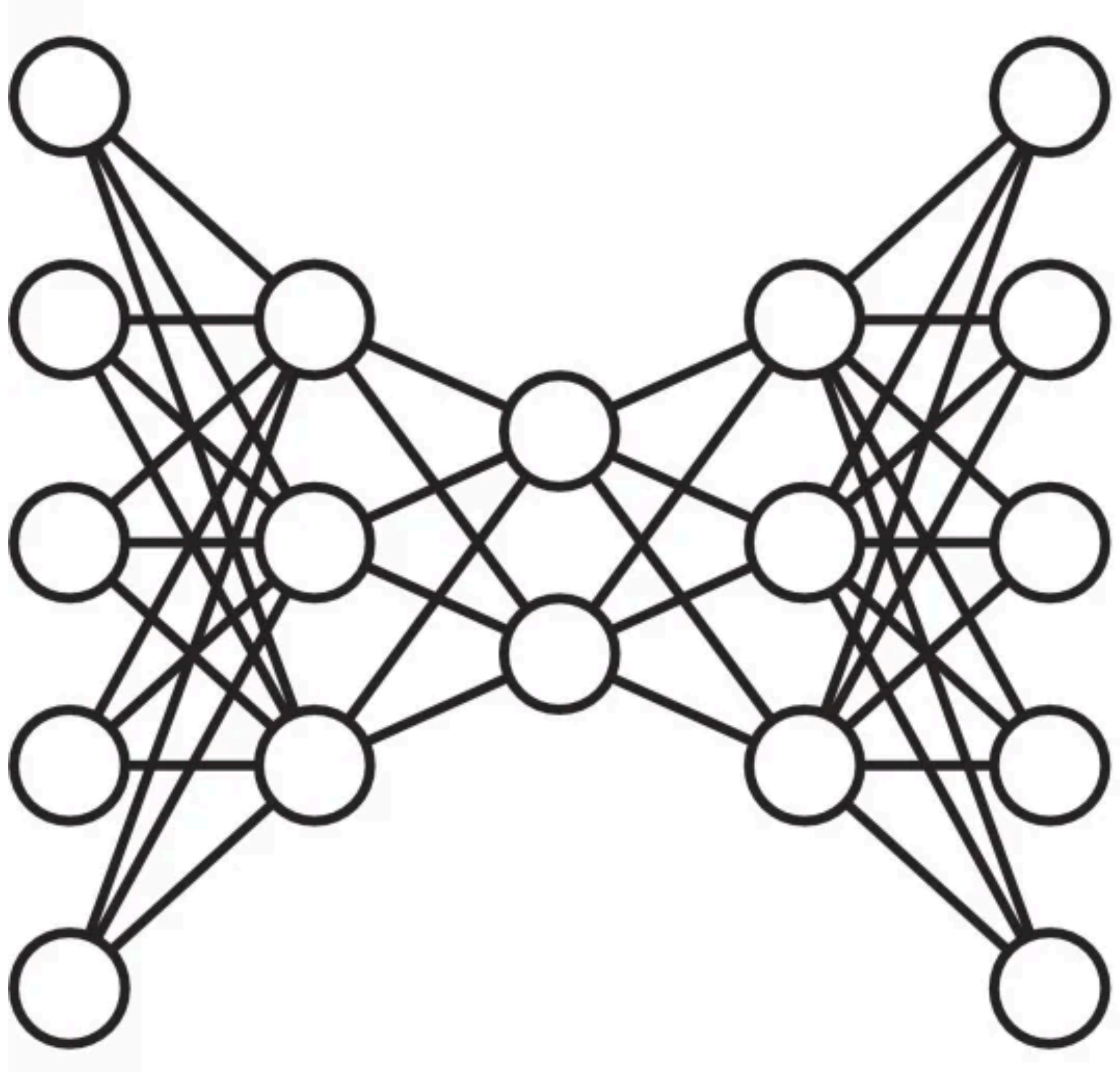
Autoencoders vs. PCA

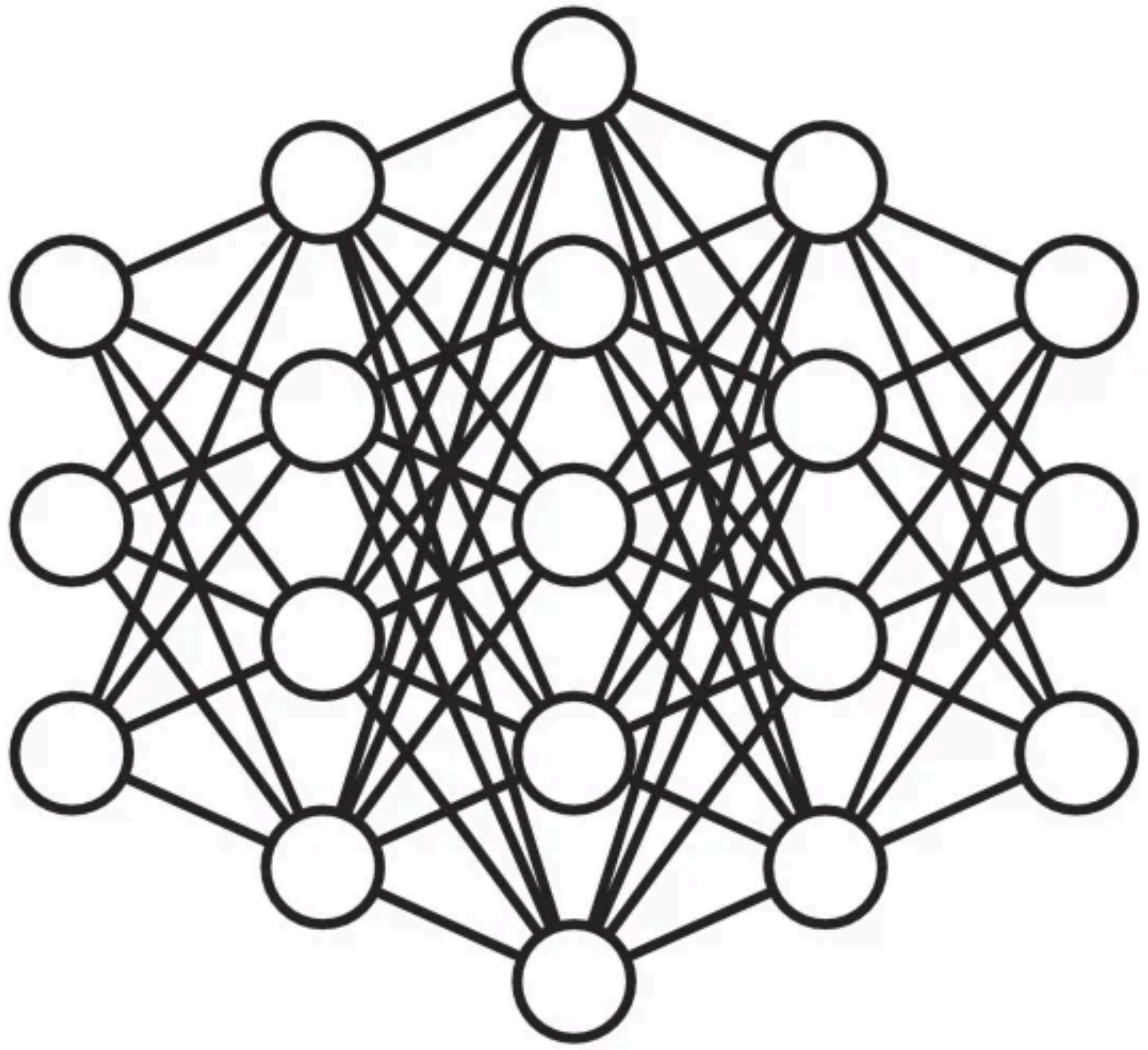
- Autoencoders can learn a nonlinear lower-dimensional manifold, while PCA attempts to discover a lower-dimensional hyperplane



Undercomplete vs. overcomplete

- Usually autoencoders are undercomplete (i.e. bottleneck layer that has some compression)
- Overcomplete ones are also possible typically with some reguarlization (such as L1 or sparsity conditions)



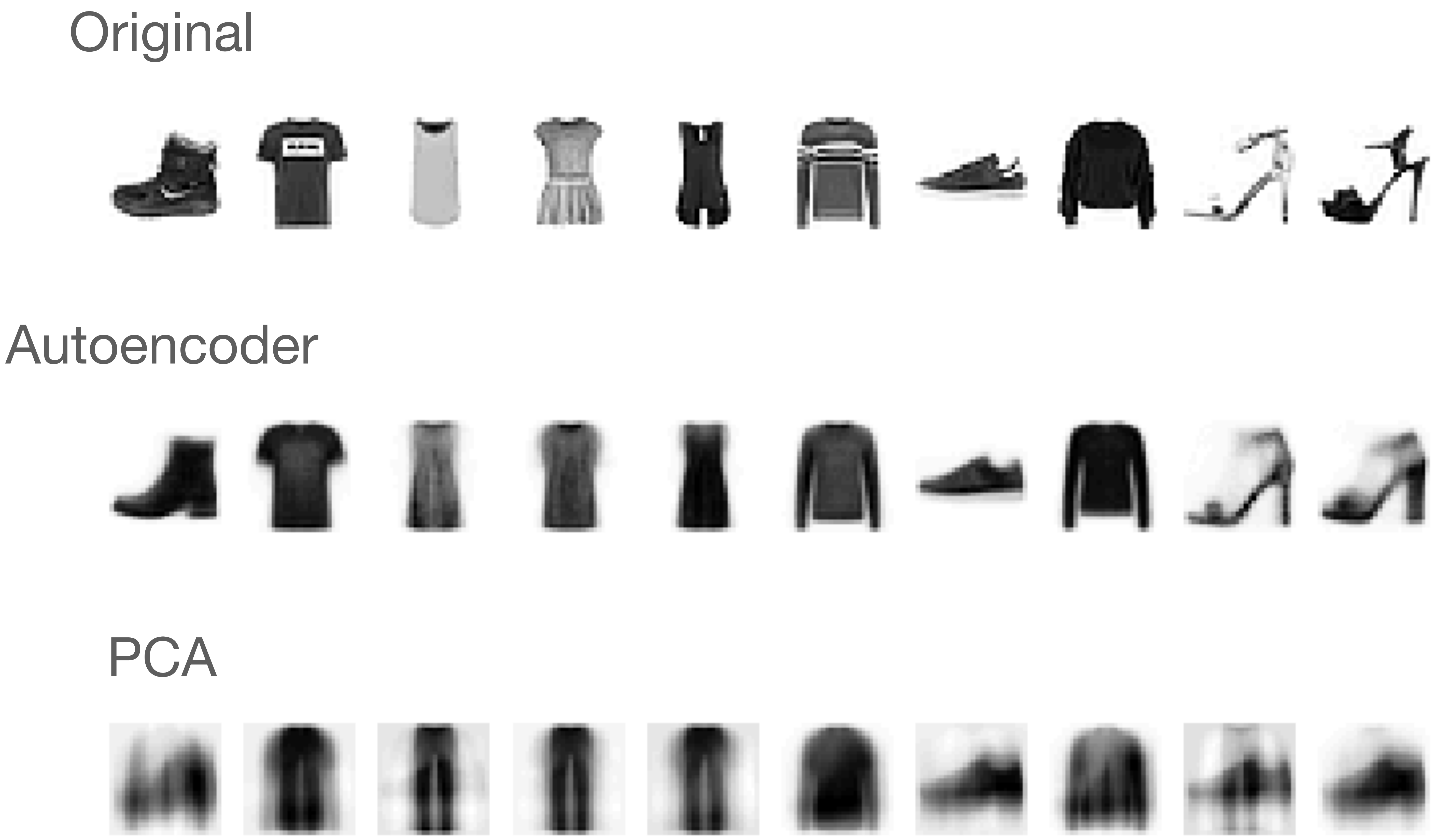


Example: fashion MNIST

https://rittikghosh.com/autoencoder.html

- Autoencoder in Keras:

```
m = Sequential()
m.add(Dense(1000, activation='relu',
            input_shape=(784,)))
m.add(Dense(500, activation='relu'))
m.add(Dense(250, activation='relu'))
m.add(Dense(32, activation='relu'))
m.add(Dense(2, activation='linear',
            name="bottleneck"))
m.add(Dense(32, activation='relu'))
m.add(Dense(250, activation='relu'))
m.add(Dense(500, activation='relu'))
m.add(Dense(1000, activation='relu'))
m.add(Dense(784, activation='sigmoid'))
```



Example: fashion MNIST

https://rittikghosh.com/autoencoder.html

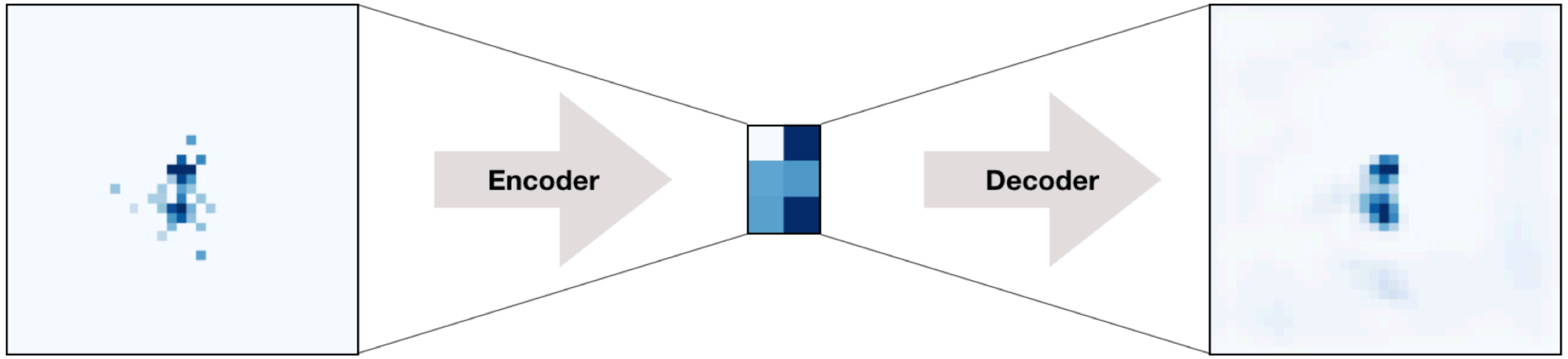
- Autoencoder in Keras:

```
m = Sequential()
m.add(Dense(1000, activation='relu',
            input_shape=(784,)))
m.add(Dense(500, activation='relu'))
m.add(Dense(250, activation='relu'))
m.add(Dense(32, activation='relu'))
m.add(Dense(2, activation='linear',
            name="bottleneck"))
m.add(Dense(32, activation='relu'))
m.add(Dense(250, activation='relu'))
m.add(Dense(500, activation='relu'))
m.add(Dense(1000, activation='relu'))
m.add(Dense(784, activation='sigmoid'))
```

Application: outlier (anomaly) detection

Credit: B. Nachman
https://indico.cern.ch/event/1188153/

- Autoencoders compress data and then uncompress it
  - Autoencoder is trained on background (and has good reconstruction performance)
  - If x is far from $\mathrm{AE}(x) = \mathrm{Decoder}(\mathrm{Encoder}(x))$, then x has low $p_{\mathrm{bkgd}}(x)$
- Directly use “reconstruction loss” $L(x, \mathrm{AE}(x))$ as an anomaly score



Next time

- Weakly supervised (QWoLa)
- Variational autoencoders