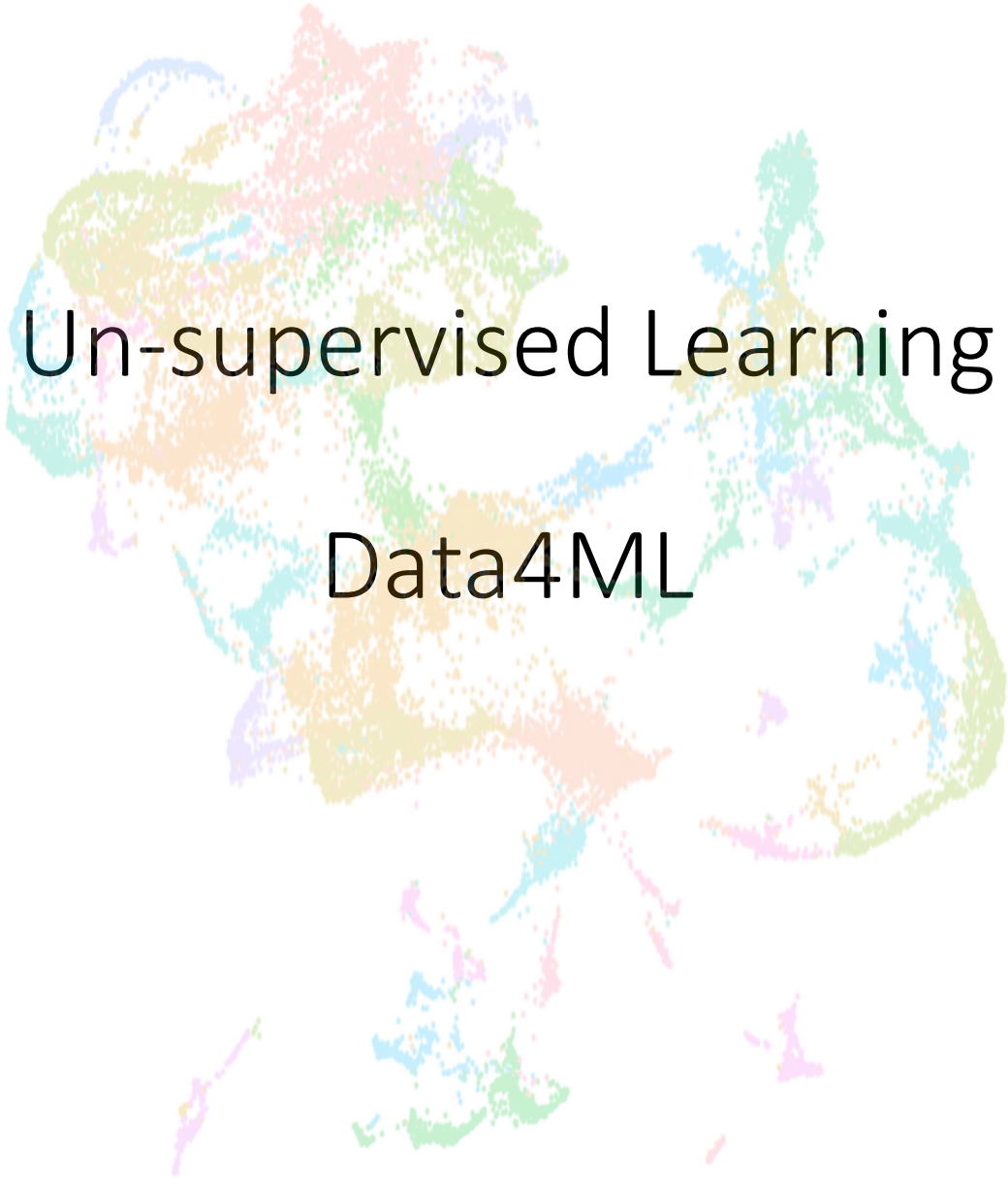
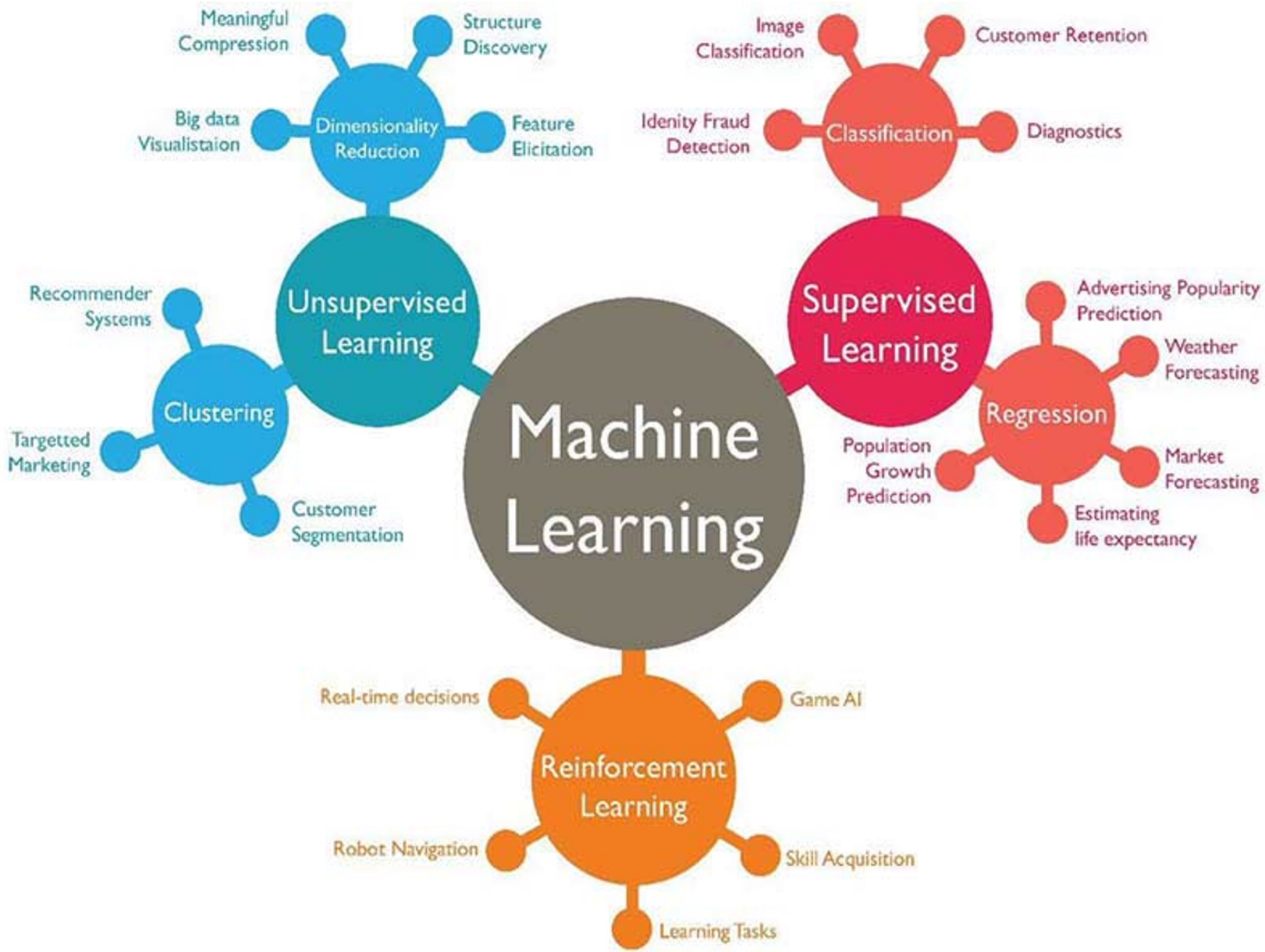


Un-supervised Learning

Data4ML


Machine Learning
Unsupervised Learning
Supervised Learning
Reinforcement Learning
Dimensionality Reduction
Clustering
Classification
Regression
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Recommender Systems
Targetted Marketing
Customer Segmentation
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

Introduction

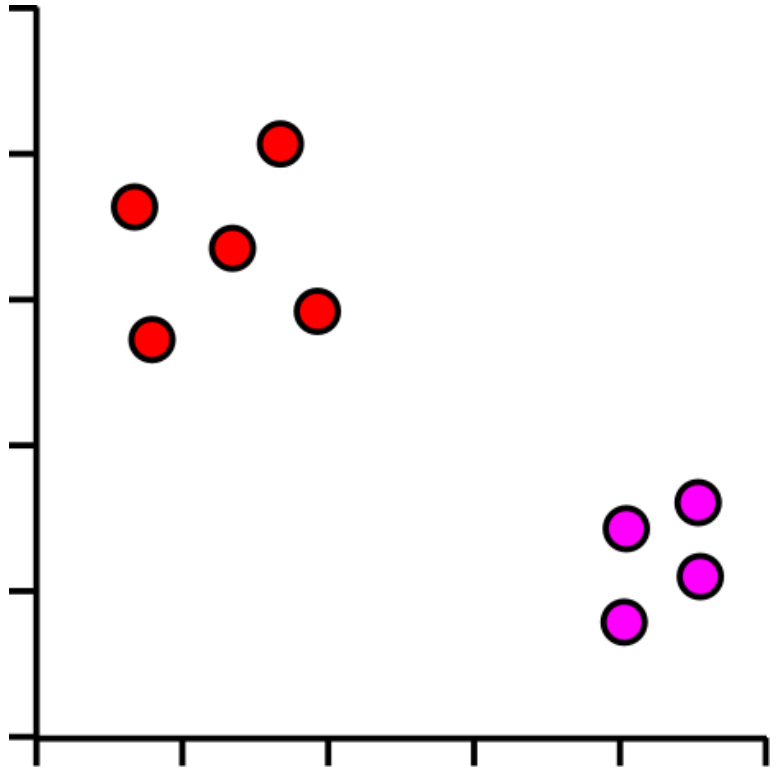
- Supervised learning: discover patterns in data with a known target (class) or label.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.

- Unsupervised learning: The data has no target attribute.
  - We want to explore the data to find some intrinsic structures in them.

Dimensionality Reduction

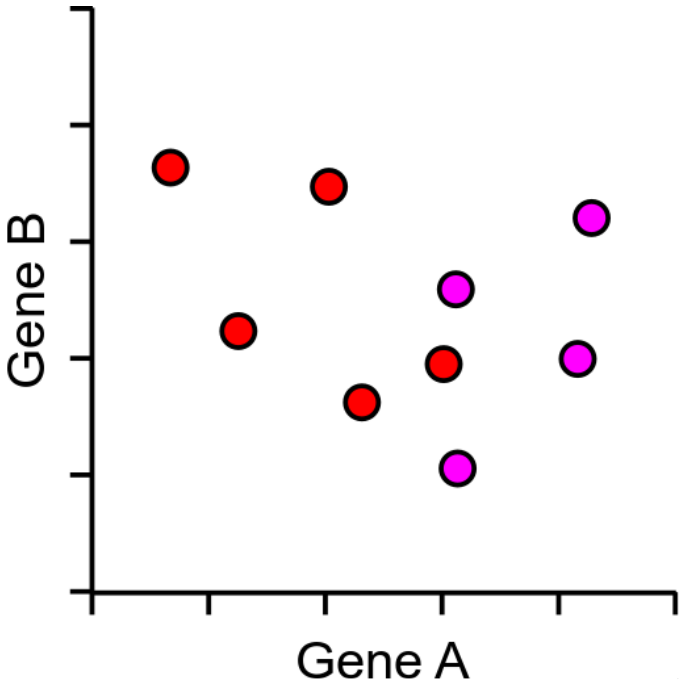
Gene	Description	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5
Inpp5d	inositol polyphosphate-5-phosphatase D	7.00	5.45	5.89	6.03	5.75
Aim2	absent in melanoma 2	3.01	4.37	4.59	4.38	4.18
Gldn	gliomedin	3.48	3.63	4.61	4.70	4.74
Frem2	Fras1 related extracellular matrix protein 2	4.75	4.66	3.46	3.74	3.45
Rps3a1	ribosomal protein S3A1	6.10	7.23	7.44	7.36	7.34
Slc38a3	solute carrier family 38, member 3	1.90	3.16	3.52	3.61	3.19
Mt1	metallothionein 1	5.07	6.49	6.46	6.04	6.05
C1s1	complement component 1, s subcomponent 1	2.74	3.02	3.86	4.10	4.10
Cds1	CDP-diacylglycerol synthase 1	4.55	4.22	3.80	3.16	3.12
Ifi44	interferon-induced protein 44	4.82	4.52	3.87	3.42	3.59
Lefty2	left-right determination factor 2	6.95	6.28	5.88	5.60	5.61
Fmr1nb	fragile X mental retardation 1 neighbor	4.28	2.78	3.10	3.25	2.57
Tagln	transgelin	7.93	7.91	7.20	7.02	6.68

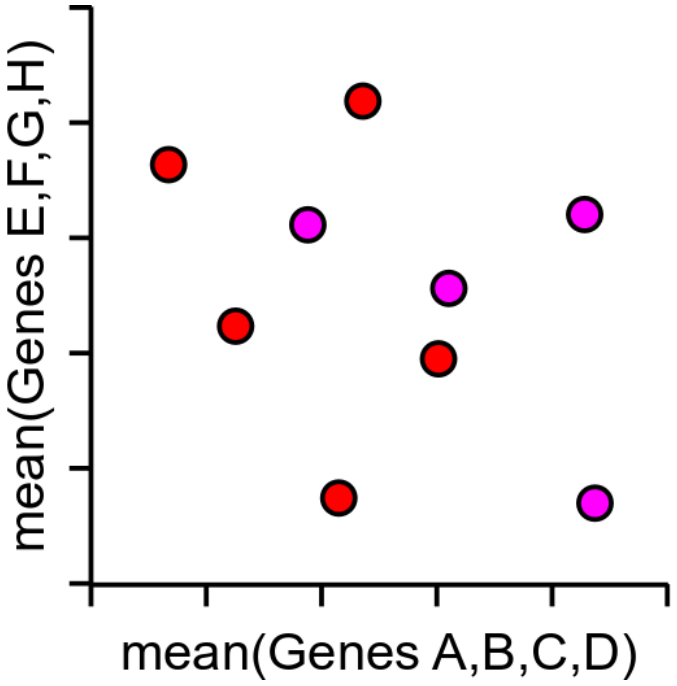
- What you have is a lot of features
- What you want is a few features you can look at
- Keep only the important information
- What counts as important?

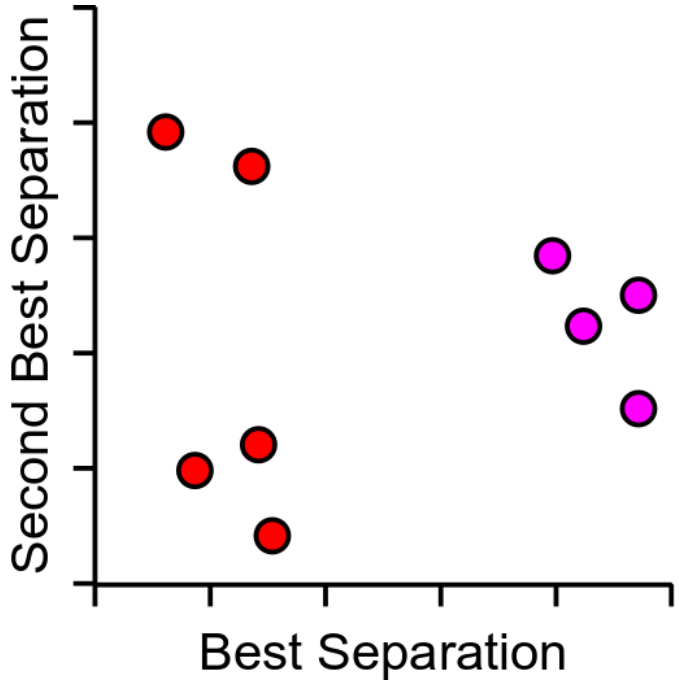


Too Much Data

- 5000 cells and 2500 measured genes
- Realistically only 2 dimensions we can plot (x,y)







Principal Components Analysis (PCA)

- Method to optimally summarise large multi-dimensional datasets
- Can find a smaller number of dimensions (ideally 2) which retain most of the useful information in the data
- Builds a **Linear** recipe for converting large amounts of data into new more useful features, called a Principle Component (PC), eg:

PC = (GeneA*10)+(GeneB*3)+(GeneC*-4)+(GeneD*-20)…

Principal Components Analysis (PCA)

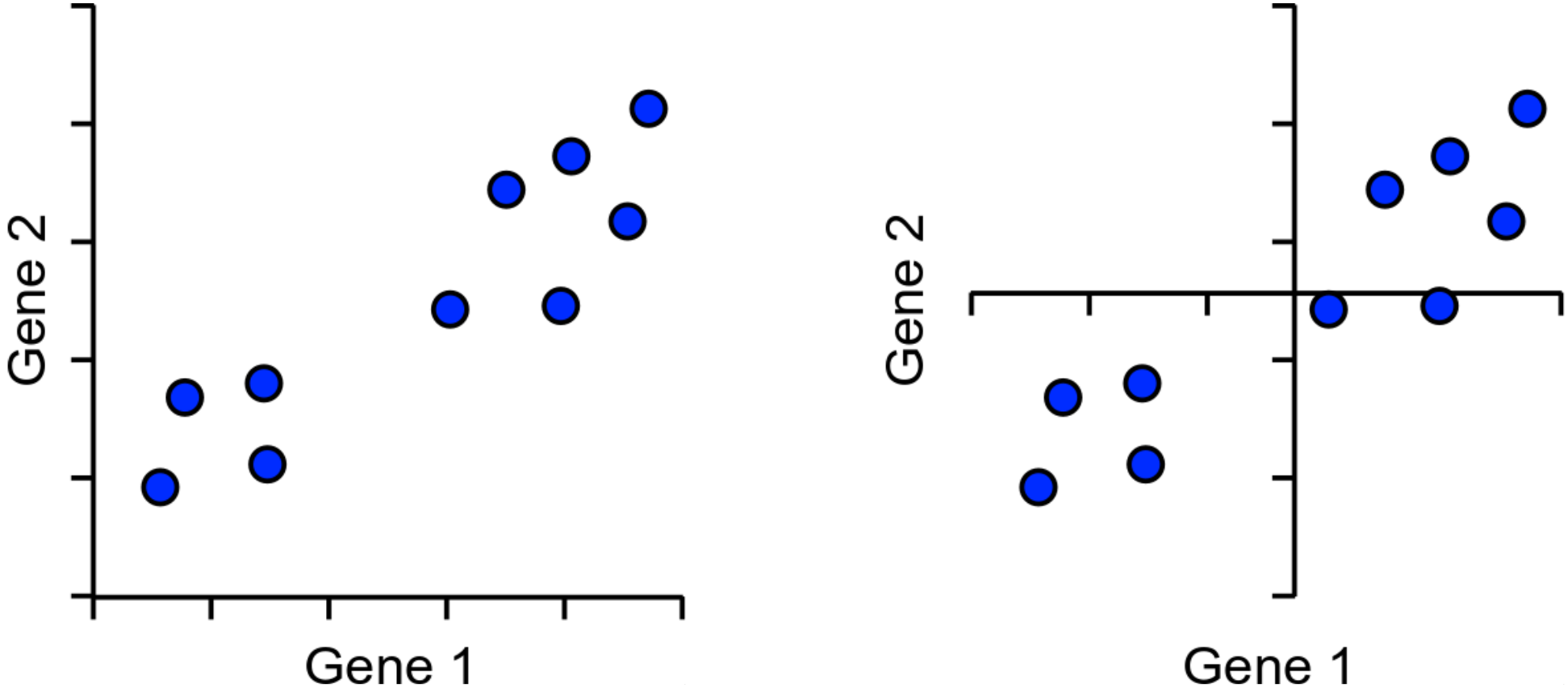
- Method to optimally summarise large multi-dimensional datasets
- Can find a smaller number of dimensions (ideally 2) which retain most of the useful information in the data

- Builds a recipe for converting large amounts of data into a single value, called a Principle Component (PC), eg:

PC = (GeneA*10)+(GeneB*3)+(GeneC*-4)+(GeneD*-20)…

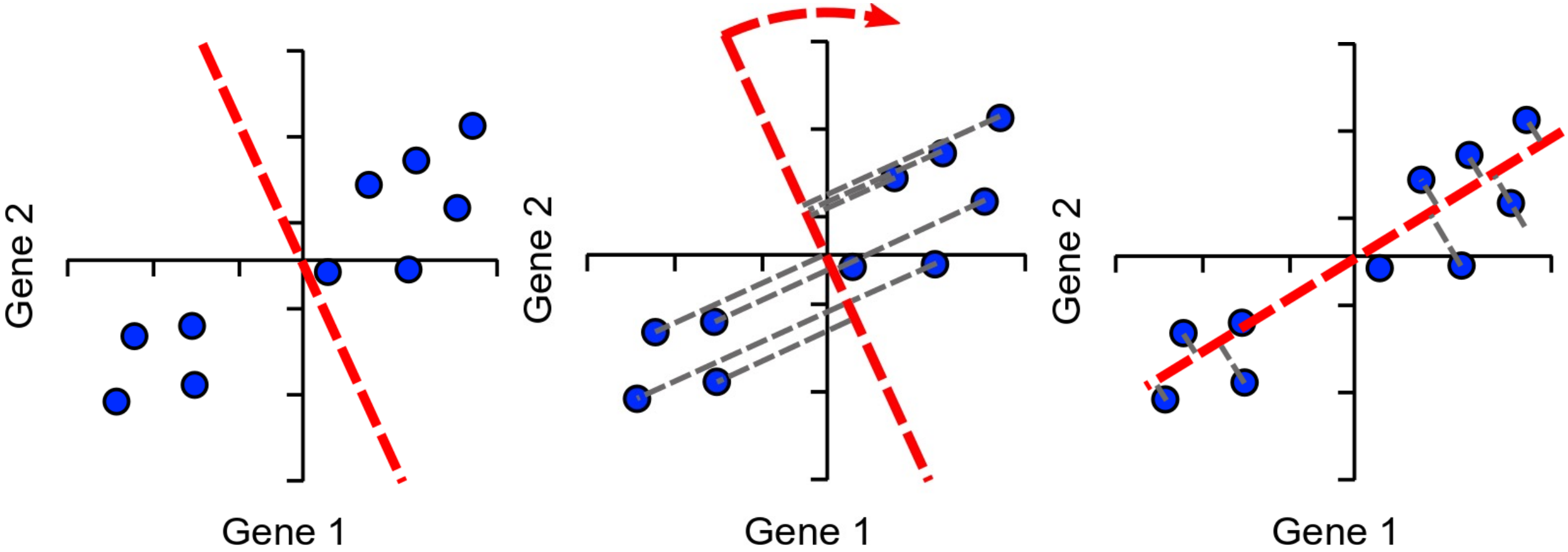
How does PCA work?

- Simple example using 2 genes and 10 cells

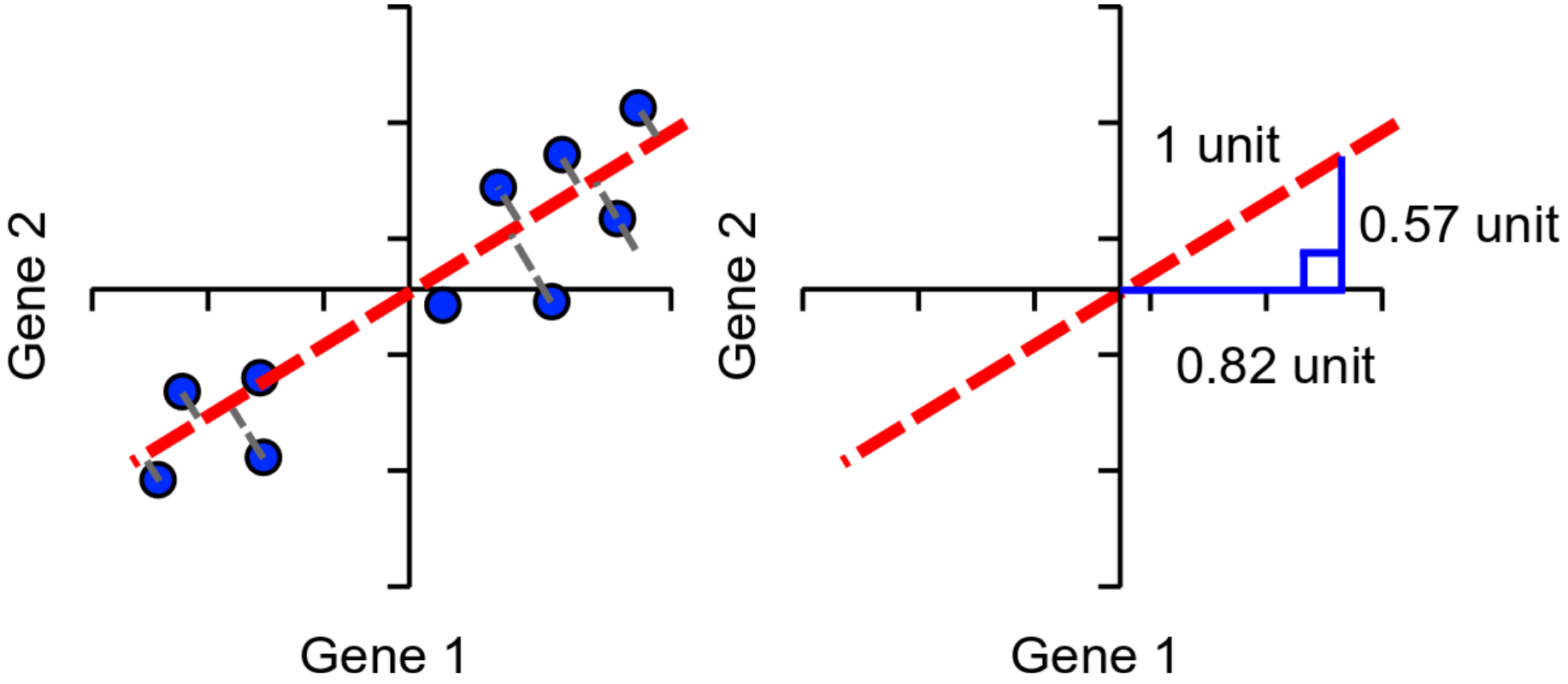


How does PCA work?

- Find line of best fit, passing through the origin



Assigning Loading to Genes



Single Vector or '**eigenvector**'

Loadings:

- Gene1 = 0.82
- Gene2 = 0.57

Higher loading equals more influence on PC

More Dimensions

- The same idea extends to larger numbers of dimensions (n)

- Calculation of first PC rotates in (n-1) -dimensions
  - Next PC is perpendicular to PC2, but rotated similarly (n-2)
  - Last PC is remaining perpendicular (no choice)
  - Same number of PCs as genes

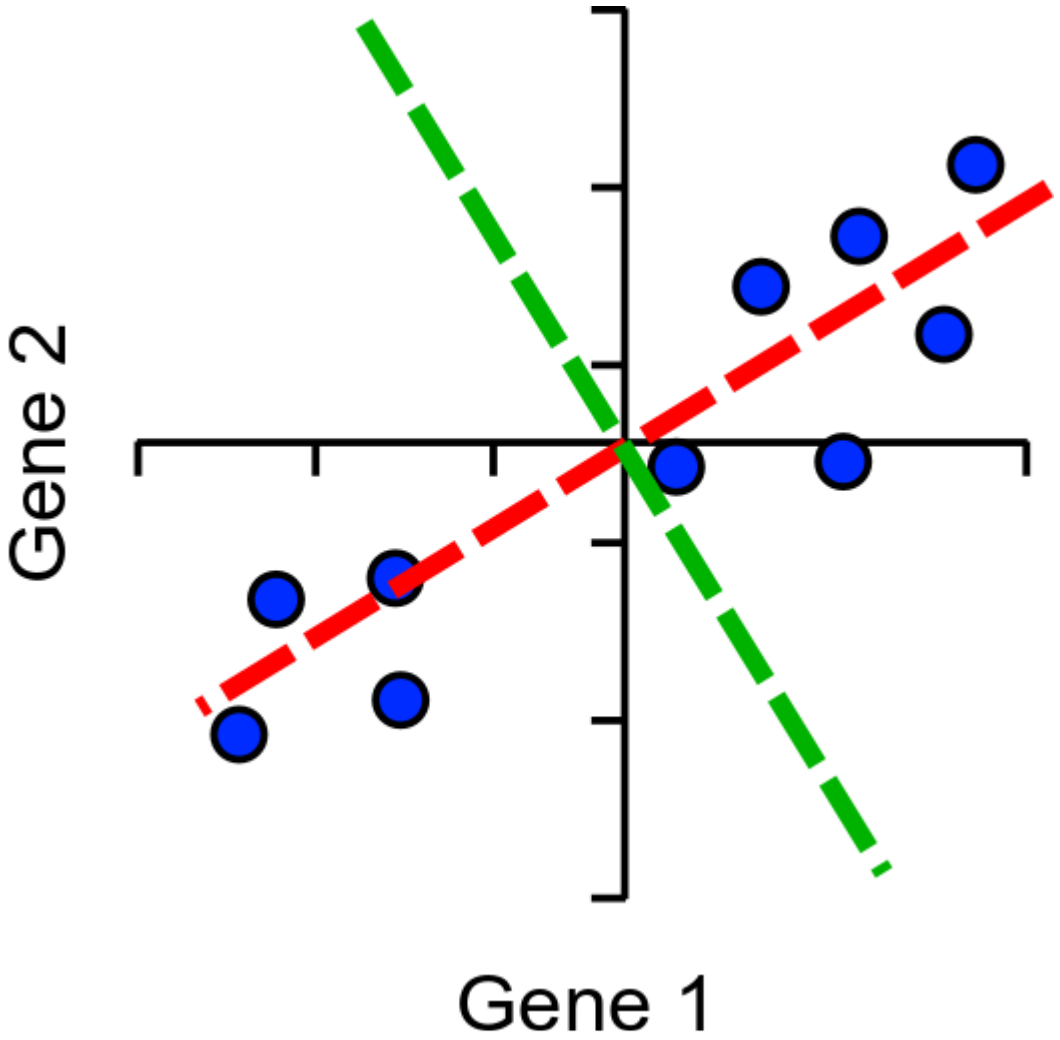
Explaining Variance

- Each PC always explains some proportion of the total variance in the data. Between them they explain everything
  - PC1 always explains the most
  - PC2 is the next highest etc. etc.

- Since we only plot 2 dimensions we'd like to know that these are a good explanation

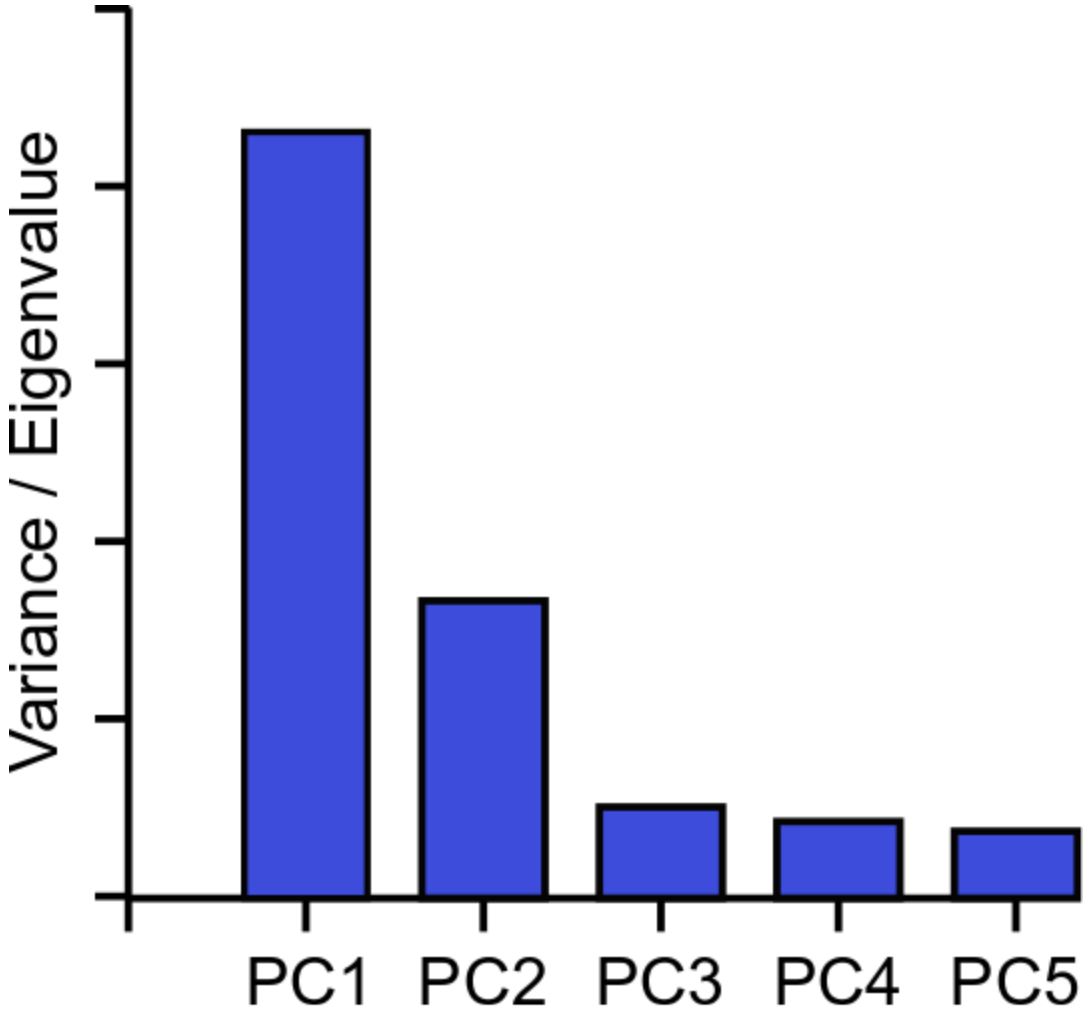
- How do we calculate this?

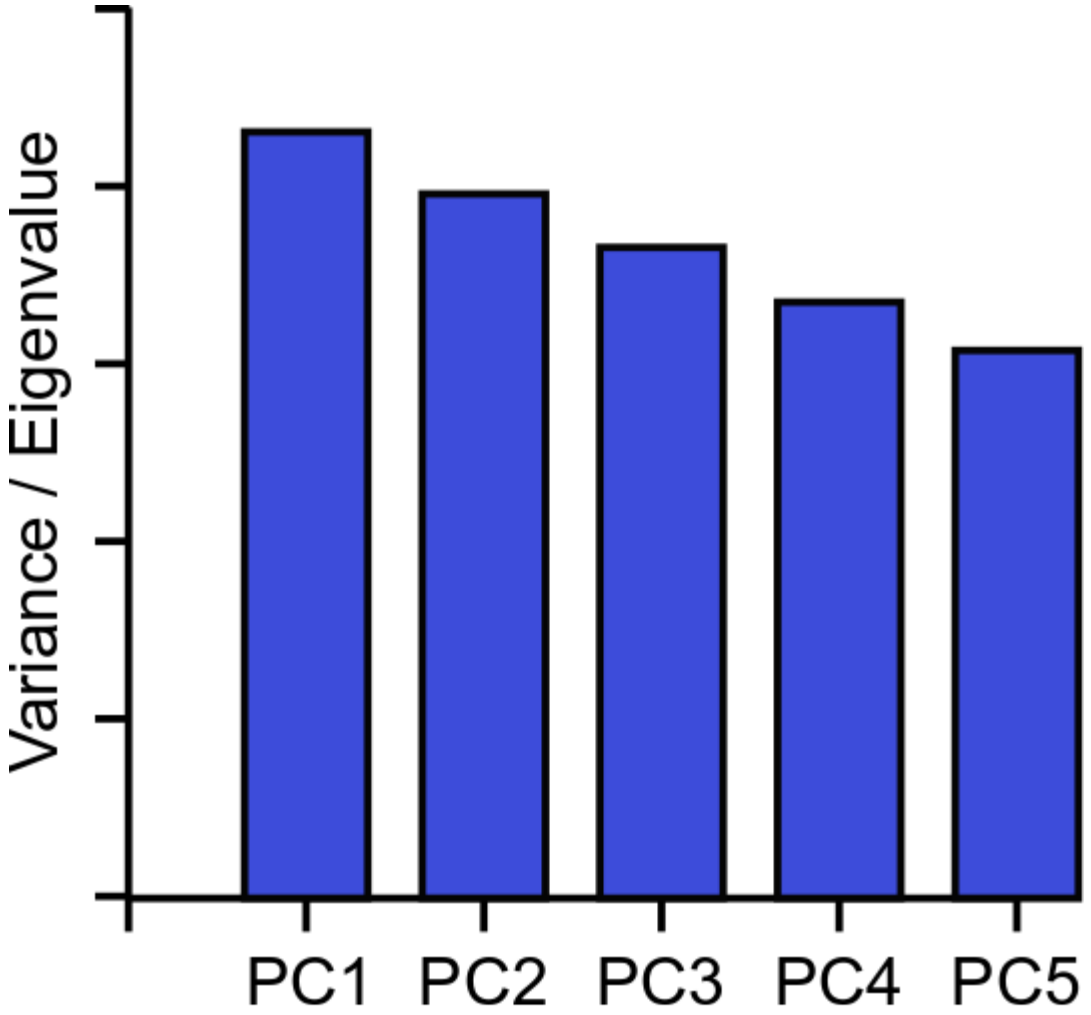
Explaining Variance



- Project onto PC
- Calculate distance to the origin
- Calculate sum of squared differences (SSD)
  - This is a measure of variance called the ‘eigenvalue’
  - Divide by (points-1) to get actual variance

Explaining Variance – Scree Plots





Problem: What if I have more than two really important PCA values

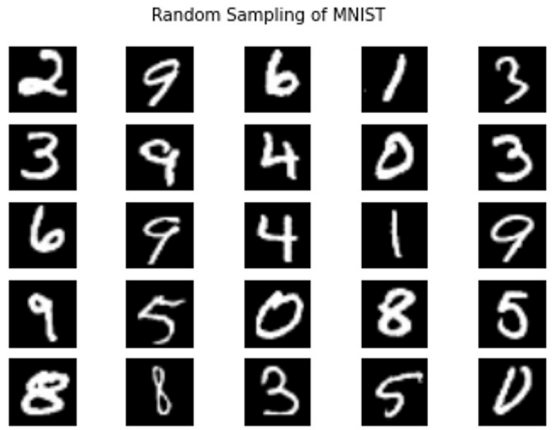
How can humans visualize
high-dimensional data?

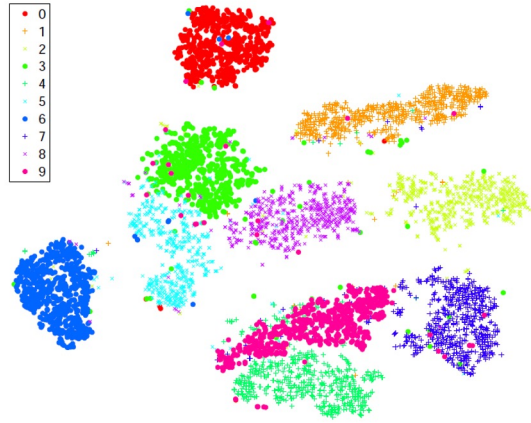
- images (pixels can be considered dimensions)
- single cell nuclei is described by 30 variables (variables = dimensions)

Specifically …

How to visualize in 2D or 3D space?

- to see relationships
- check if features are separating the data



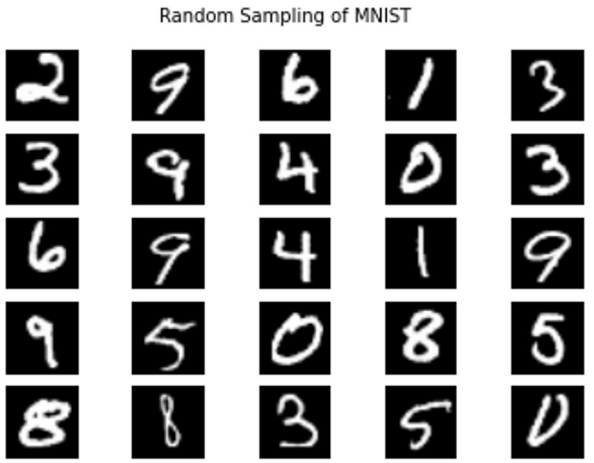


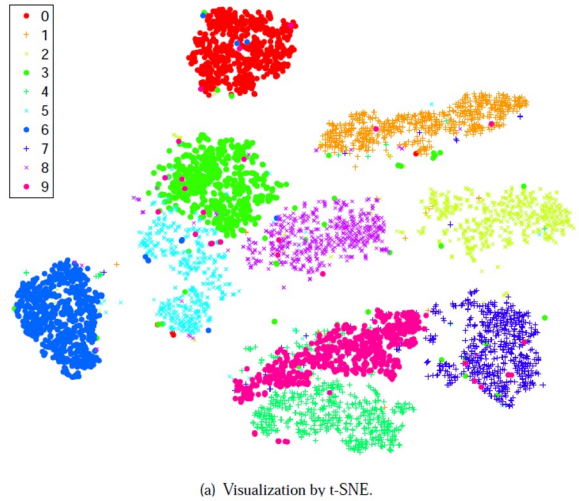
(a) Visualization by t-SNE.

http://arxiv.org/pdf/1310.1531.pdf

What is t-SNE?

- t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Unsupervised (does not use class labels)
- Models the probabilities that **high-d** points *xi,xj* are neighbors
- Models the probabilities the corresponding **low-d** points *yi,yj* are neighbors
- Finds a **low-d** representation that is faithful to the **high-d** model

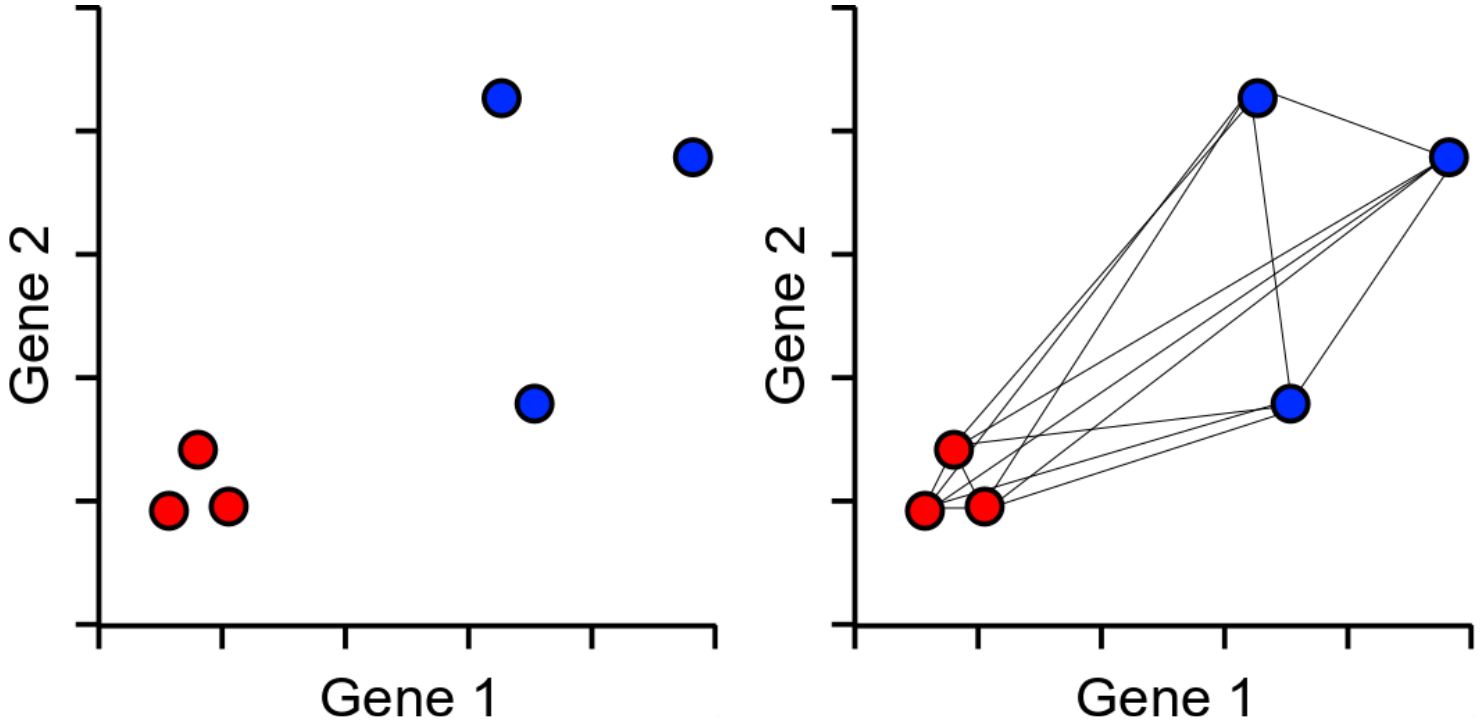




(a) Visualization by t-SNE.

How does tSNE work?

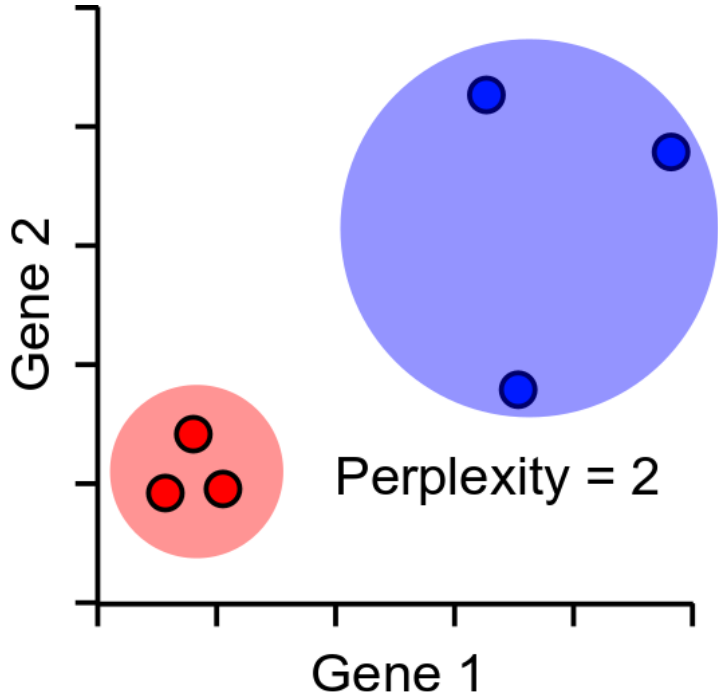
Based around all-vs-all table of pairwise cell to cell distances



	0	10	10	295	158	153
	9	0	1	217	227	213
	1	8	0	154	225	238
	205	189	260	0	23	45
	248	227	246	44	0	54
	233	176	184	41	36	0

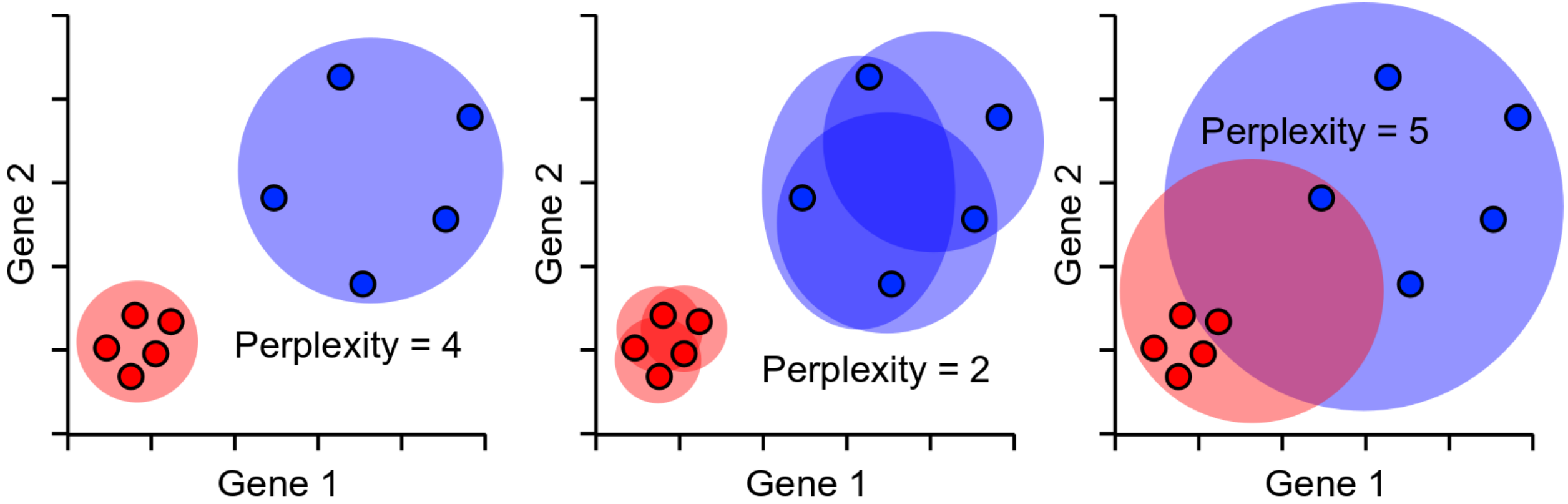
Distance scaling and perplexity

- Perplexity = expected number of neighbours within a cluster
- Distances scaled relative to perplexity neighbours
- The performance of TSNE is fairly robust to changes in the perplexity, and typical values are between 5 and 50



	0	4	6	586	657	836
	4	0	4	815	527	776
	9	3	0	752	656	732
	31	28	29	0	4	7
	31	24	25	4	0	7
	40	37	32	8	8	0

Perplexity Robustness

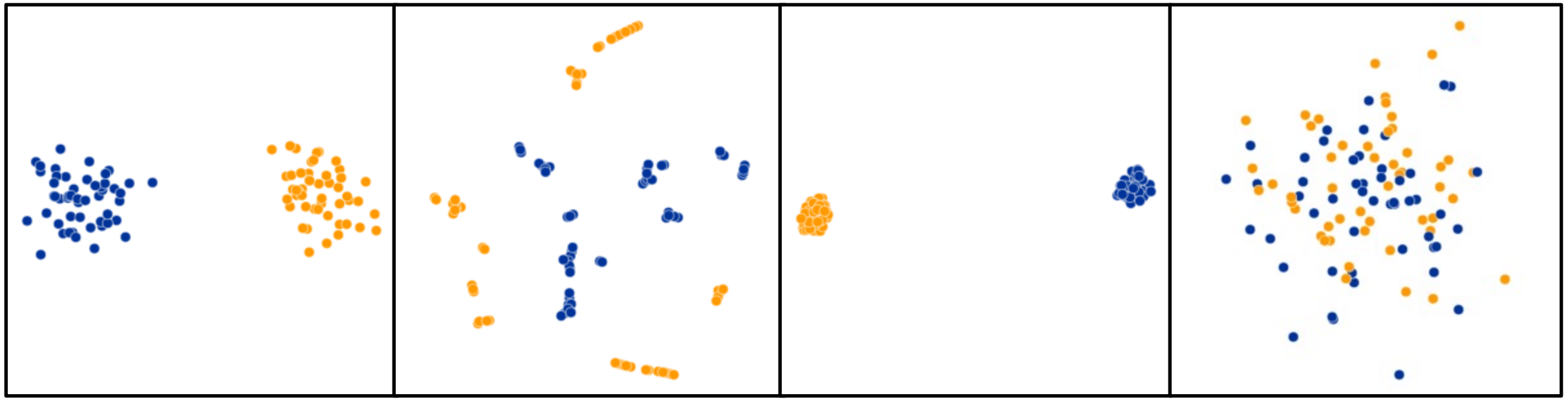


tSNE Projection

- Normally projects in 2D, but can be any number of dimensions
- Randomly scatter all points within the space
- Start a simulation
  - Aim is to make the point distances match the distance matrix
  - Shuffle points based on how well they match
  - Stop after fixed number of iterations, or
  - Stop after distances have converged

tSNE Practical Examples

Perplexity Settings Matter

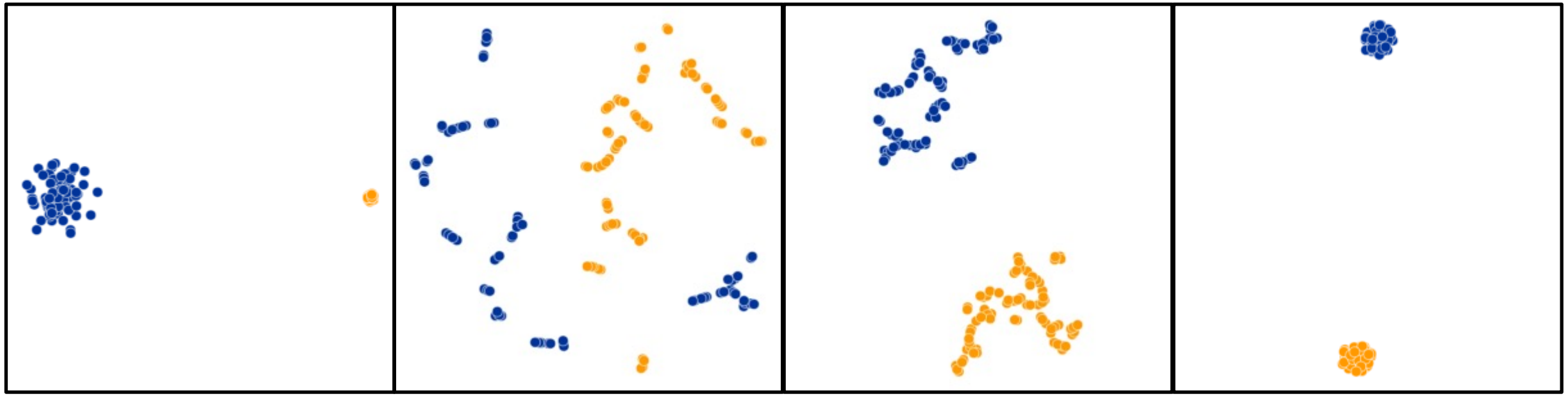


Original | Perplexity = 2 | Perplexity = 30 | Perplexity = 100

https://distill.pub/2016/misread-tsne/

tSNE Practical Examples

Cluster Sizes are Meaningless

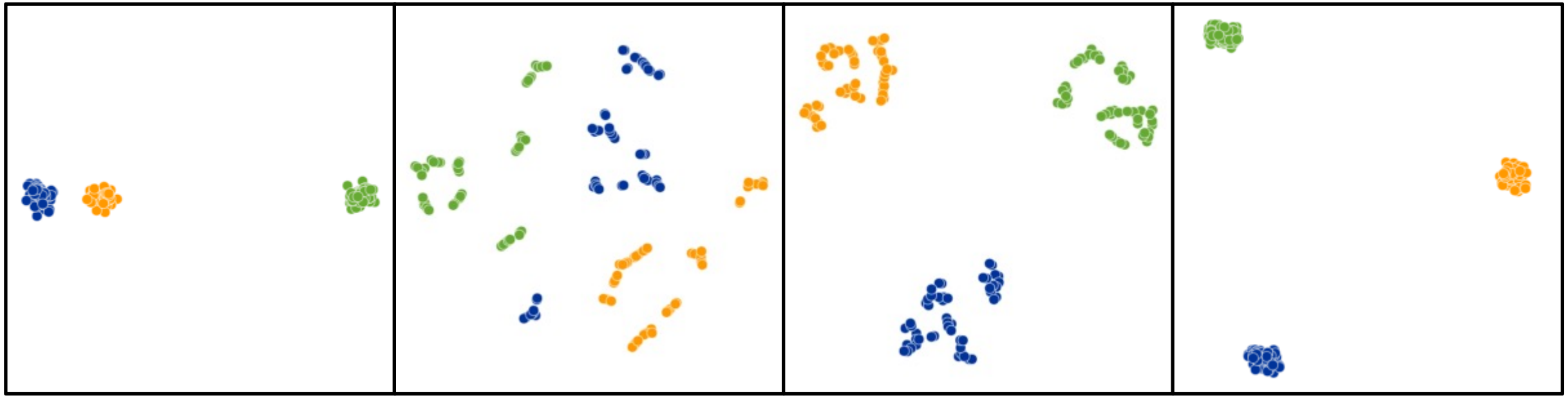


Perplexity = 2 Perplexity = 5 Perplexity = 50

https://distill.pub/2016/misread-tsne/

tSNE Practical Examples

Distances between clusters can’t be trusted



Perplexity = 2 Perplexity = 5 Perplexity = 30

https://distill.pub/2016/misread-tsne/

Shortcomings?

- PCA
  - May Require more than 2 dimensions
  - Thrown off by quantised data
  - Expects linear relationships
- tSNE
  - Can’t cope with noisy data
  - Loses the ability to cluster

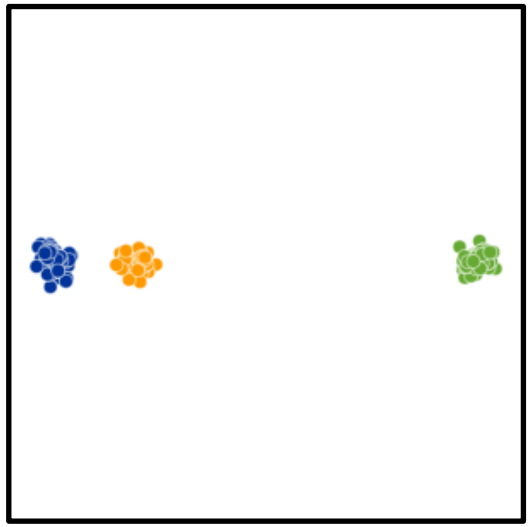
Answer: Combine the two methods, get the best of both worlds

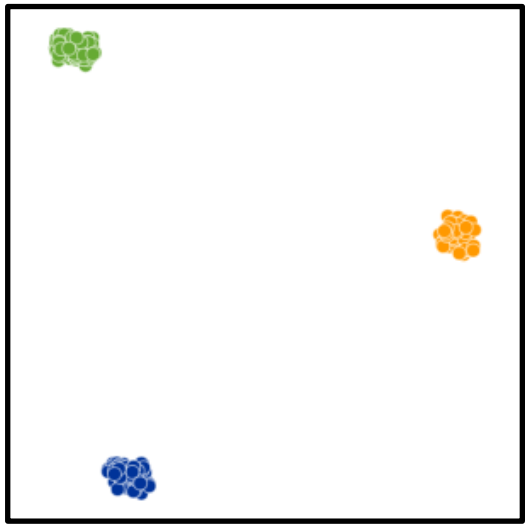
- PCA
  - Good at extracting signal from noise
  - Extracts informative dimensions
- tSNE
  - Can reduce to 2D well
  - Can cope with non-linear scaling

This is what CellRanger does in its default analysis

So PCA + tSNE is great then?

- Up to some extent..
  - tSNE is slow. This is probably it's biggest disadvantage
    - tSNE doesn't scale well to large numbers of cells (10k+)
  - tSNE only gives reliable information on the closest neighbours
  - large distance information is almost irrelevant





UMAP to the rescue!

- UMAP is a replacement for tSNE to fulfil the same role
- Conceptually very similar to tSNE, but with a couple of relevant (and somewhat technical) changes
- Practical outcome is:
  - UMAP is quite a bit quicker than tSNE
  - UMAP can preserve more global structure than tSNE*
  - UMAP can run on raw data without PCA preprocessing*
  - UMAP can allow new data to be added to an existing projection

* In theory, but possibly not in practice

UMAP differences

- Instead of the single perplexity value in tSNE, UMAP defines
  - **Nearest neighbours**: the number of expected nearest neighbours – basically the same concept as perplexity
  - **Minimum distance**: how tightly UMAP packs points which are close together
- Nearest neighbours will affect the influence given to global vs local information.
- Min dist will affect how compactly packed the local parts of the plot are.

UMAP differences

- Speed – mostly a level of maths I'm not going to get into!
  - UMAP skips a normalisation step in the calculation of high dimensional distances which speeds it up
  - In the 2D projection UMAP uses a more efficient method to shuffle the cells into their final position
    - Doesn't have to measure every cell to decide on what to move
    - Uses an algorithm which can be multi-threaded
    - Algorithm is more deterministic, allowing more data to be projected later

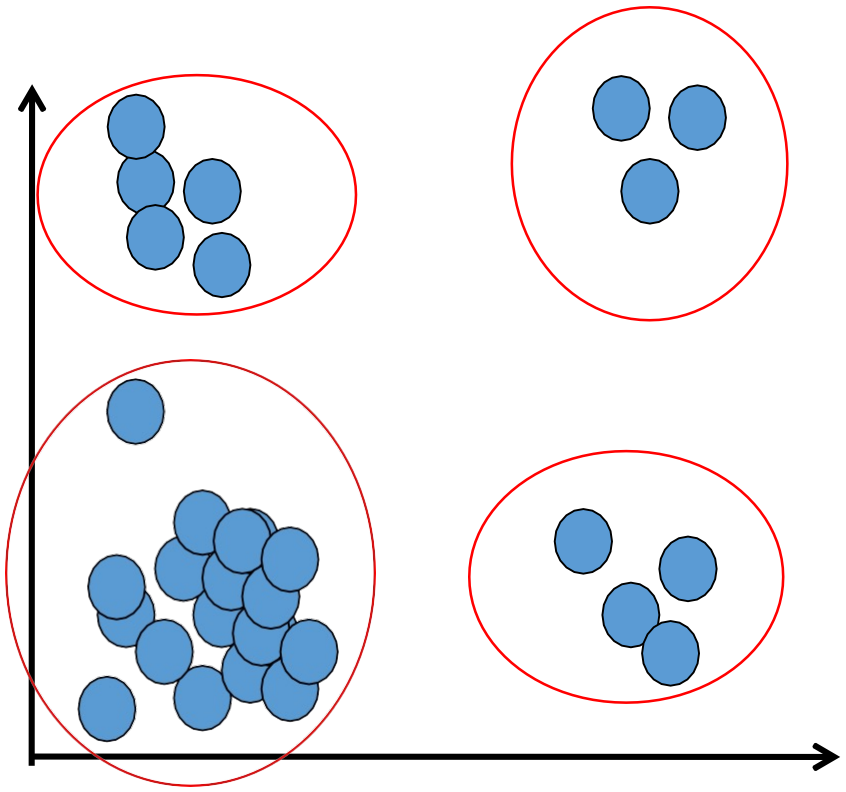
Practical approach PCA + tSNE/UMAP

- Filter heavily before starting
  - Nicely behaving cells
  - Expressed genes
  - Variable genes

- Do PCA
  - Extract most interesting signal
  - Take top PCs. Reduce dimensionality (but not to 2)

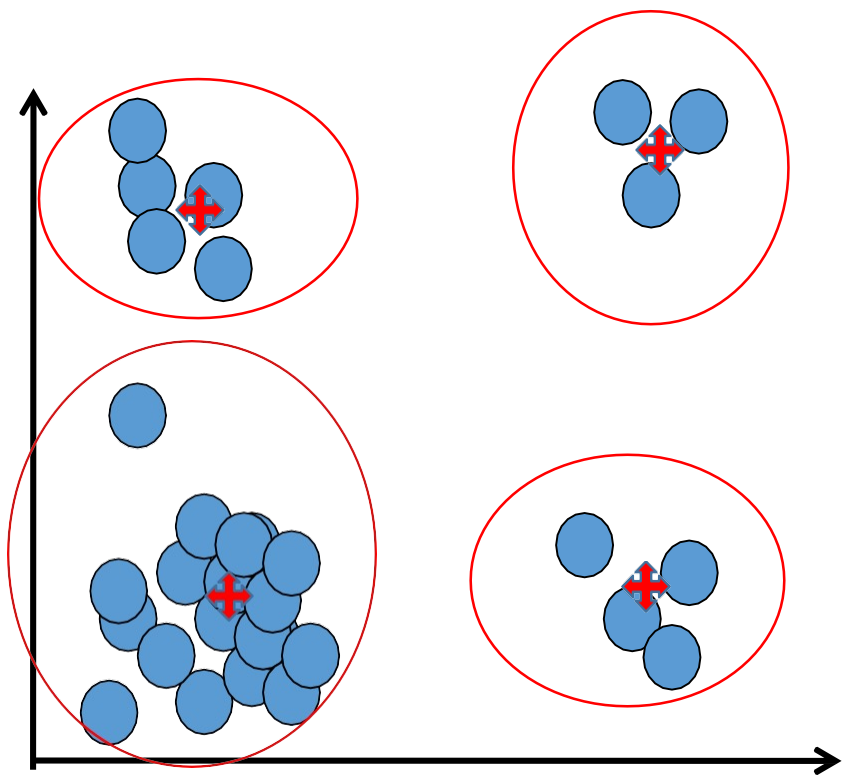
- Do tSNE/UMAP
  - Calculate distances from PCA projections
  - Scale distances and project into 2-dimensions

Clustering Motivation



- The goal of clustering is to
  - group data points that are close (or **similar**) to each other
  - identify such groupings (or clusters) in an **unsupervised** manner
- How to define similarity ?
- How many iterations for checking cluster quality ?

Cluster



- A cluster is represented by a single point, known as centroid (or cluster center) of the cluster.
- Centroid is computed as the mean of all data points in a cluster

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_j$$

- Cluster boundary is decided by the farthest data point in the cluster.

Applications

- **Example 1**: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.
- **Example 2**: In marketing, segment customers according to their similarities
  - To do targeted marketing.
- **Example 3**: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy

Types of clustering

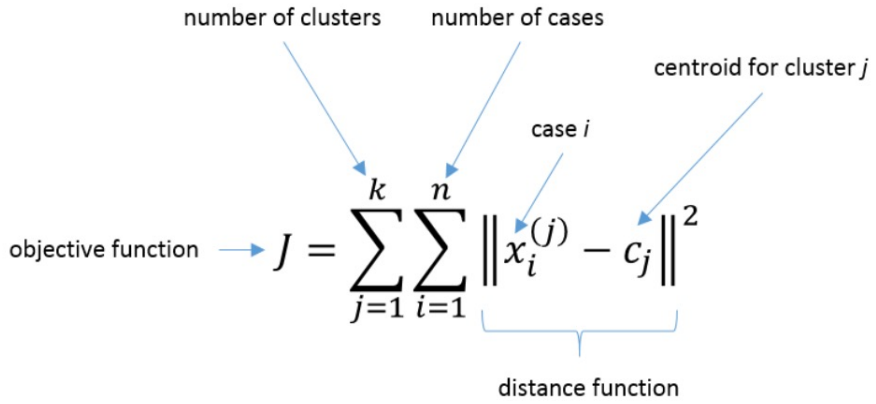
- Clustering: Task of grouping a set of data points such that data points in the same group are more similar to each other than data points in another group (group is known as cluster)
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.

Types:

1. Exclusive Clustering: K-means
2. Overlapping Clustering: Fuzzy C-means
3. Hierarchical Clustering: Agglomerative clustering, divisive clustering
4. PCA
5. T-SNE
6. UMAP

1. Exclusive clustering: K-means

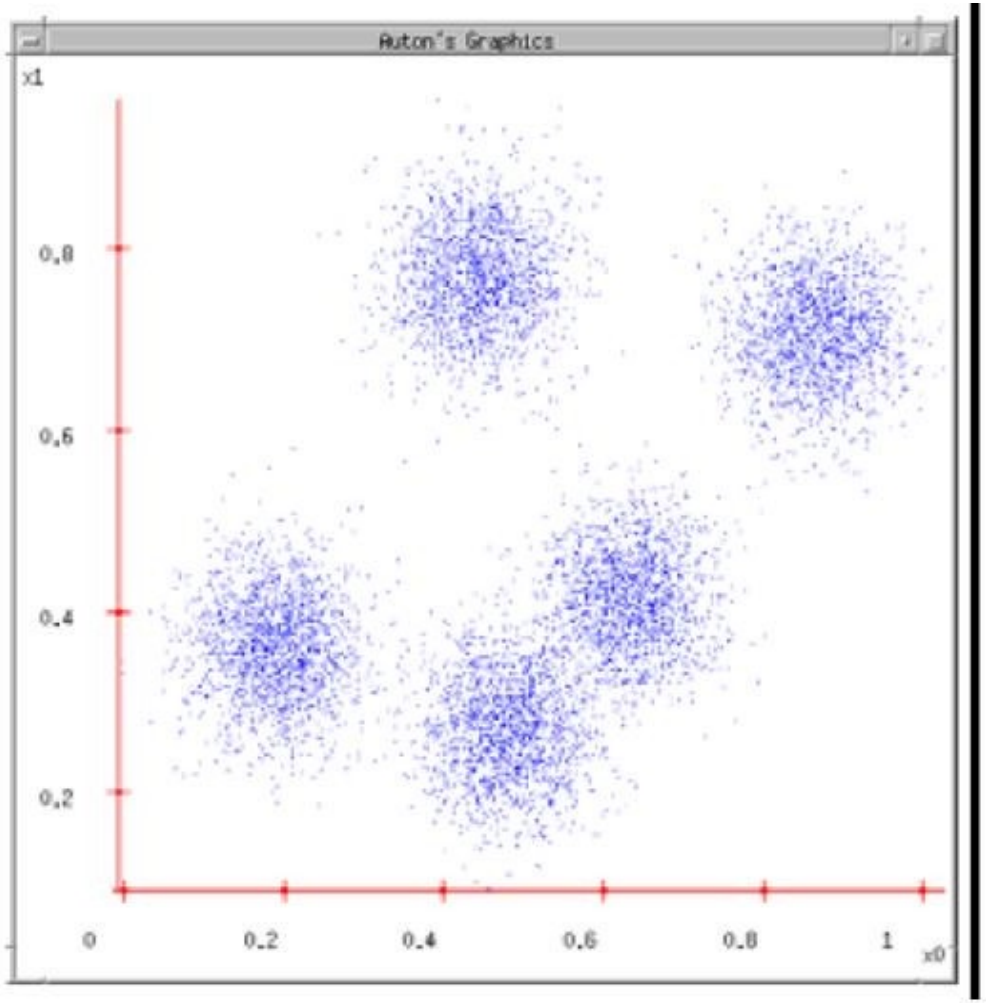
- Basic idea: randomly initialize the *k* cluster centers, and iterate between the two steps explained below.
  1. Randomly initialize the cluster centers, c_1, ..., c_K
  2. Given cluster centers, determine points in each cluster
     - For each point p, find the closest c_i. Put p into cluster i
  3. Given points in each cluster, solve for c_i
     - Set c_i to be the mean of points in cluster i
  4. If c_i have changed, repeat Step 2



K-means example

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*

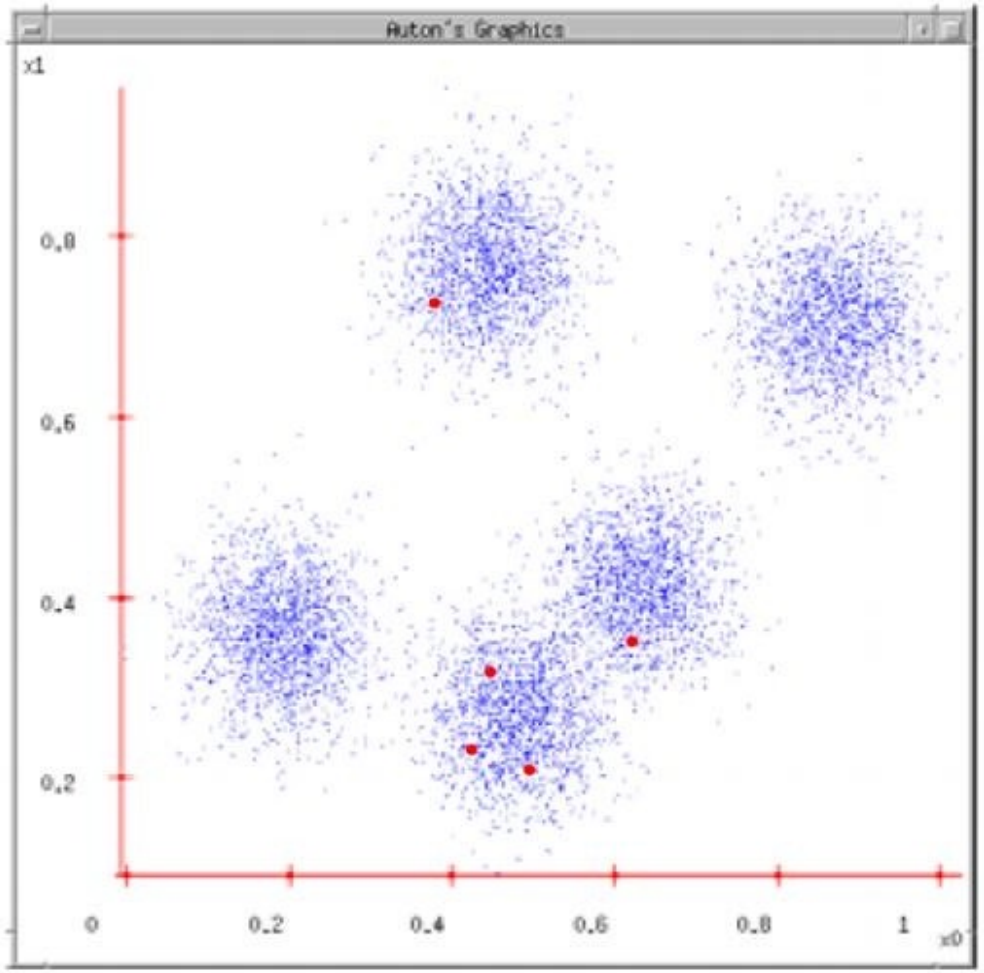


This slide is taken from: Andrew Moore

Example contd..

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations

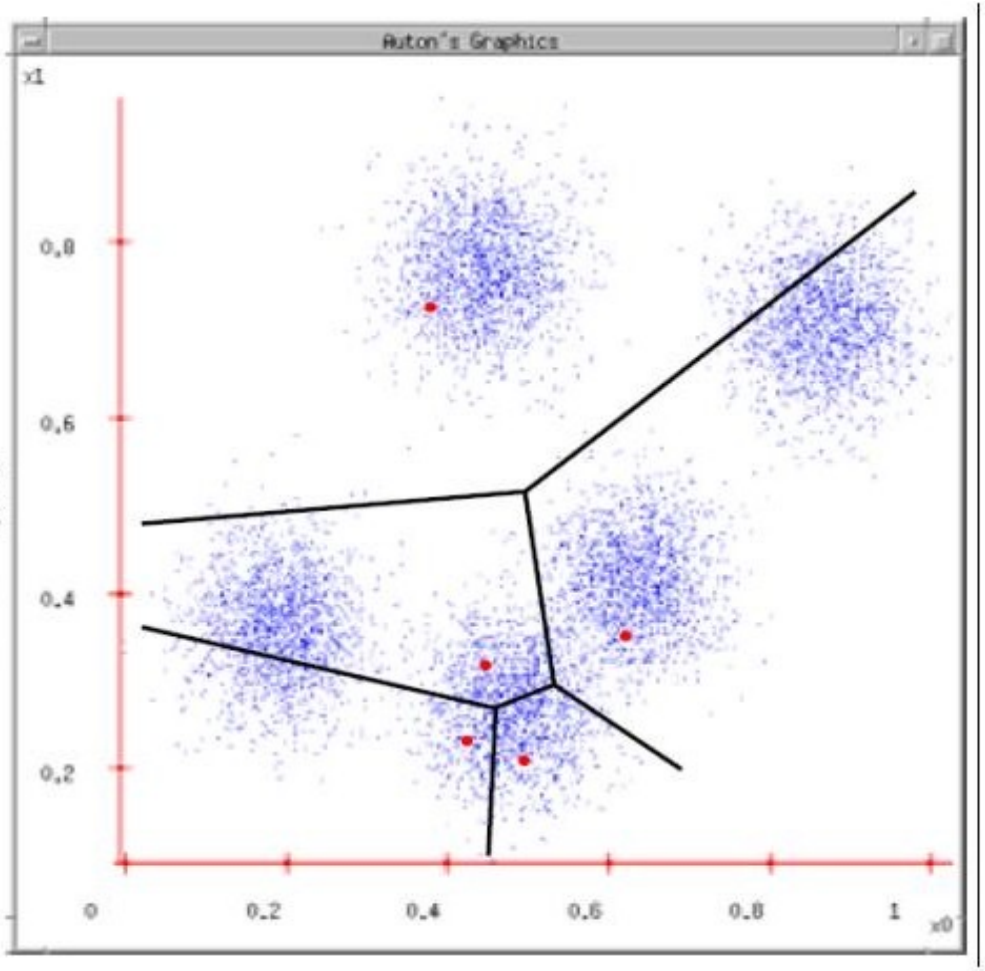


This slide is taken from: Andrew Moore

Example contd..

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)

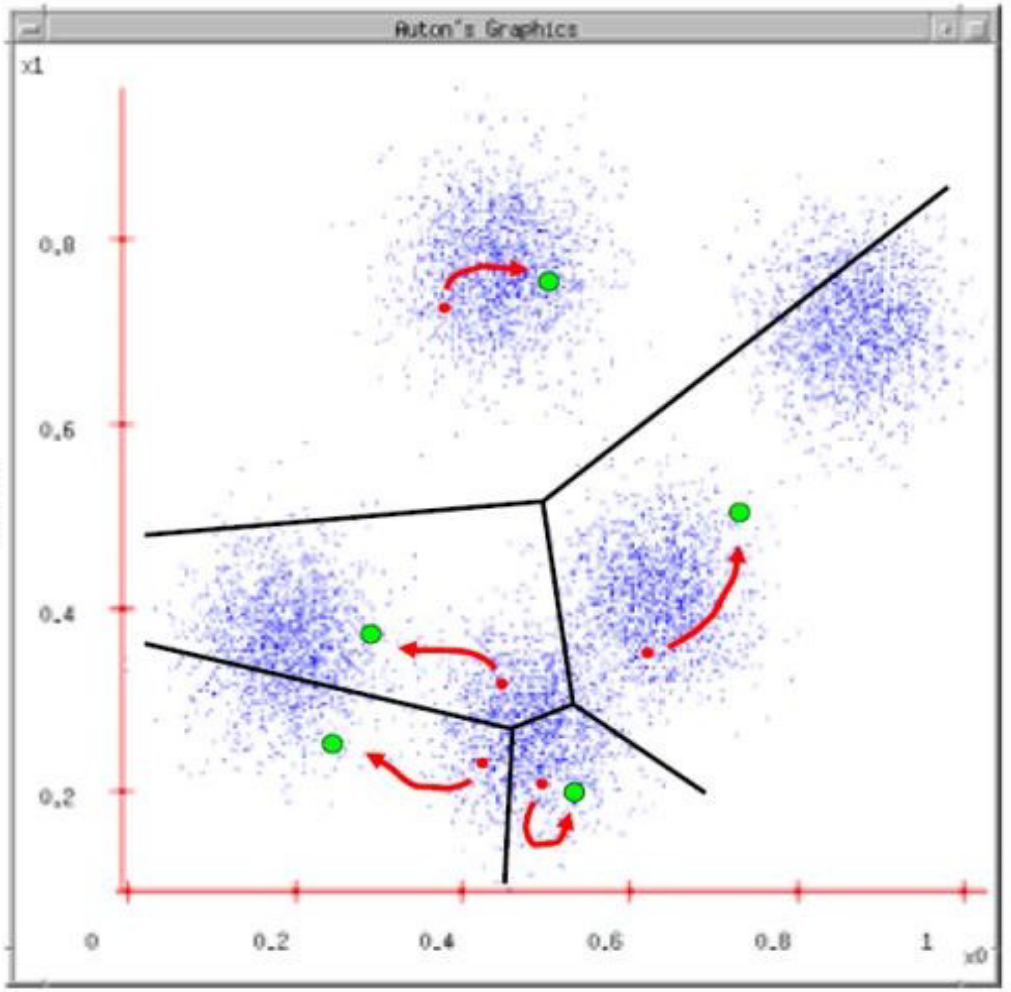


This slide is taken from: Andrew Moore

Example contd..

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns

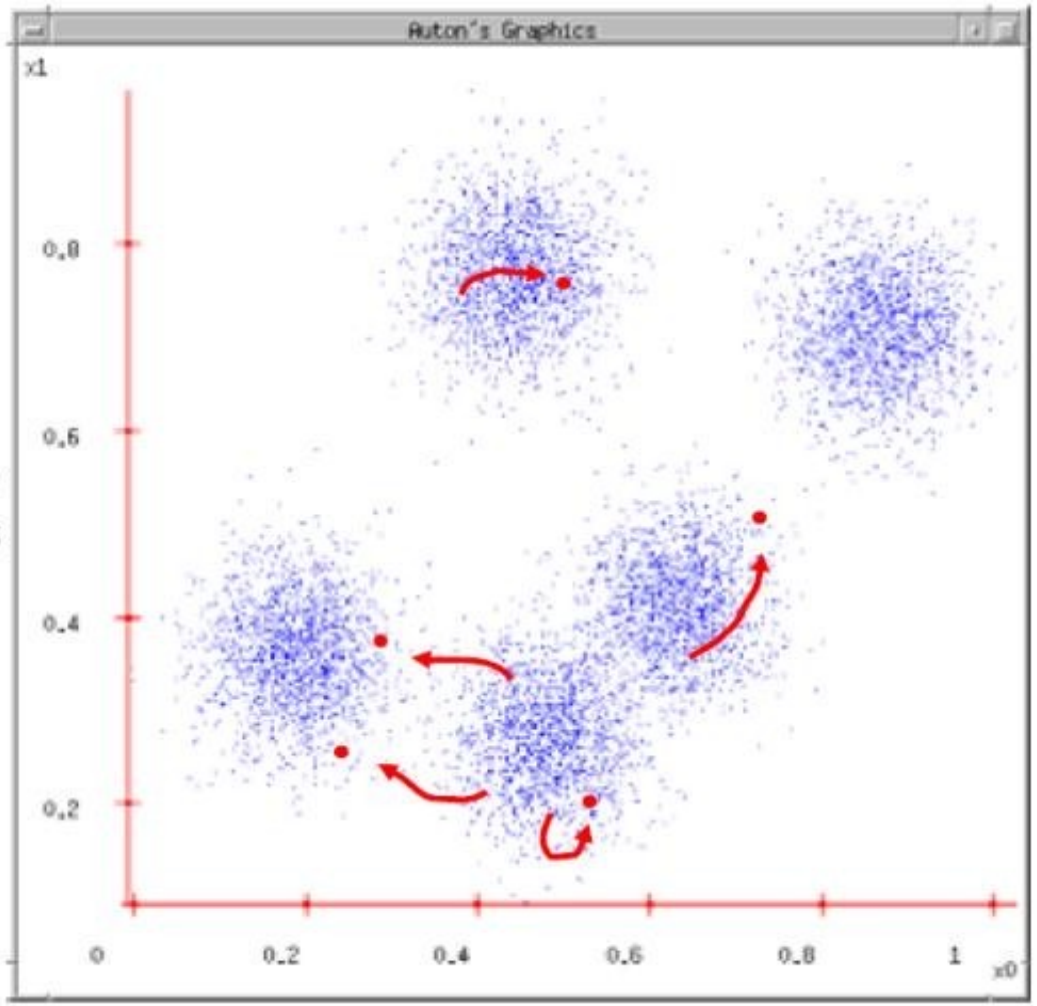


This slide is taken from: Andrew Moore

Example contd..

K-means

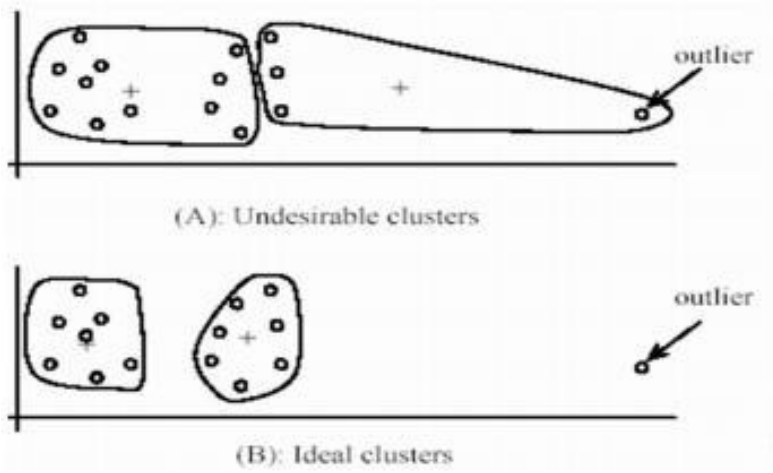
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!



This slide is taken from: Andrew Moore

Contd..

- Pros
  - Simple, fast to compute
  - Converges to local minimum of within-cluster squared error
- Cons
  - Setting k?
  - Sensitive to initial centers
  - Sensitive to outliers
  - Detects spherical clusters
  - Assuming means can be computed



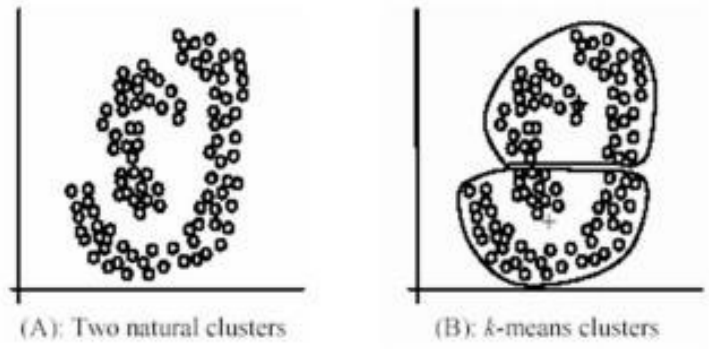


Image source: Google

2. Fuzzy C-Means Clustering

- One data point may belong to two or more cluster with different memberships.
- Objective function:

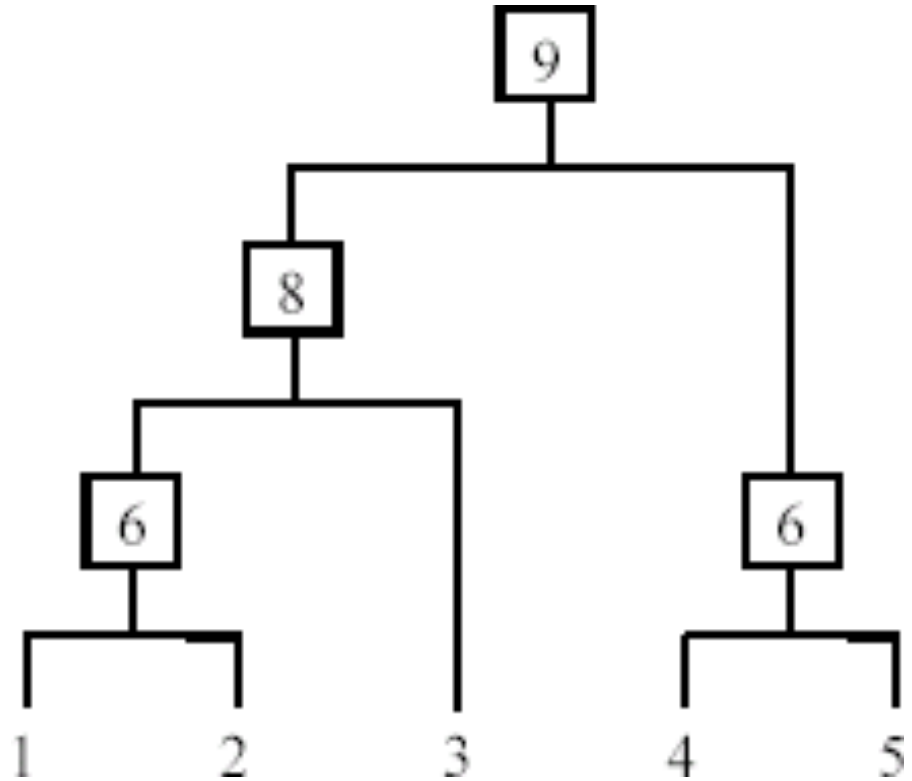
$$J = \sum_{j=1}^{K} \sum_{i=1}^{n} u_{i,j}^{m} ||x_i^J - c_j||^2$$

where $1 \leq m < \infty$

- An extension of k-means

3. Hierarchical Clustering

- Produce a nested sequence of clusters, a tree, also called Dendrogram.



Types of hierarchical clustering

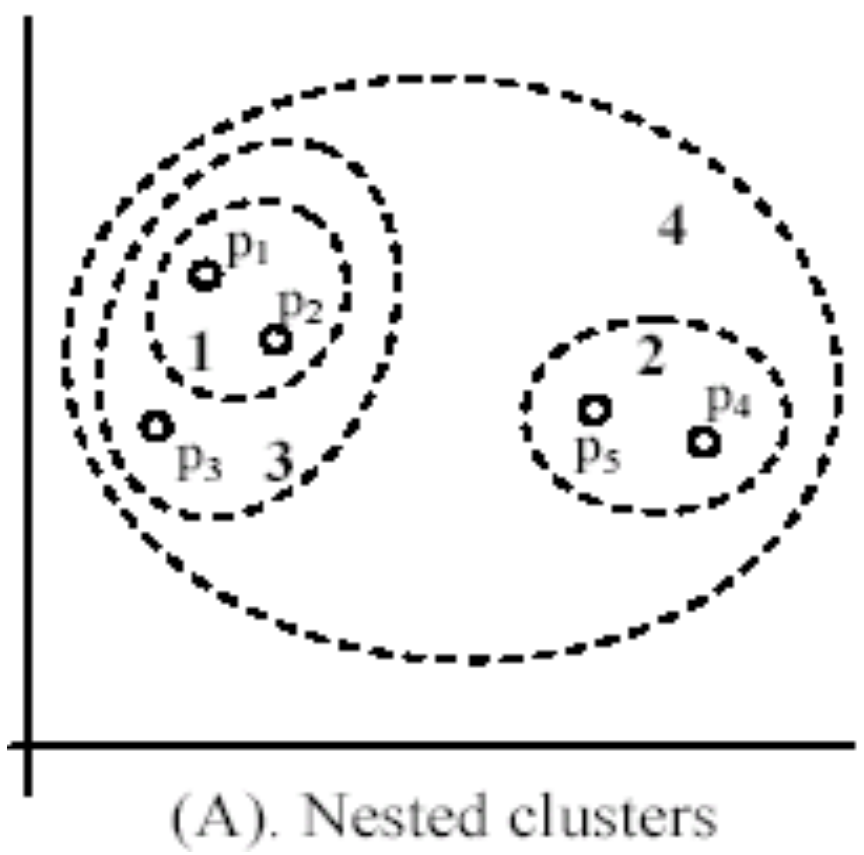
- Agglomerative (bottom up) clustering: It builds the dendrogram (tree) from the bottom level, and
  - merges the most similar (or nearest) pair of clusters
  - stops when all the data points are merged into a single cluster (i.e., the root cluster).
- Divisive (top down) clustering: It starts with all data points in one cluster, the root.
  - Splits the root into a set of child clusters. Each child cluster is recursively divided further
  - stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point

Agglomerative clustering

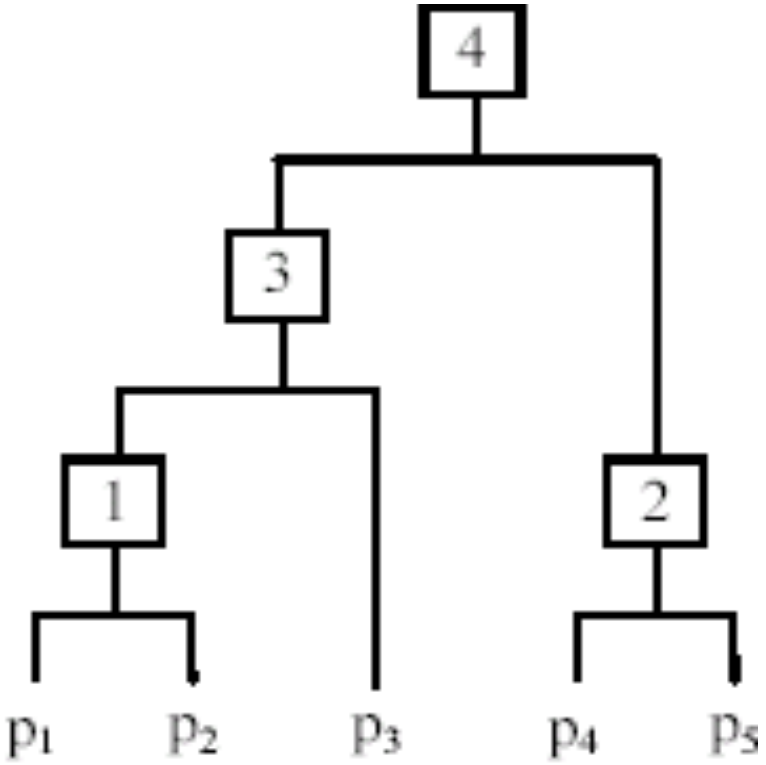
It is more popular than divisive methods.

- At the beginning, each data point forms a cluster (also called a node).
- Merge nodes/clusters that have the least distance.
- Go on merging
- Eventually all nodes belong to one cluster

An example: working of the algorithm



(A). Nested clusters



(B) Dendrogram

Hierarchical clustering

- Pros
  - Dendograms are great for visualization
  - Provides hierarchical relations between clusters
  - Shown to be able to capture concentric clusters
- Cons
  - Not easy to define levels for clusters
  - Experiments showed that other clustering techniques outperform hierarchical clustering

Normalization

- Technique to force the attributes to have a common value range
- What is the need ?
  - Consider the following pair of data points

    $\mathbf{x}_i$: (0.1, 20) and $\mathbf{x}_j$: (0.9, 720).

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(0.9 - 0.1)^2 + (720 - 20)^2} = 700.000457,$$

- Two main approaches to standardize interval scaled attributes, **range** and **z-score**. *f* is an attribute

$$range(x_{if}) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)},$$

Contd..

- Z-score: transforms the attribute values so that they have a mean of zero and a **mean absolute deviation** of 1. The mean absolute deviation of attribute *f*, denoted by s_f, is computed as follows

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \ldots + x_{nf}\right),$$

$$s_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|\right),$$

Z-score:

$$z(x_{if}) = \frac{x_{if} - m_f}{s_f}.$$

How to choose a clustering algorithm

- A vast collection of algorithms are available. Which one to choose for our problem ?
- Choosing the “best” algorithm is a challenge.
  - Every algorithm has limitations and works well with certain data distributions.
  - It is very hard, if not impossible, to know what distribution the application data follow. The data may not fully follow any “ideal” structure or distribution required by the algorithms.
  - One also needs to decide how to standardize the data, to choose a suitable distance function and to select other parameter values.

Contd..

- Due to these complexities, the common practice is to
  - run several algorithms using different distance functions and parameter settings, and
  - then carefully analyze and compare the results.
- The interpretation of the results must be based on insight into the meaning of the original data together with knowledge of the algorithms used.
- Clustering is highly application dependent and to certain extent subjective (personal preferences).

Summary

- Studied need for unsupervised learning
- Types of clustering dimensionality reduction:
  - K-means, hierarchical, PCA, t-SNE, UMAP
- Similarity functions:
  - Euclidean distance, Manhattan distance
- Stopping criteria:
  - SSD
- Which algorithm to choose