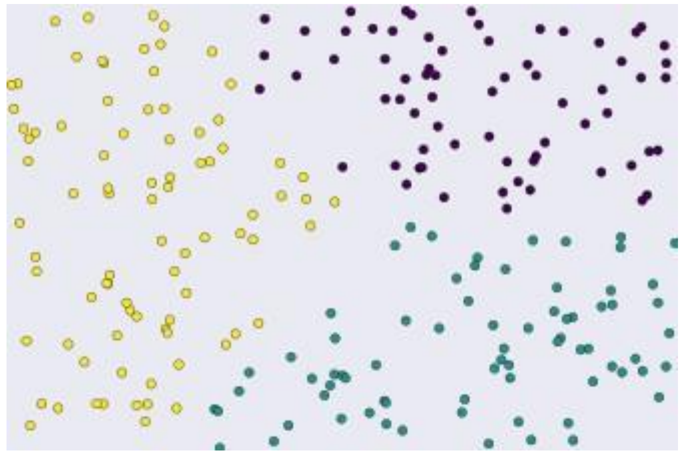
CARTE ML Workshop

Lecture 2-1: Unsupervised Learning and Visualization https://developers.google.com/machine-learning/clustering/overview

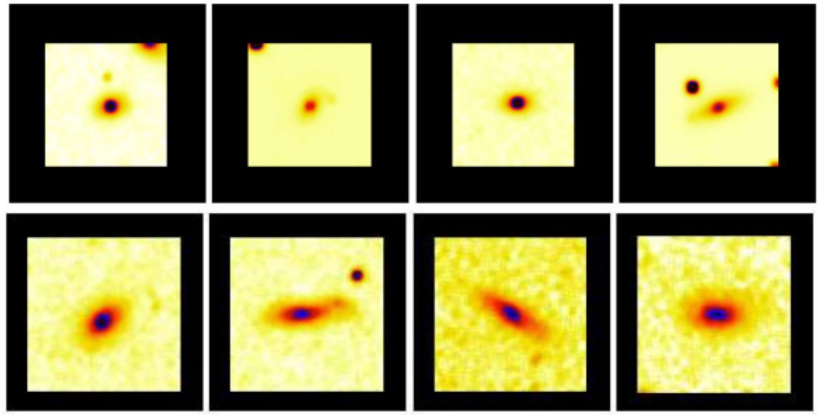


Clustering

The most common type of unsupervised learning

Goal: group “similar” data points together

Unsupervised because we don’t label the data as we did in classification/regression: let the features speak for themselves!

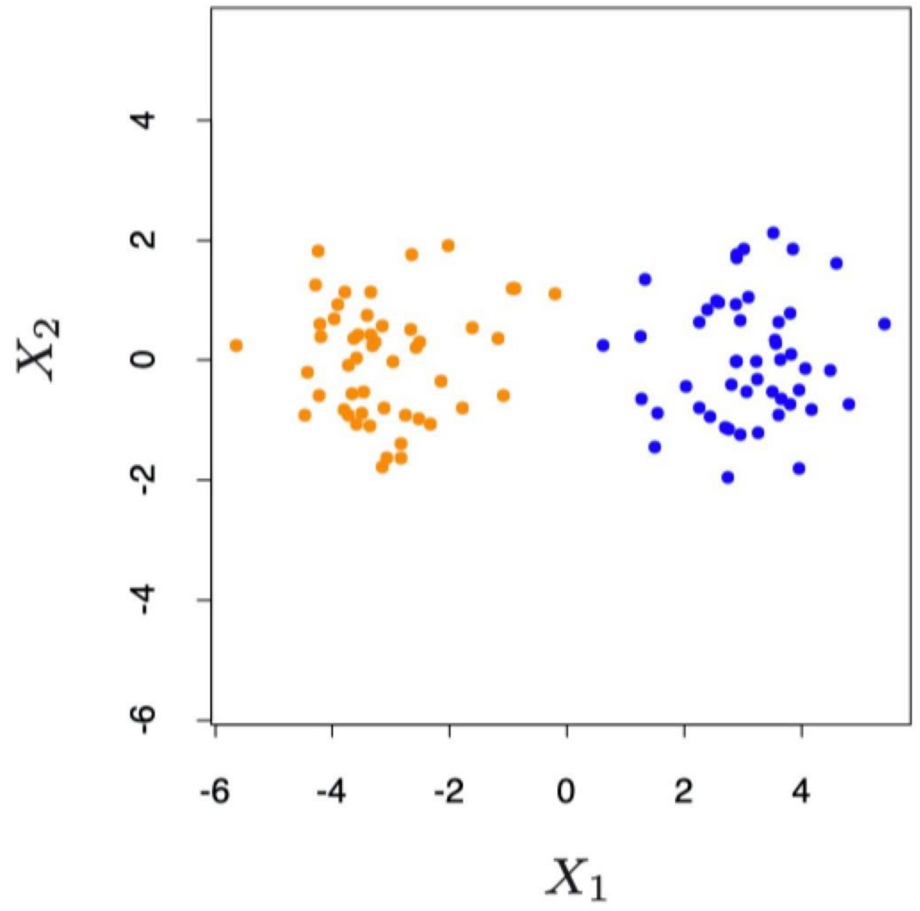


Clustering galaxies, from Miller et al. (2005)

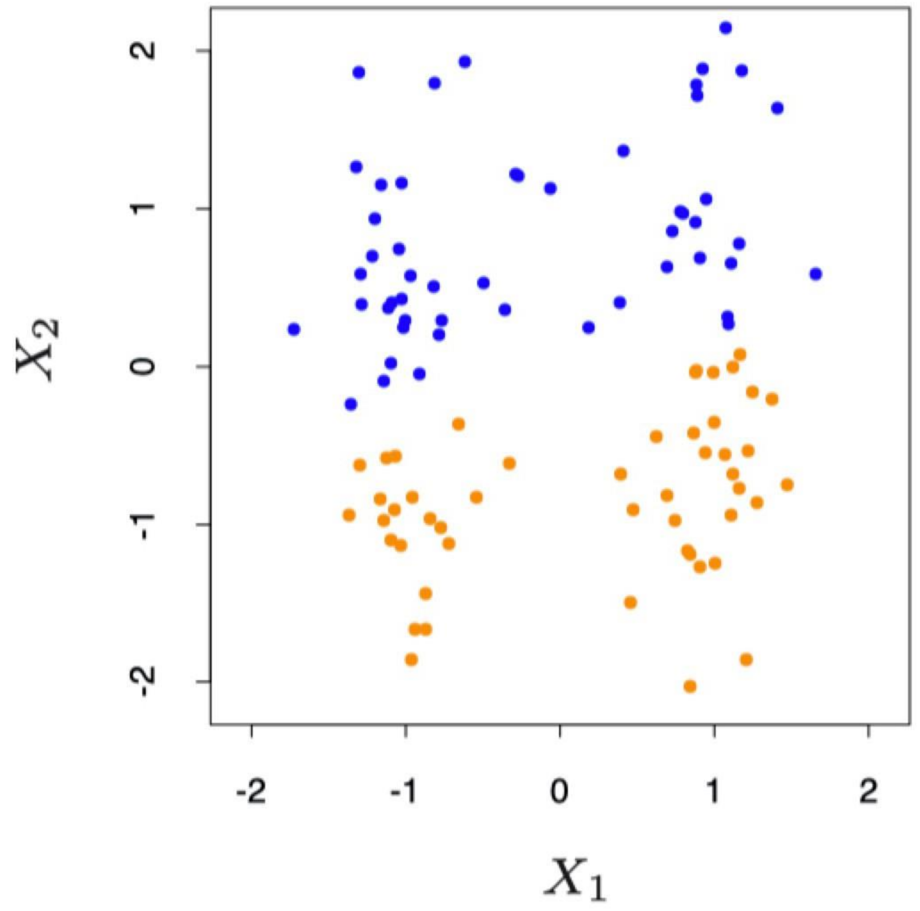
Data Preparation for Clustering

The points are colored by a clustering algorithm

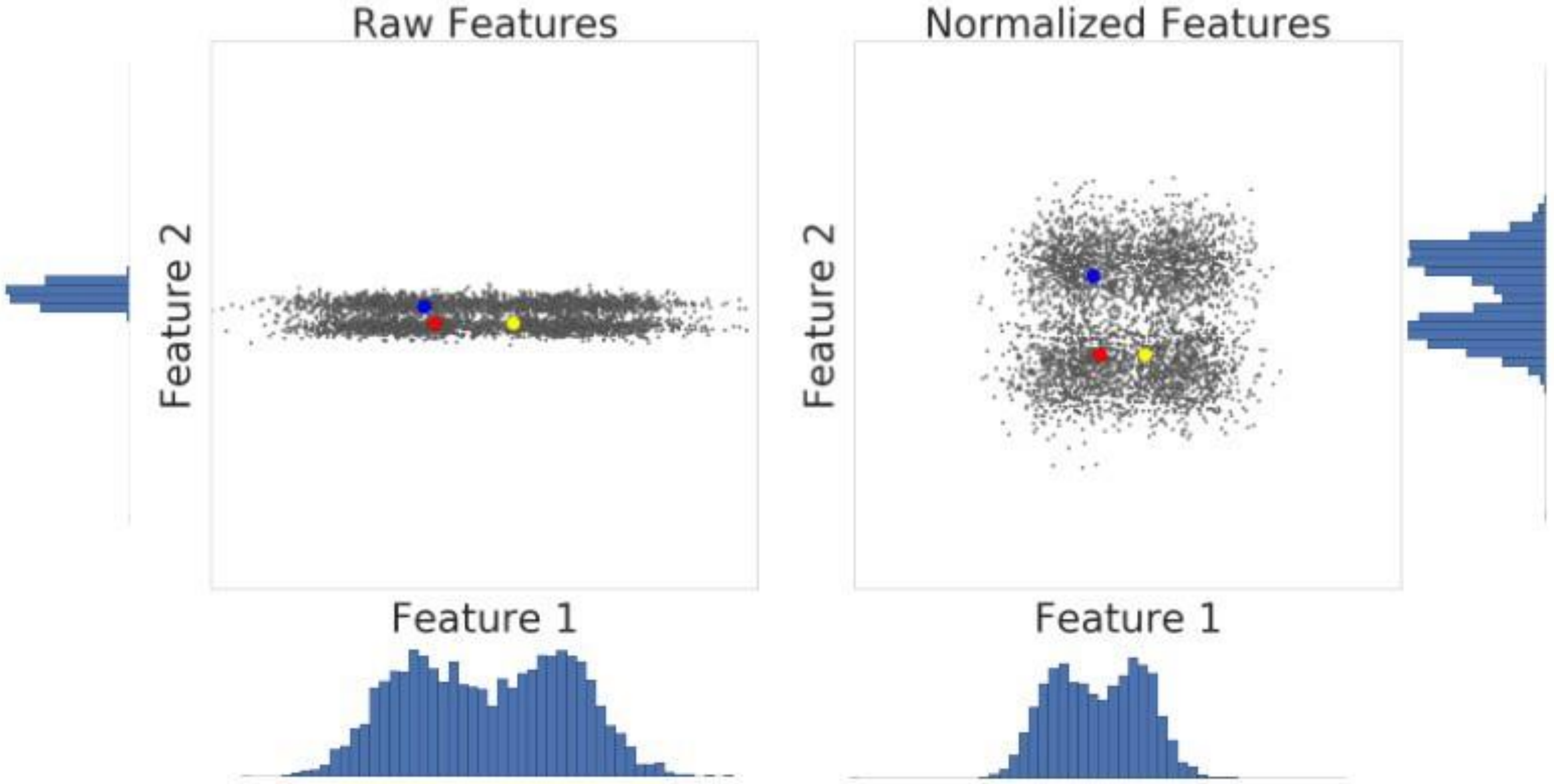
Raw data



Standardized data

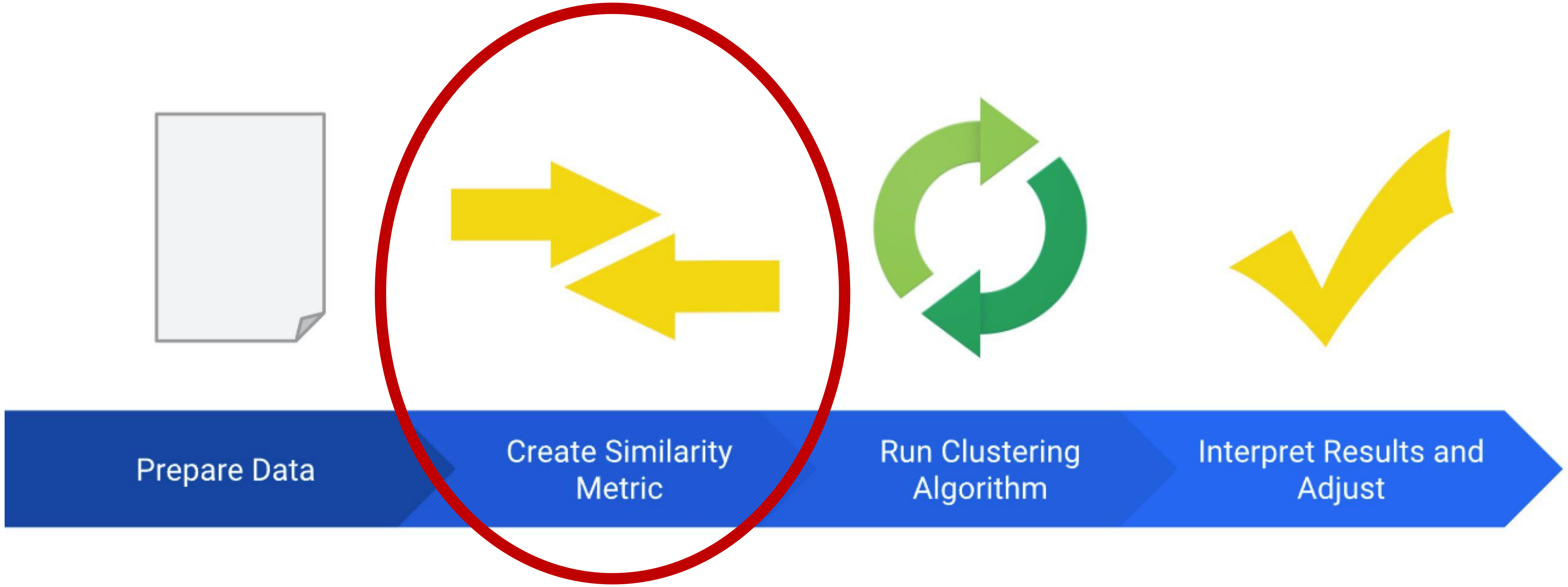


Data Preparation for Clustering



https://developers.google.com/machine-learning/clustering/prepare-data

Clustering workflow



From Google's Clustering lesson: https://developers.google.com/machine-learning/clustering/

Distance Metrics

The right distance metric depends on your application!

Distance of vectors $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$

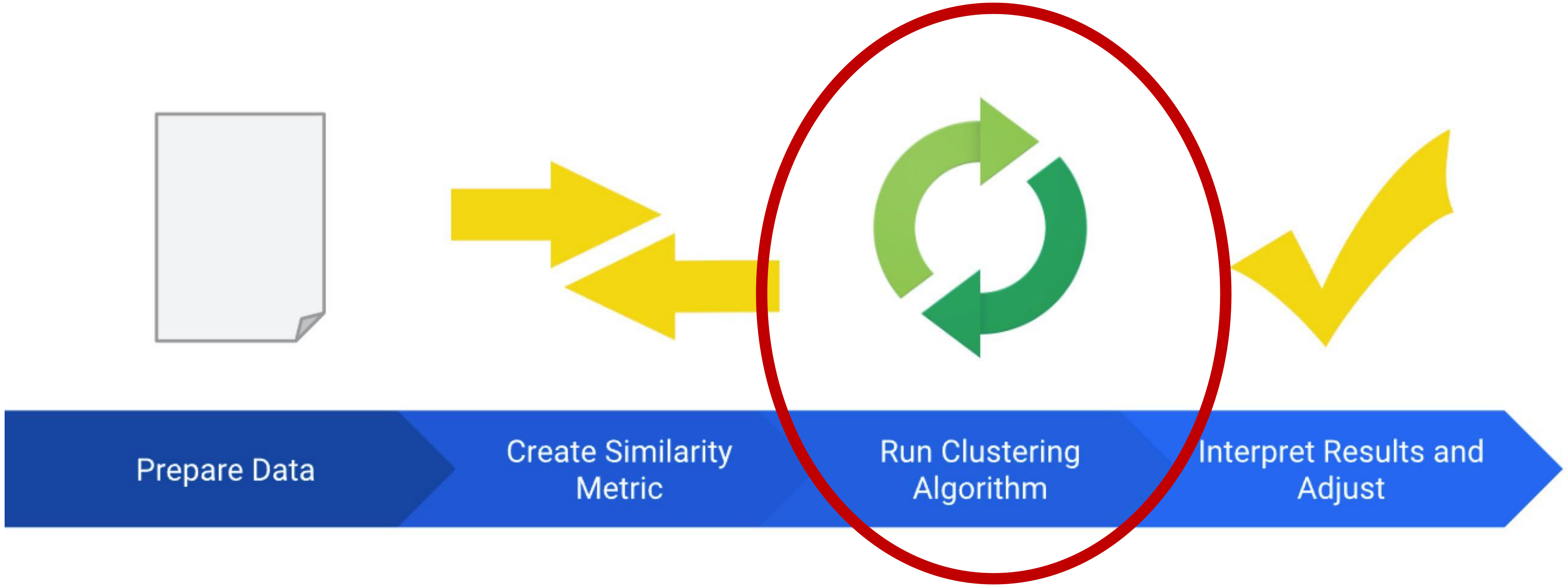
- Euclidean distance $d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
- Manhattan distance $d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$
- Correlation distance $d(x,y) = 1 - r(x,y)$ — $r(x,y)$ is Pearson correlation coefficient

Distance of sequences $\texttt{ACCTTG}$ and $\texttt{TACCTG}$

- Hamming distance $\begin{matrix} \underline{\texttt{AC}}\texttt{C}\underline{\texttt{T}}\texttt{TG} \\ \underline{\texttt{TA}}\texttt{C}\underline{\texttt{C}}\texttt{TG} \end{matrix}$ => 3
- Levenshtein distance $\begin{matrix} \underline{\texttt{.}}\texttt{ACC}\underline{\texttt{T}}\texttt{TG} \\ \underline{\texttt{T}}\texttt{ACC}\underline{\texttt{.}}\texttt{TG} \end{matrix}$ => 2

Based on slides by Elena Sügis: http://bioinformaticsinstitute.ru/sites/default/files/preprocessing_unsupervised.pdf

Clustering workflow

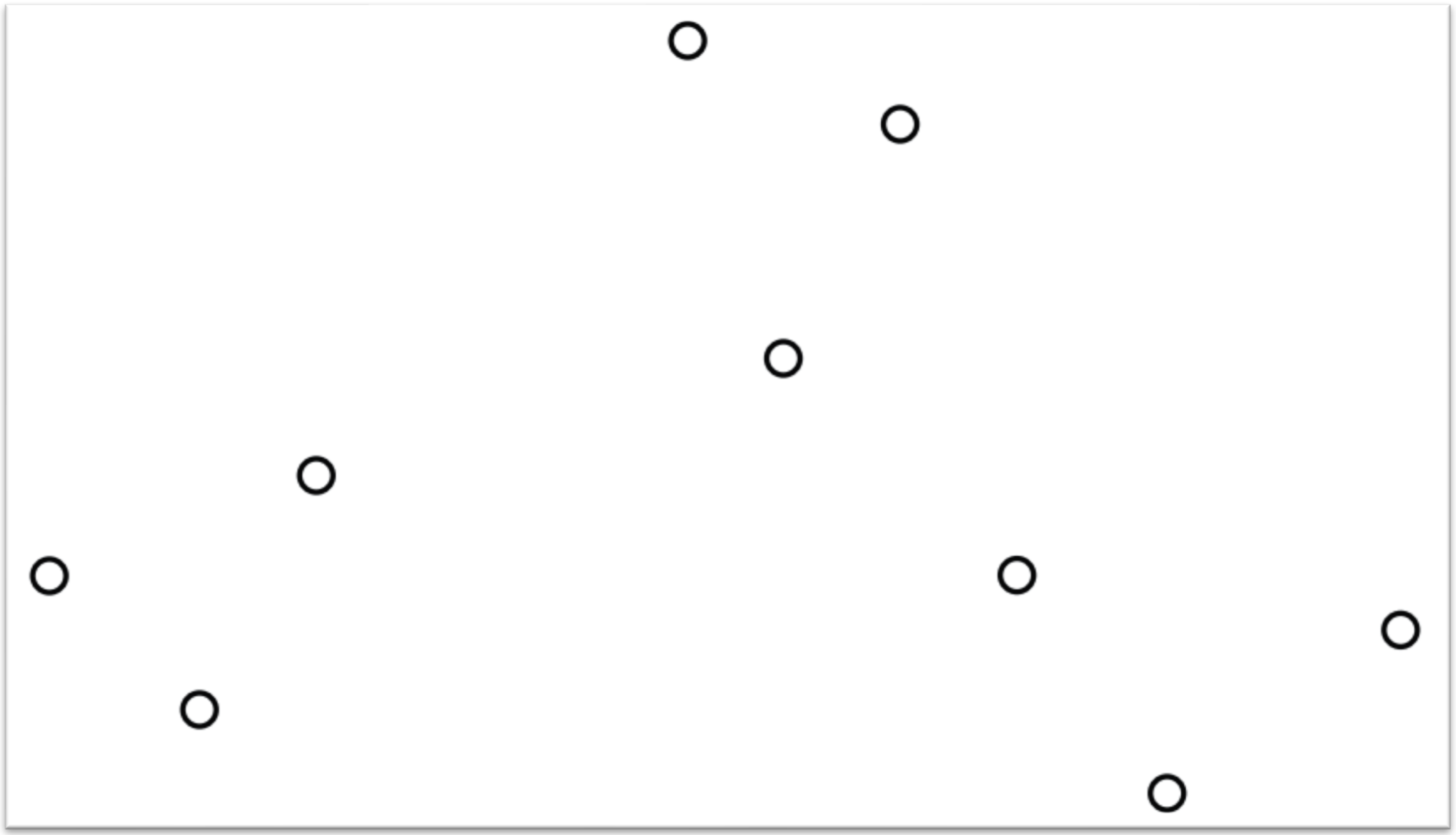


From Google's Clustering lesson: https://developers.google.com/machine-learning/clustering/

K-means Clustering

Euclidean distance metric

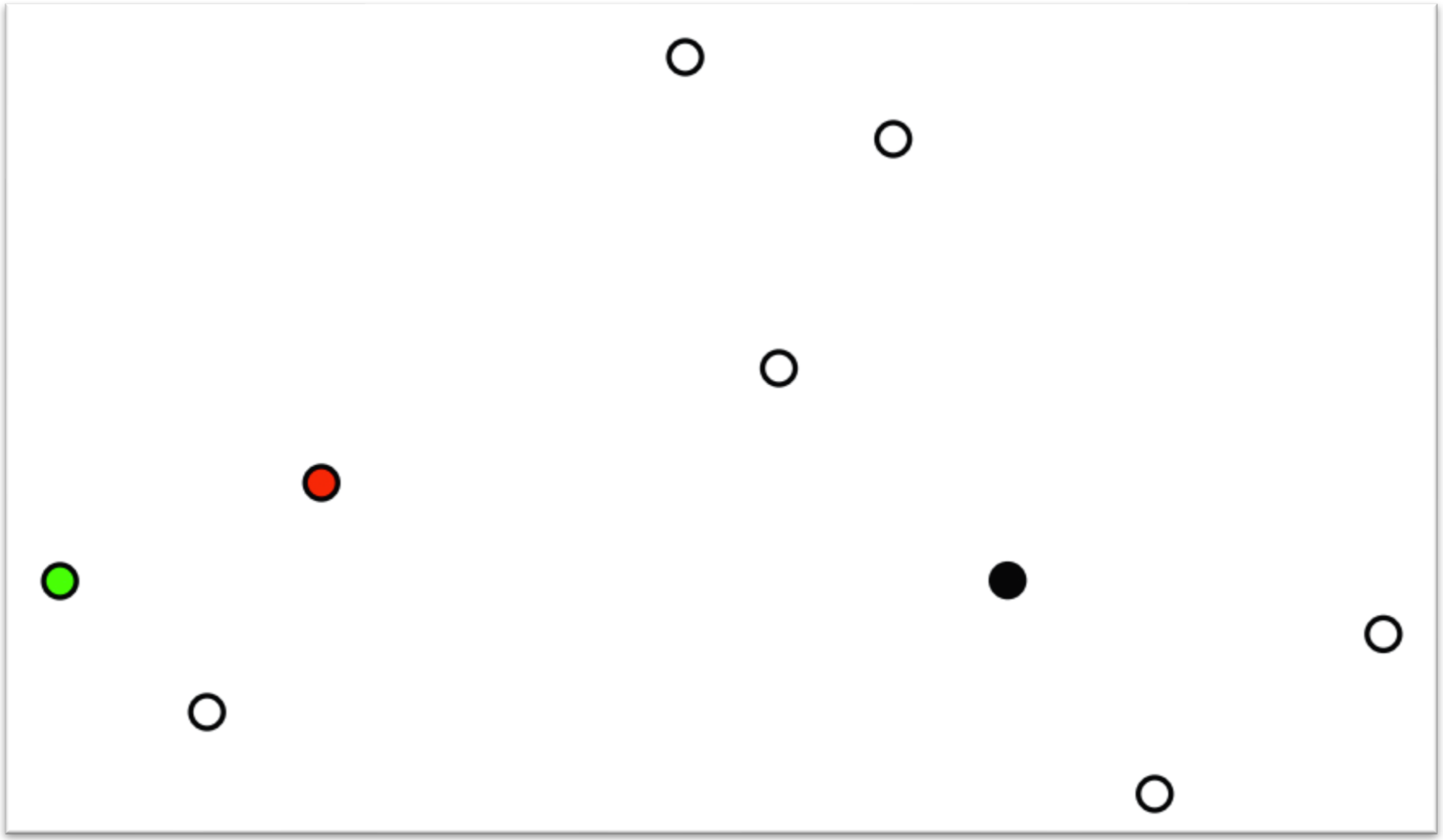
Given a set of data points…



K-means Clustering

Euclidean distance metric

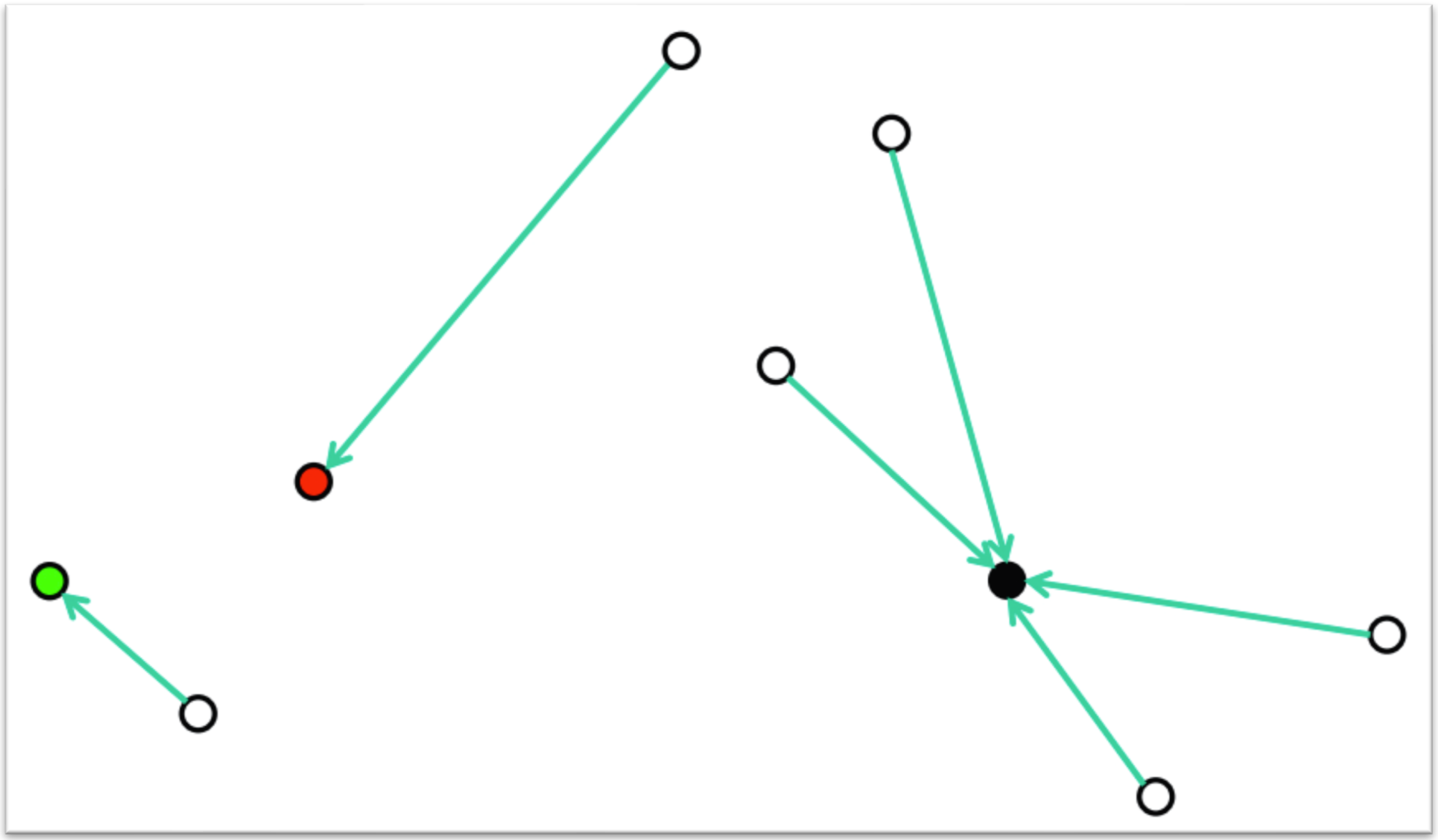
Select k=3 initial centers at random



K-means Clustering

Euclidean distance metric

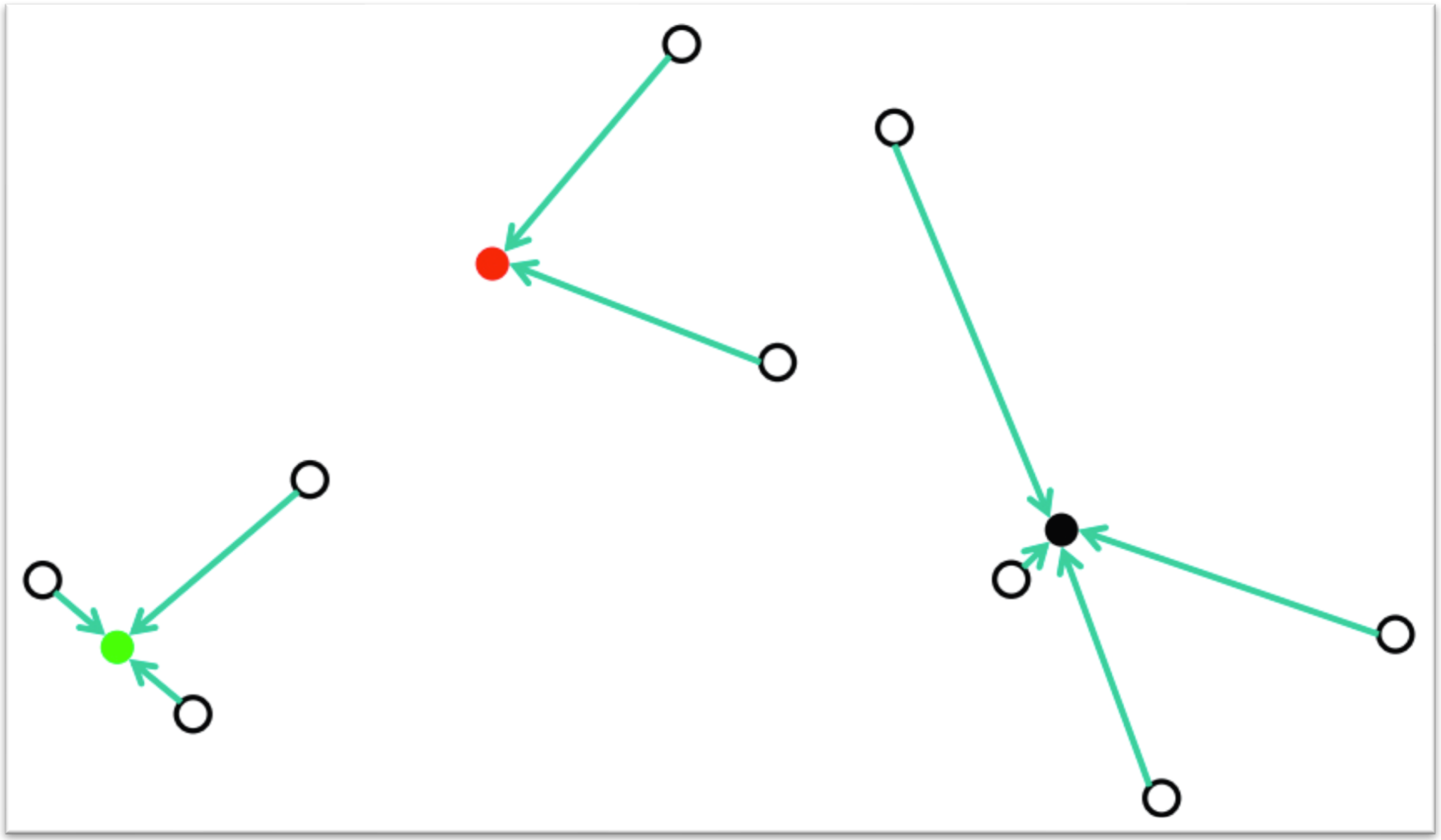
Recompute optimal centers given a fixed clustering



K-means Clustering

Euclidean distance metric

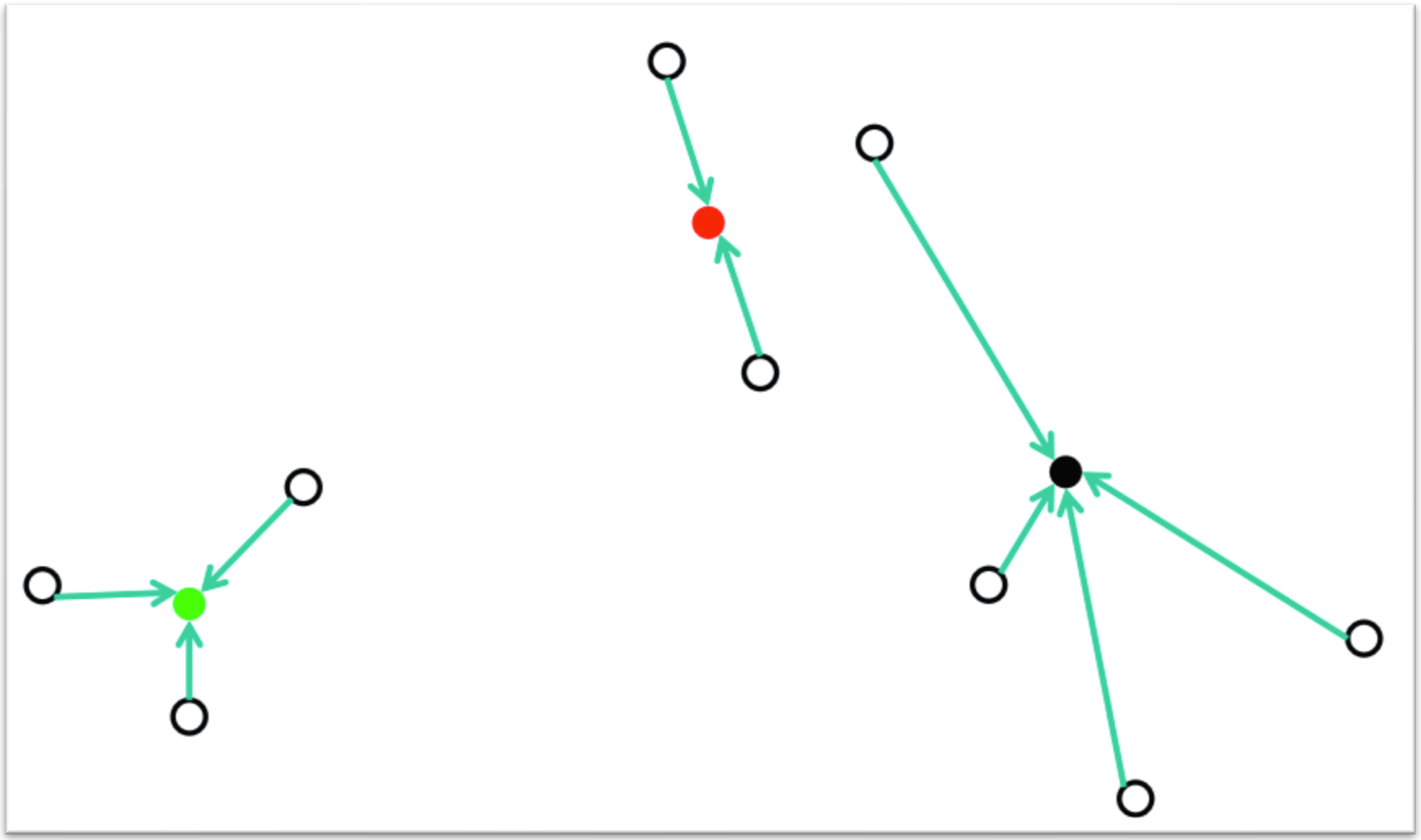
Assign each point to its nearest center



K-means Clustering

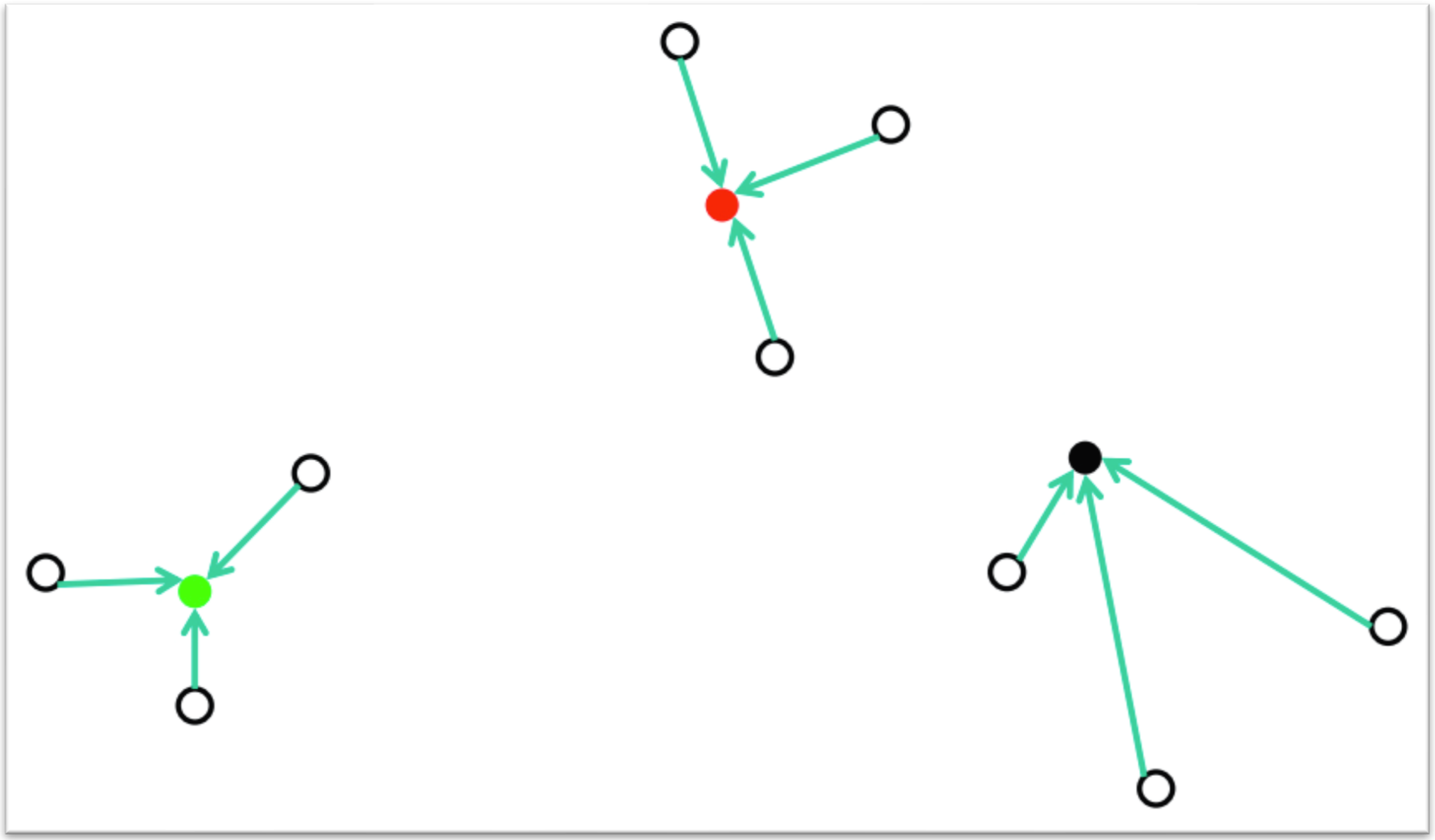
Euclidean distance metric

Recompute optimal centers given a fixed clustering



K-means Clustering

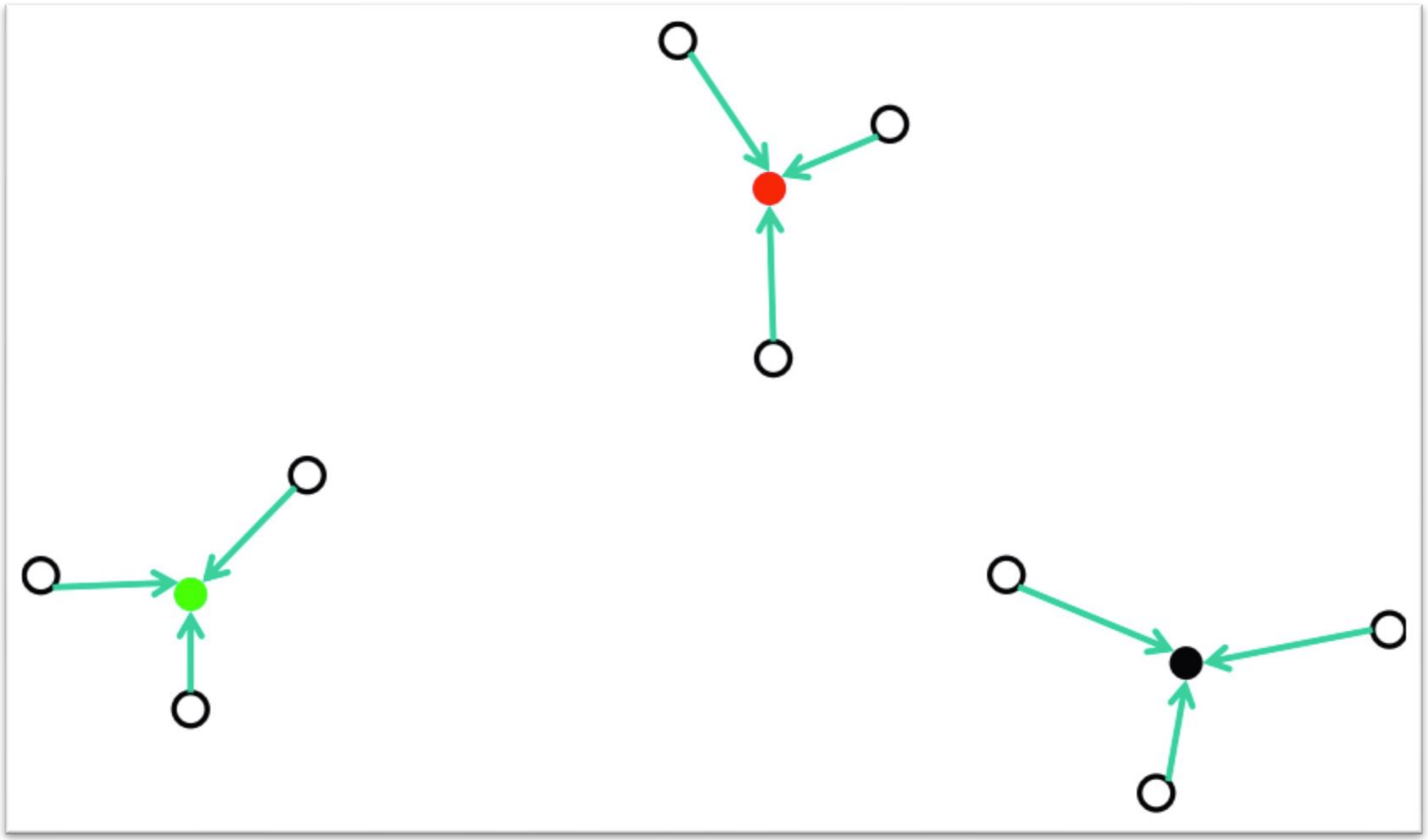
Assign each point to its nearest center



K-means Clustering

Euclidean distance metric

Recompute optimal centers given a fixed clustering



Selecting the number of clusters

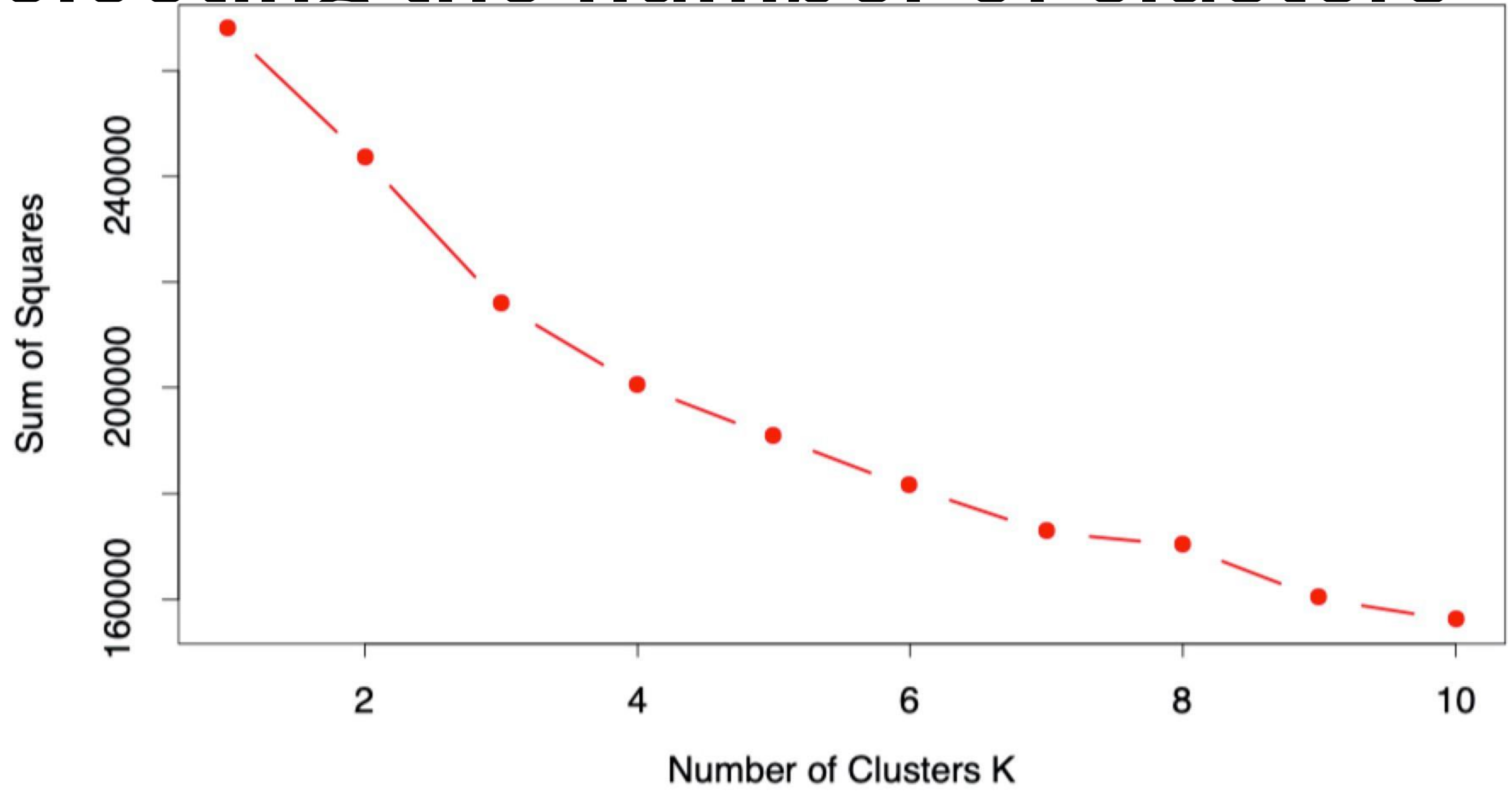
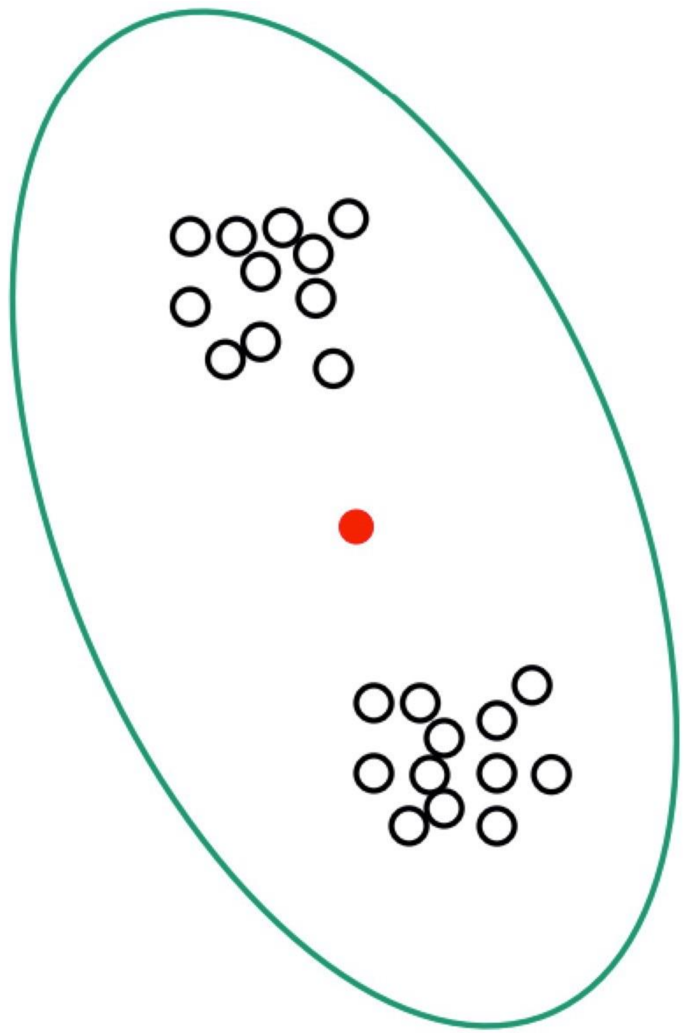
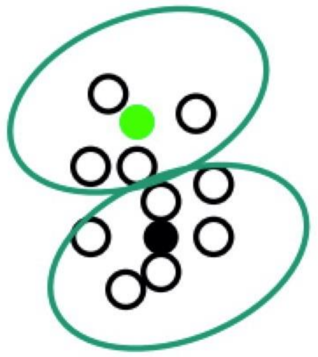


FIGURE 14.8. *Total within-cluster sum of squares for K-means clustering applied to the human tumor microarray data.*

K-means can fail

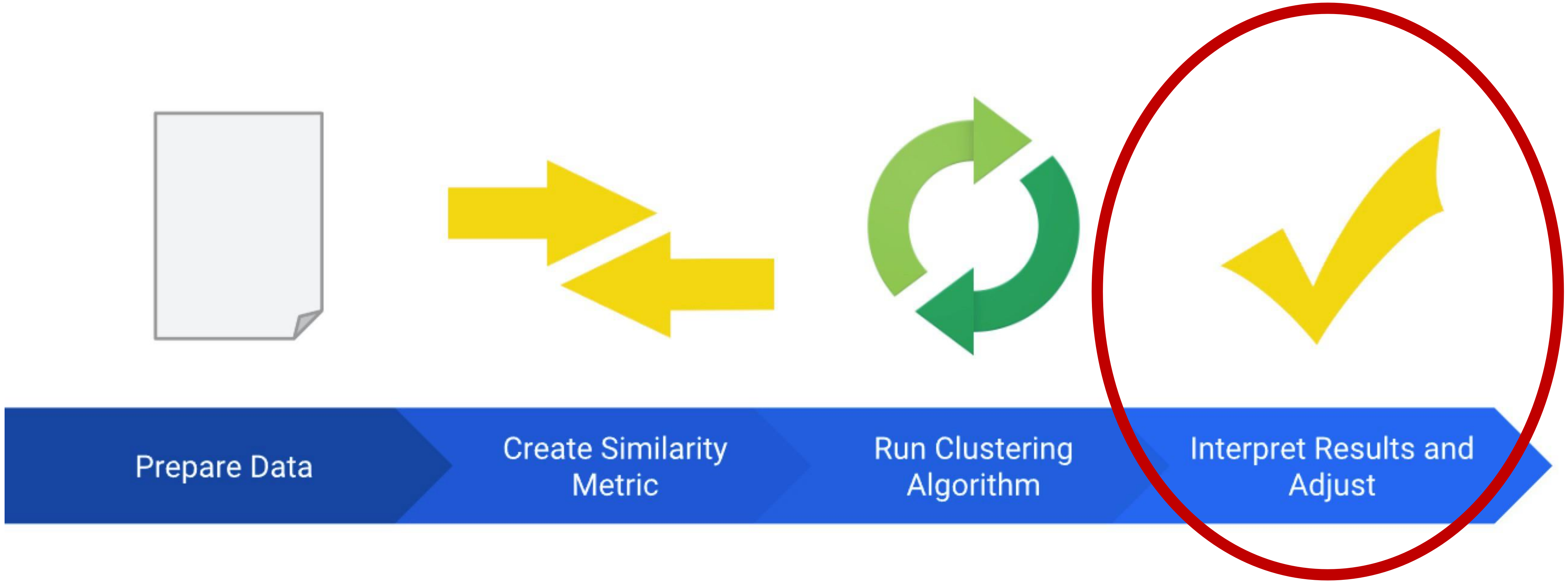


This is a heuristic!
No guarantees it'll find optimum



In practice, smarter centroid initialization solves this

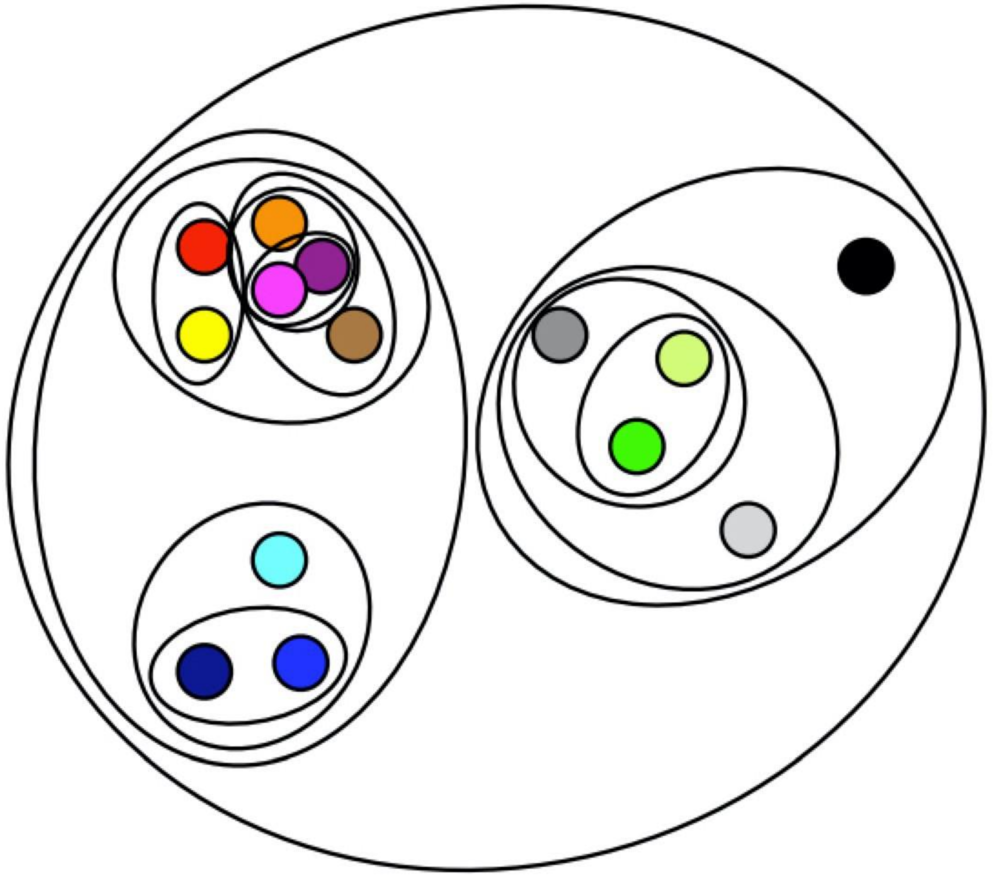
Clustering workflow

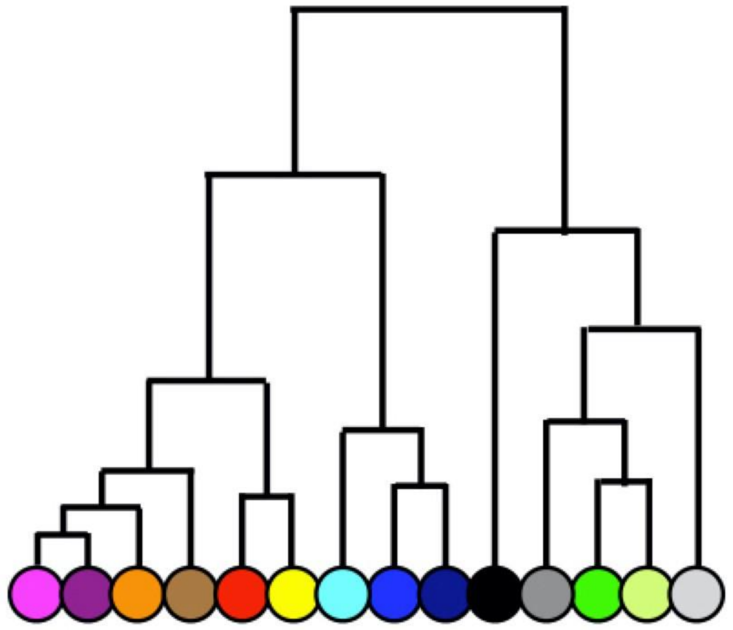


From Google's Clustering lesson: https://developers.google.com/machine-learning/clustering/

Hierarchical Clustering

High-level idea: build a tree (hierarchy) of clusters





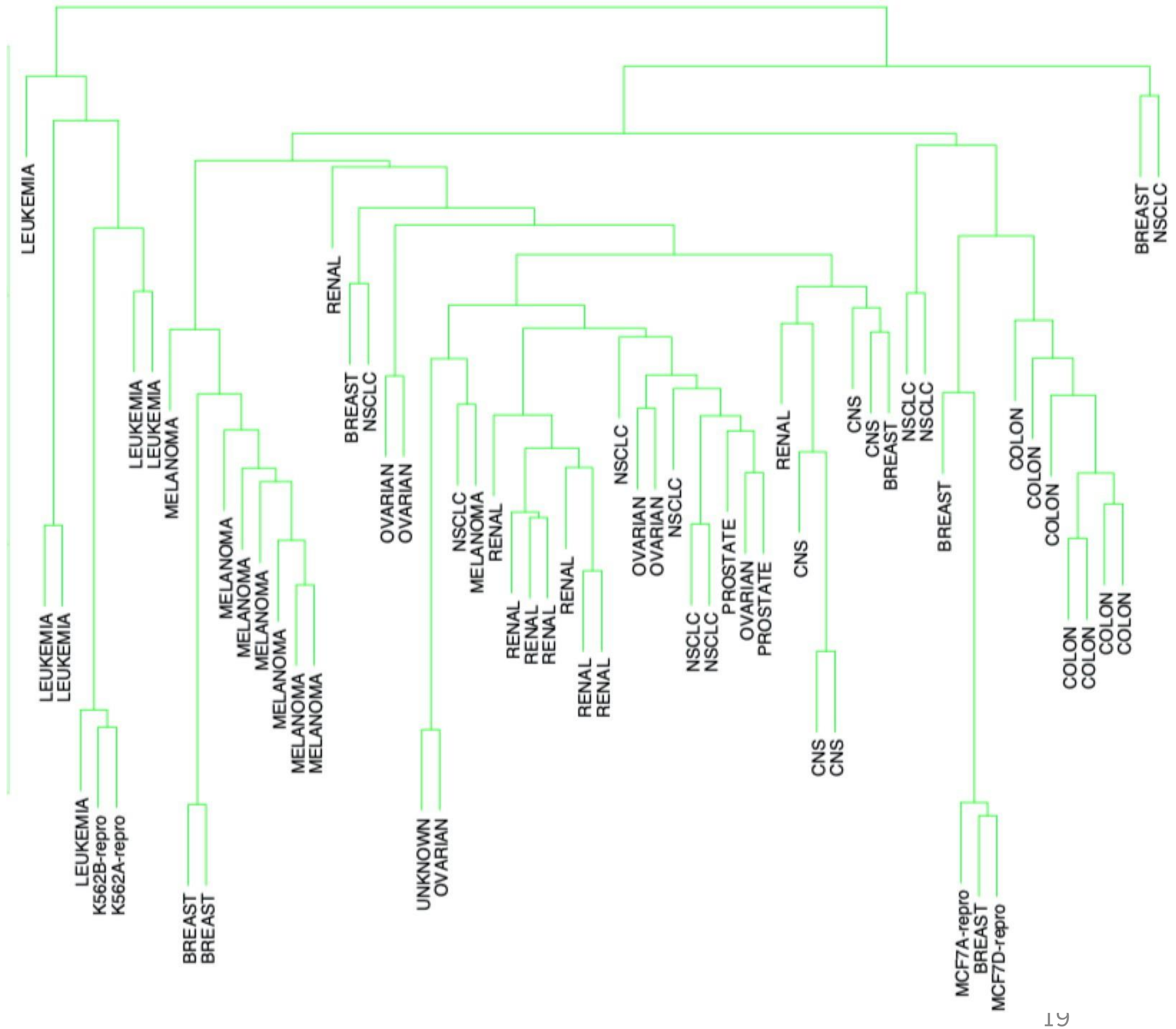
Based on slides by Elena Sügis: http://bioinformaticsinstitute.ru/sites/default/files/preprocessing_unsupervised.pdf

Hierarchical Clustering

Human Tumor Microarray Data

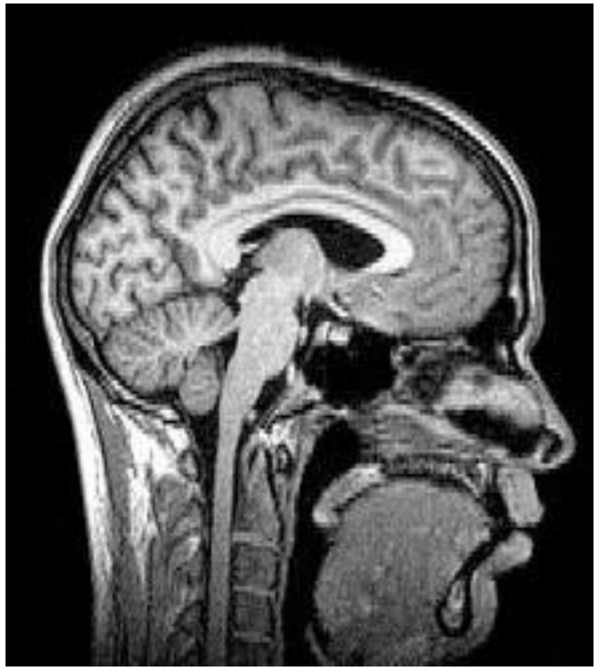
64 samples, 6830 features (gene expression levels)

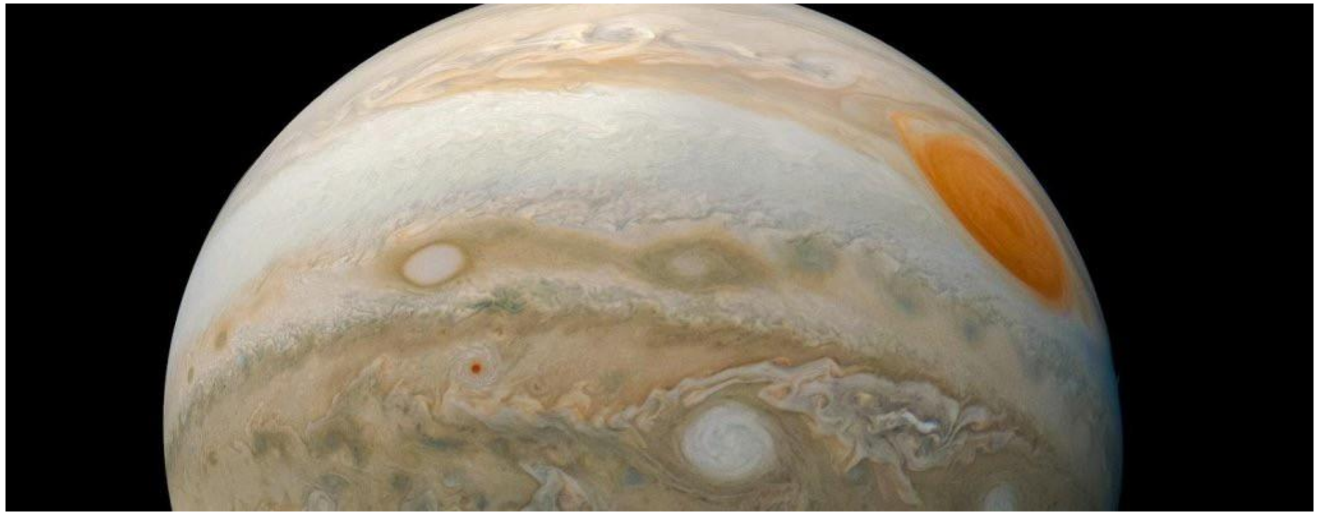
Elements of Statistical Learning, Chapter 14.3.8

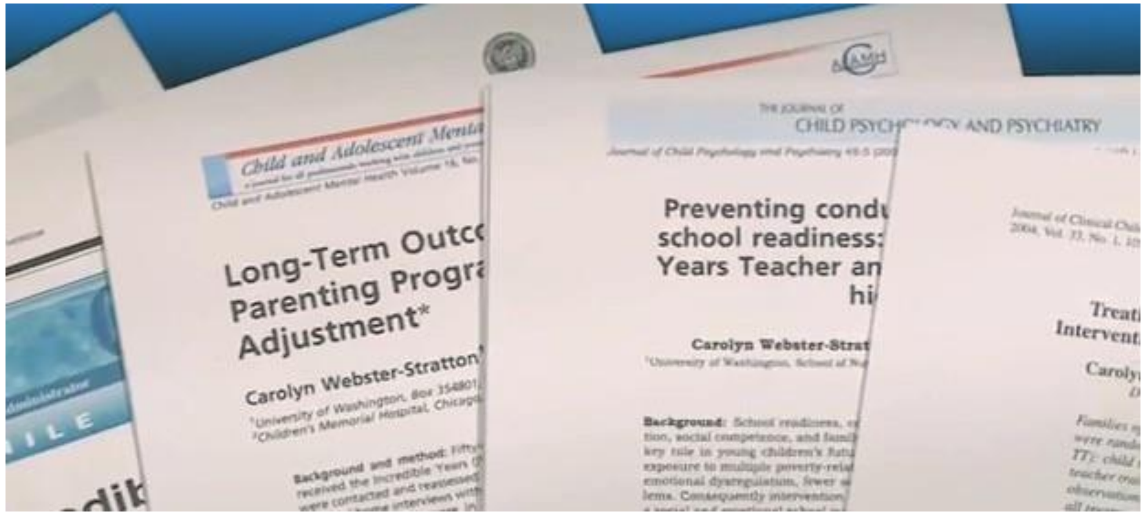


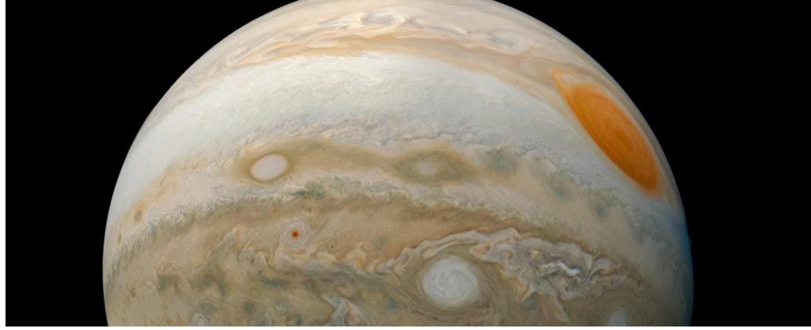
Dimensionality Reduction

or dealing with very high-dimensional data





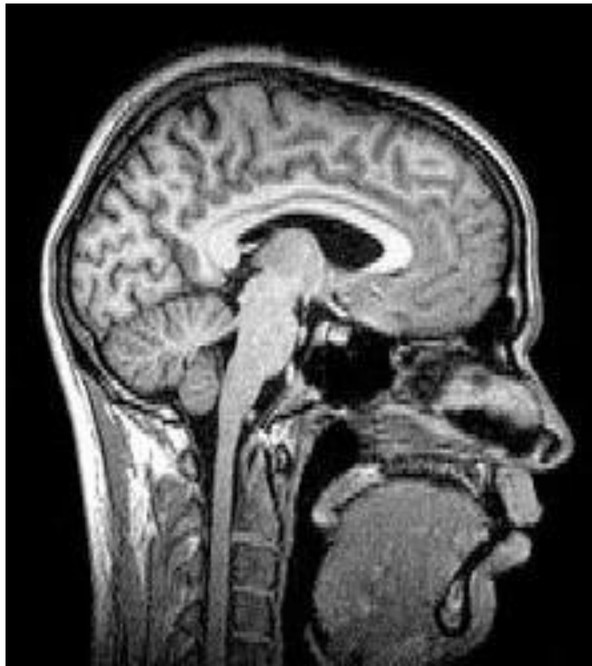




Dimensionality Reduction

Examples: Principal Component Analysis (PCA), t-SNE, …

Powerful unsupervised learning techniques for extracting hidden (potentially lower dimensional) structure from high dimensional datasets.



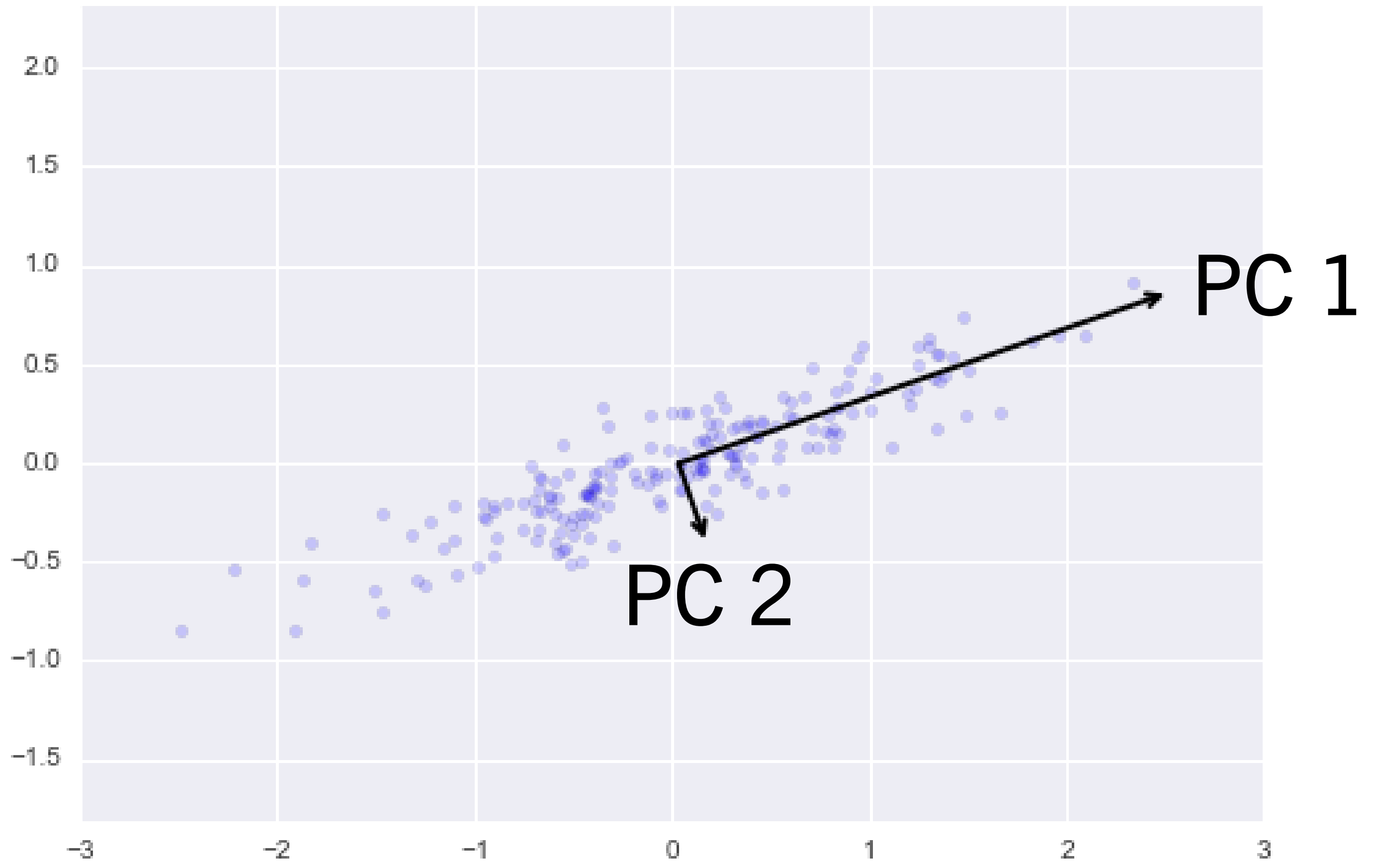
Useful for:

- Visualization
- Data compression for faster supervised learning
- Noise removal

Based on slide by Nina Balcan

Principal Component Analysis (PCA)

- Principal Components (PC) are orthogonal directions that capture most of the variance in the data.

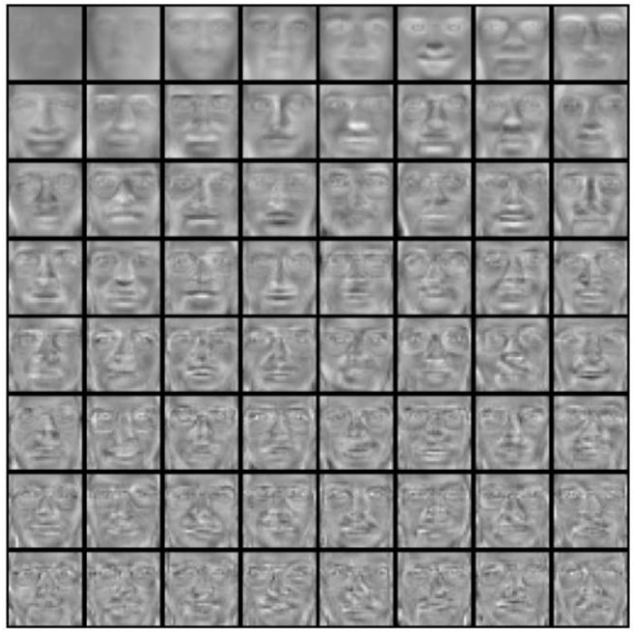


PCA for Face Reconstruction

Eigenfaces, slide based on Derek Hoeim's, UIUC CS543

Image dataset

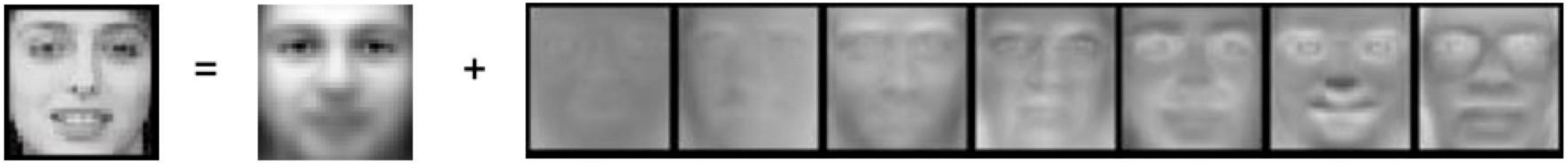
64 Principal Components



PCA for Face Reconstruction

Eigenfaces, slide based on Derek Hoeim's, UIUC CS543

Face Reconstruction using the Principal Components

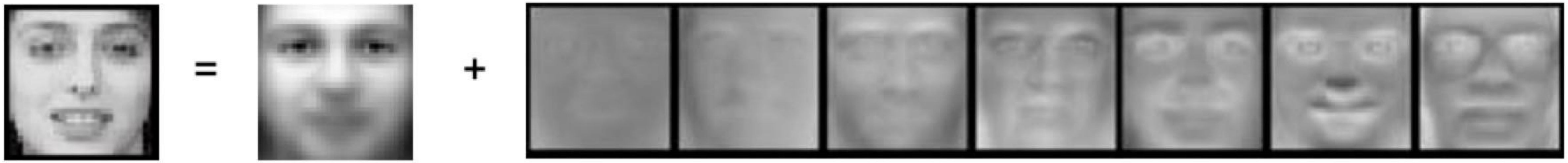


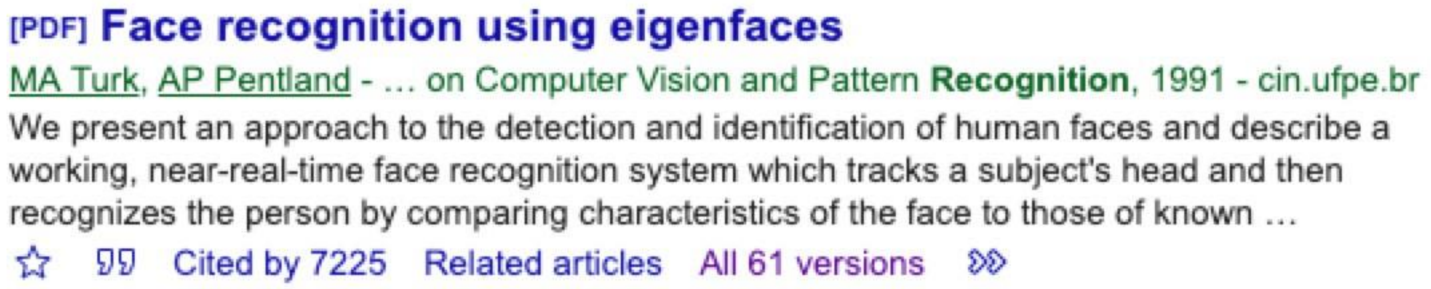
PCA for Face Reconstruction

Eigenfaces, slide based on Derek Hoeim's, UIUC CS543

Face Recognition using PCA:

1. Given face image datasets, extract Principal Components $v_1, v_2, \ldots, v_k$.
2. Given new image, project onto PCs.
3. Find closest (projected) image in training dataset




[PDF] **Face recognition using eigenfaces**

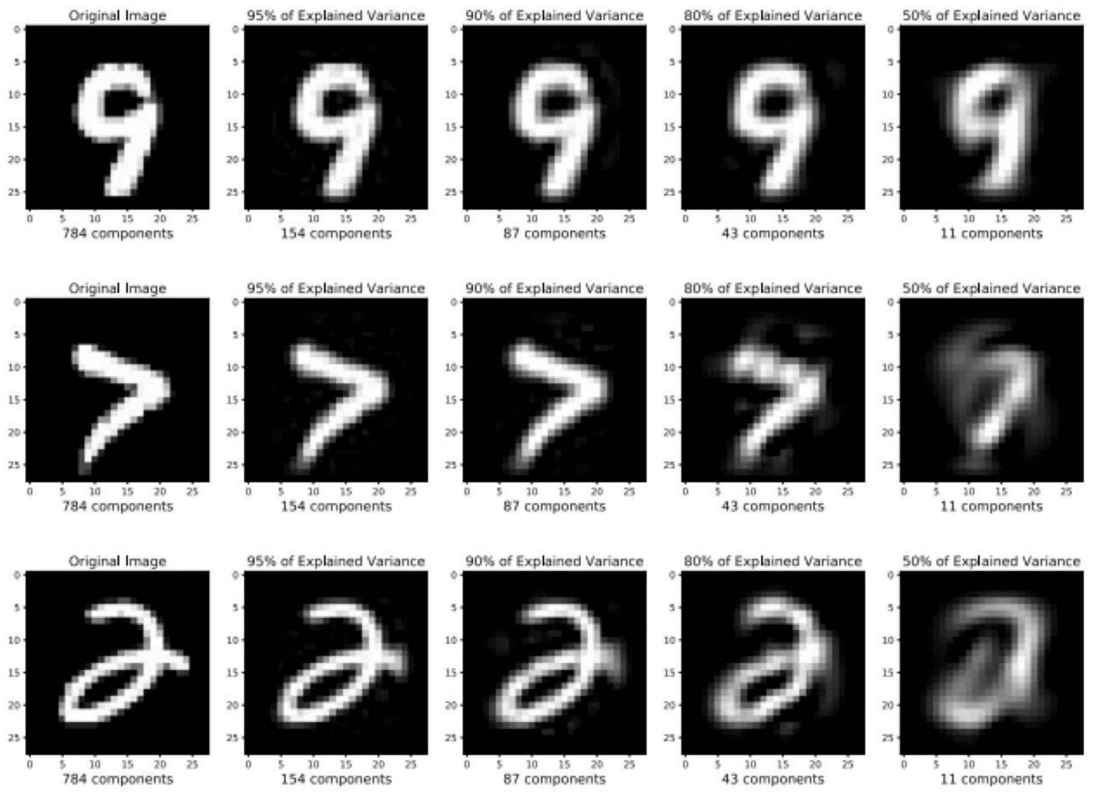
MA Turk, AP Pentland - … on Computer Vision and Pattern **Recognition**, 1991 - cin.ufpe.br

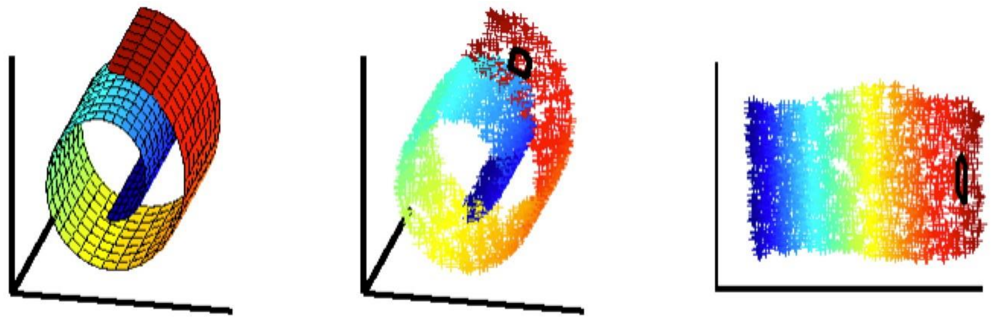
We present an approach to the detection and identification of human faces and describe a working, near-real-time face recognition system which tracks a subject's head and then recognizes the person by comparing characteristics of the face to those of known …

Cited by 7225 Related articles All 61 versions

Final words on PCA

- Advantages
  - Fast to compute an optimal solution: an eigenvector problem
  - No hyper-parameters to tune
- Caveats
  - Discards information
  - Limited to linear projections





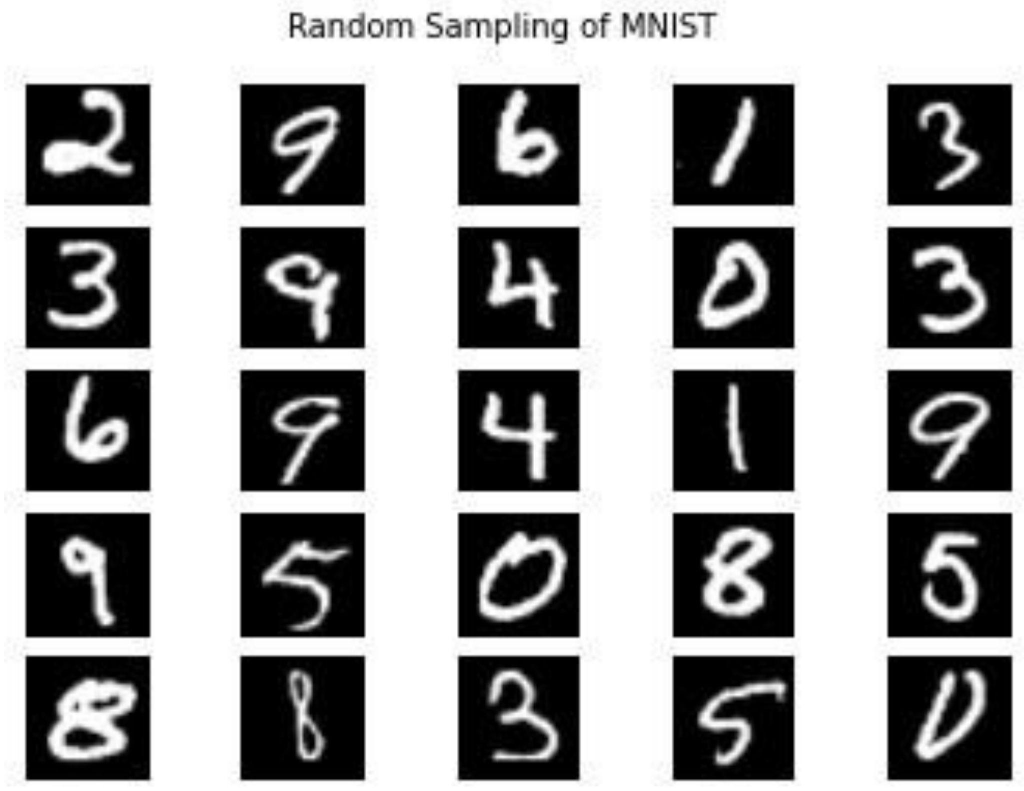
- From Michael Guerzhoy's slides, UofT CSC320

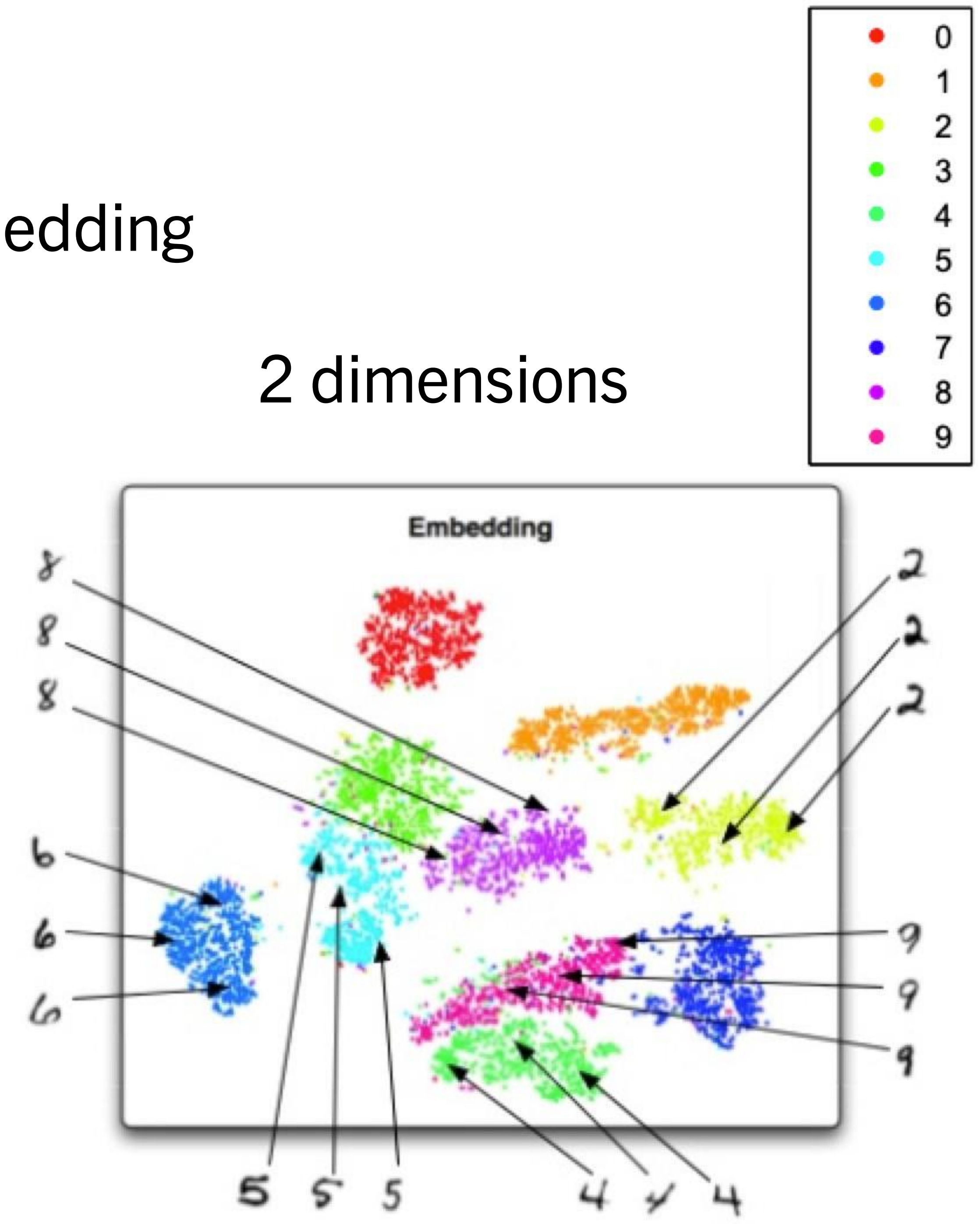
t-SNE

t-Distributed Stochastic Neighbor Embedding

784 dimensions = 28x28 pixels

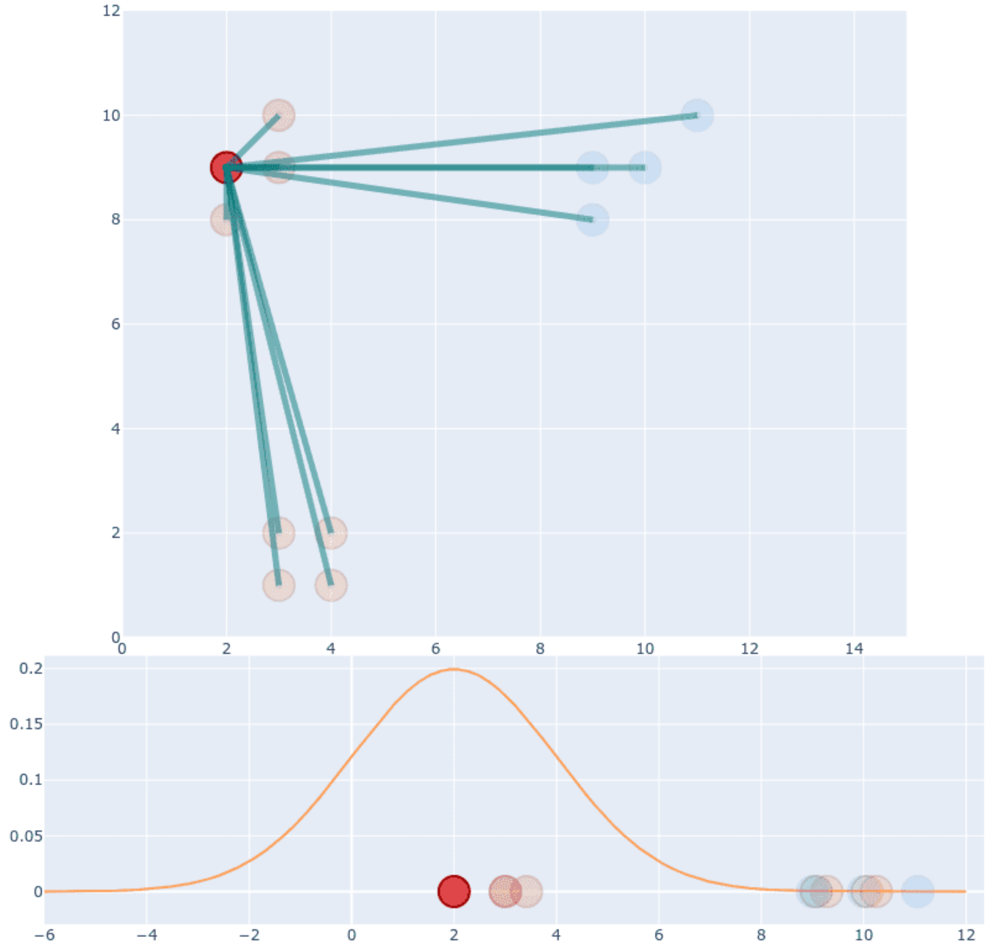
2 dimensions





t-SNE

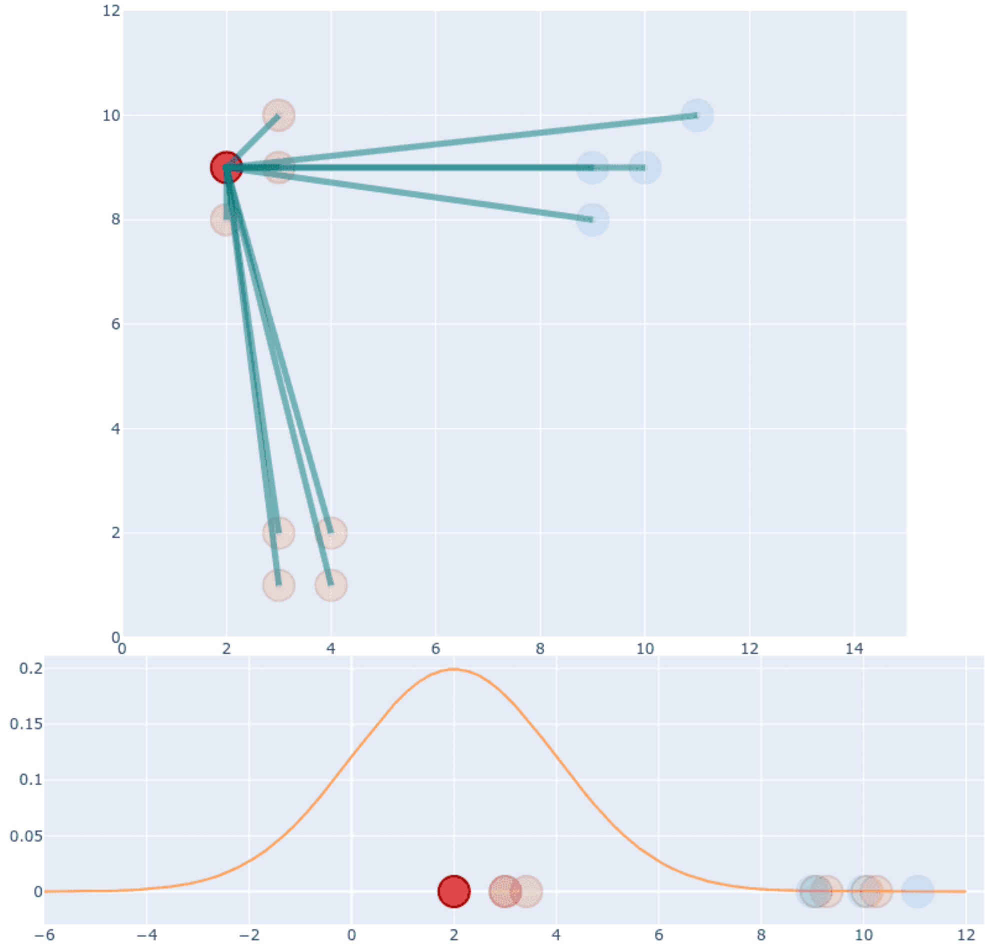
- Create a distribution representing the similarity of points
- Similarity of A to B is the conditional probability $p_{A|B}$ that B would pick A as its neighbour



https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

t-SNE

- Create a matching distribution in the low dimensional space, such that the divergence between $p_{A|B}$ and $q_{A|B}$ is minimized
- Key hyperparameter: width of the gaussian is controlled by *perplexity*



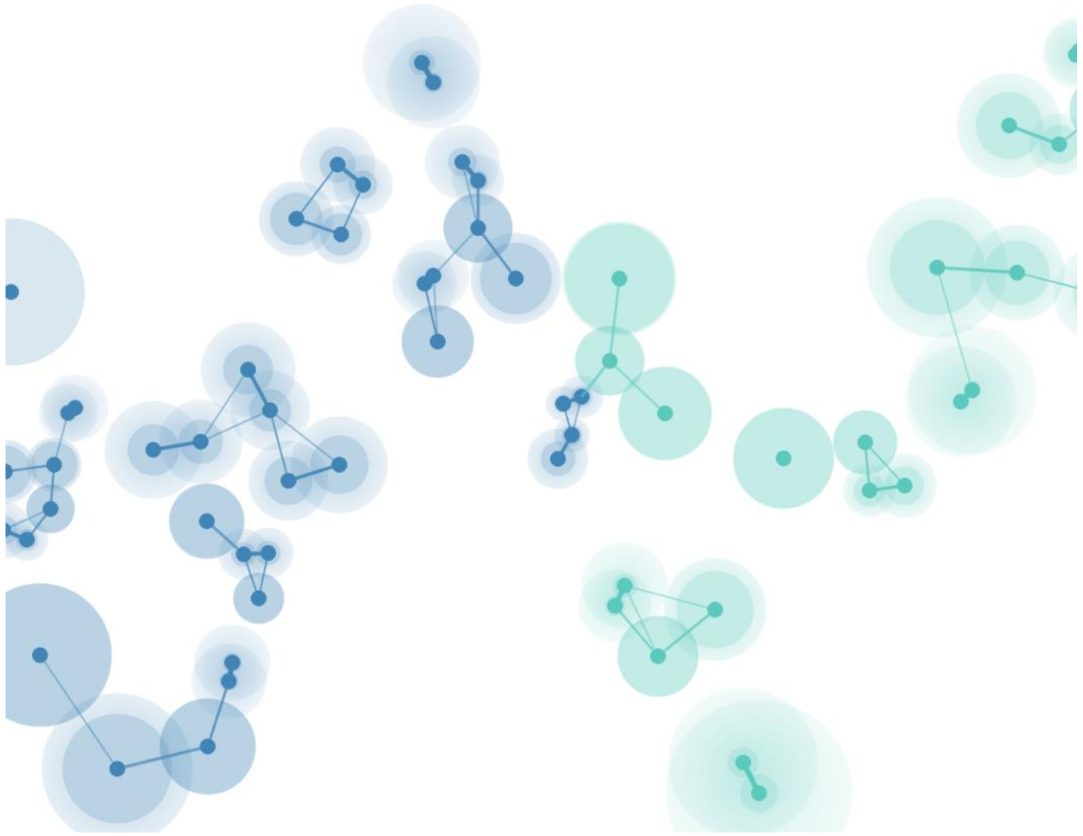
https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

t-SNE

- Last step: while we use a **Gaussian** in the original space, we use a **Student's t-distribution** in the lower dimensional space
- Intuition: because we are losing dimensions, we need more distances that map to approximately the same probability
- Student's t-distribution has a longer tail than a Gaussian, allowing for more "options" in how the result is organized

UMAP

- Both UMAP and t-SNE try to find a low-dimensional representation that matches the structure of the high-dimensional one
- But while t-SNE looks at pairwise distances between points, UMAP looks at a network of neighbours



More info

- https://pair-code.github.io/understanding-umap/