Unsupervised Learning

– Emma Duan

Quiz

- ✘ What are unsupervised methods, how do they differ from supervised learning?
- ✘ Name a few methods for clustering?
- ✘ How do you select parameters for k-means (k, iteration)?
- ✘ How do you evaluate clustering results?
- ✘ What are some dimension reduction methods?

Topics

- ✘ Intro to unsupervised learning
- ✘ Clustering
  - k-means
  - DBSCAN
- ✘ Dimensionality reduction
  - PCA
  - Matrix factorization

Intro to Unsupervised Learning

Unsupervised Learning

"Given only samples X of the data, we compute a function f such that y = f(X) is "simpler". "

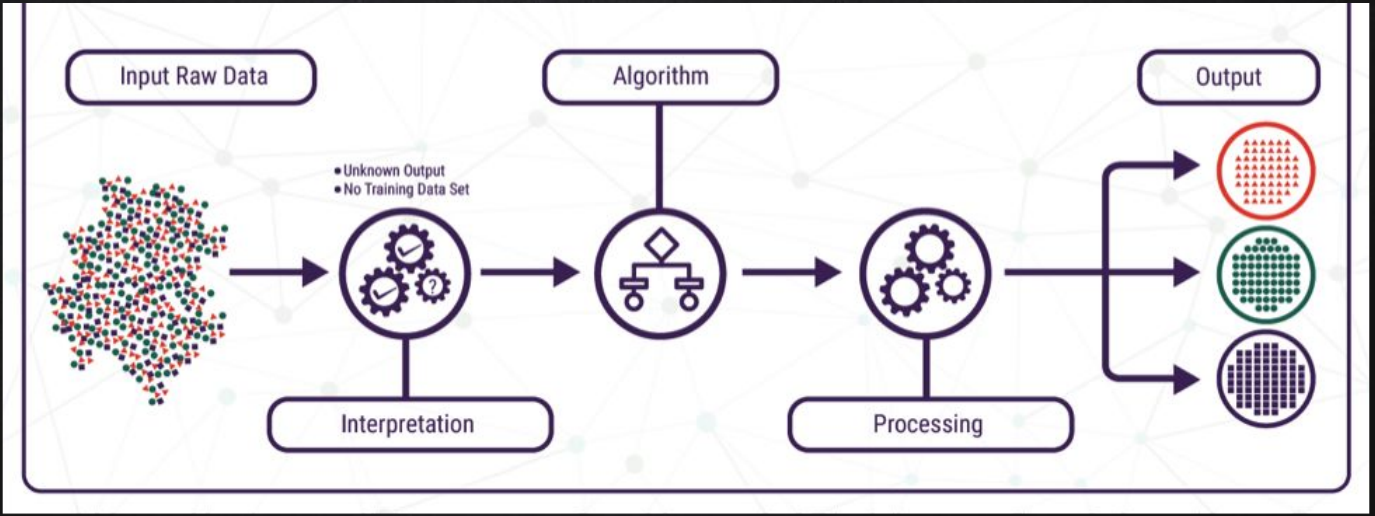
- ✘ Key difference with supervised learning: no label/target, y here is unknown
- ✘ If y is discrete, this is clustering
- ✘ If y is continuous, this is dimensionality reduction

Unsupervised learning algorithm focuses solely on detecting structure in unlabelled input data.

Clustering

Clustering

Goal: to find homogeneous subgroups within the data. The grouping is based on how similar the observations are to each other, or the distance between observations.



https://www.technative.io/why-unsupervised-machine-learning-is-the-future-of-cybersecurity/

K-MEANS

- ✘ The standard k-means algorithm is based on Euclidean distance.
- ✘ The cluster quality measure is an intra-cluster measure only, equivalent to the sum of item-to-centroid kernels.

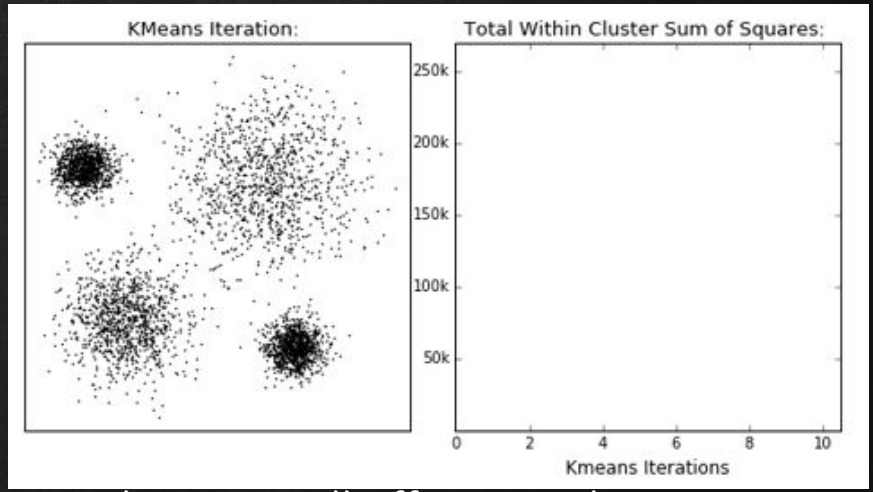
$$minimize\left\{\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}\left(x_{ij}-x_{i'j}\right)^2\right\}$$

- ✘ A simple greedy algorithm locally optimizes this measure (usually called Lloyd's algorithm):
  - Randomly initialize k centroids for cluster
  - Iterate the following steps (n iters or until convergence):
    - Compute cluster centroid by taking the mean of observations in cluster
    - Find the closest cluster center for each observation, and assign it to that cluster

K-MEANS

Key parameters:

- ✘ k: number of clusters
  - Use elbow plot or silhouette score
  - AIC (lower is better)
- ✘ n: iteration
  - Fixed
  - Until no change
  - Until small change tolerance

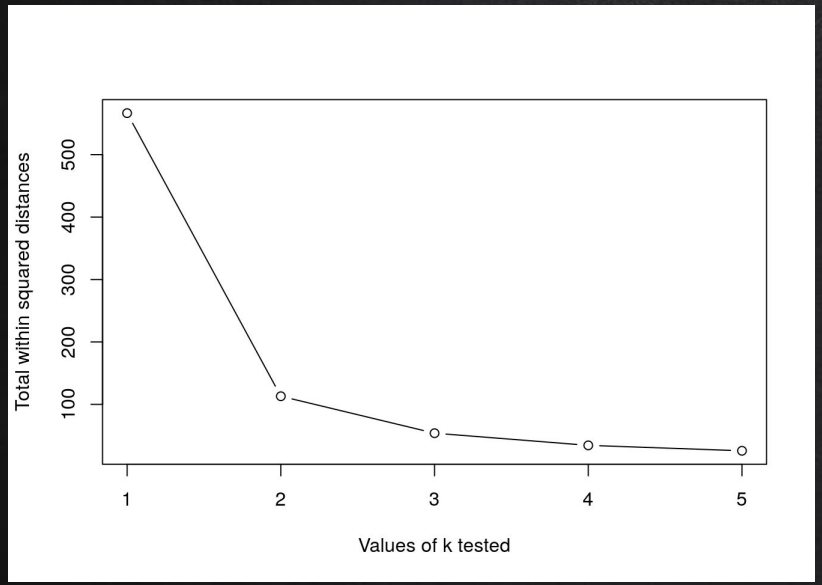


The solution is local optima and unstable – initialization will affect results.

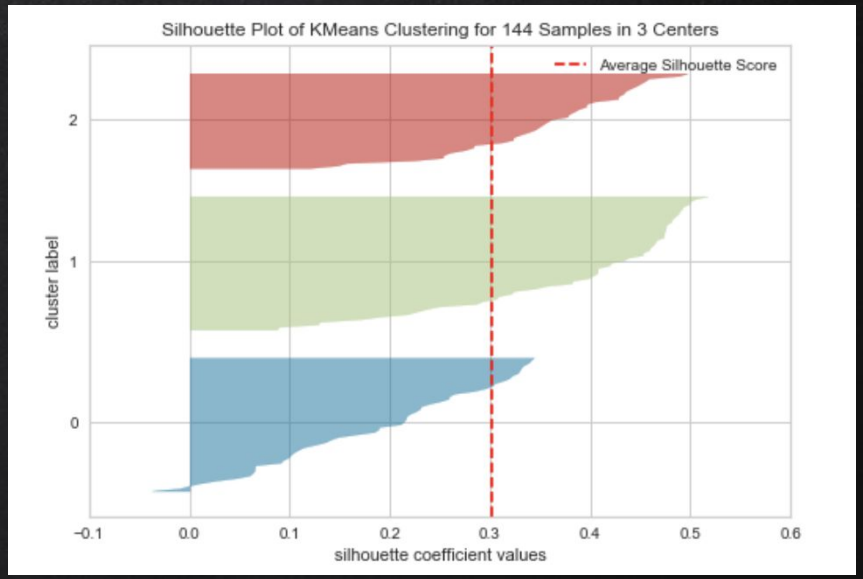
Technically only works well on lower dimension data (in high dim Euclidean distance is not a good metric, maybe switch to a different one)

https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb#:~:text=The%20Elbow%20Method,becomes%20first%20starts%20to%20diminish.

K-MEANS



Usually pick the “elbow” point where you see a huge drop indicating cluster homogeneity



The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters

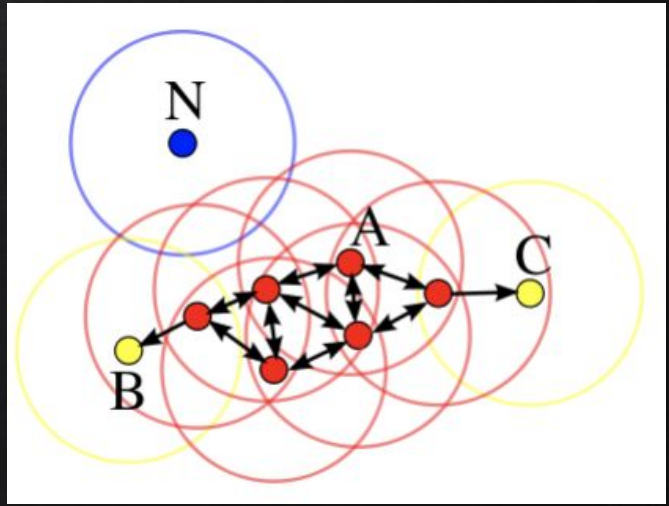
https://towardsdatascience.com/beginners-guide-to-the-three-types-of-machine-learning-3141730ef45d

DBSCAN

- ✘ Motivation: Centroid-based clustering methods like kMeans favor clusters that are spherical, and have great difficulty with anything else.
- ✘ Density-based spatial clustering of applications with noise (DBSCAN) performs density-based clustering, and follows the shape of dense neighborhoods of points

Core points(red) have at least **minPts** neighbors in a sphere of **ε-diameter** around them.
Border point(yellow) have at least 1 core points within ε-diameter
Noise point(blue) have no core points within ε-diameter

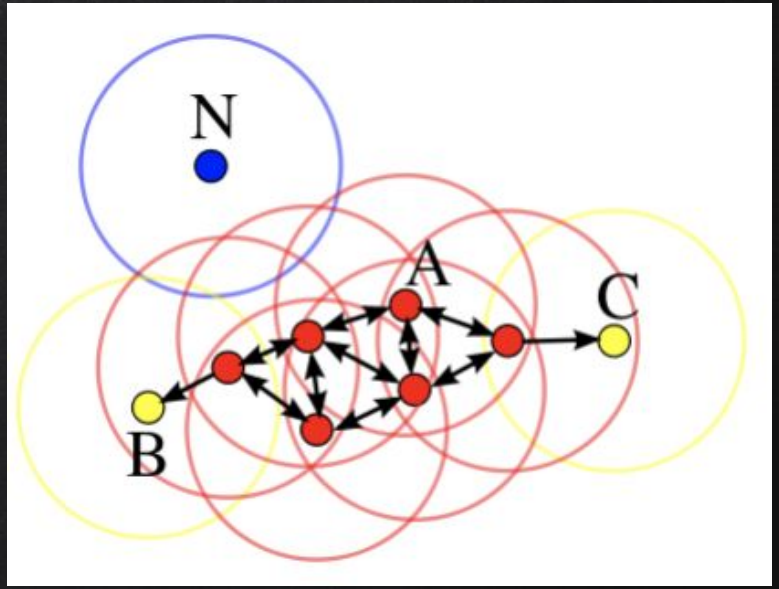
The red points here are core points with at least minPts = 4 neighbors in an ε-sphere around them.



DBSCAN

The DBSCAN algorithm:

1. Find the points in the ε (eps) neighborhood of every point, and identify the core points with more than minPoints neighbors.
2. Find the connected components of core points on the neighbor graph, ignoring all non-core points.
3. Assign each non-core point to a nearby cluster if the cluster is an ε (eps) neighbor, otherwise assign it to noise.



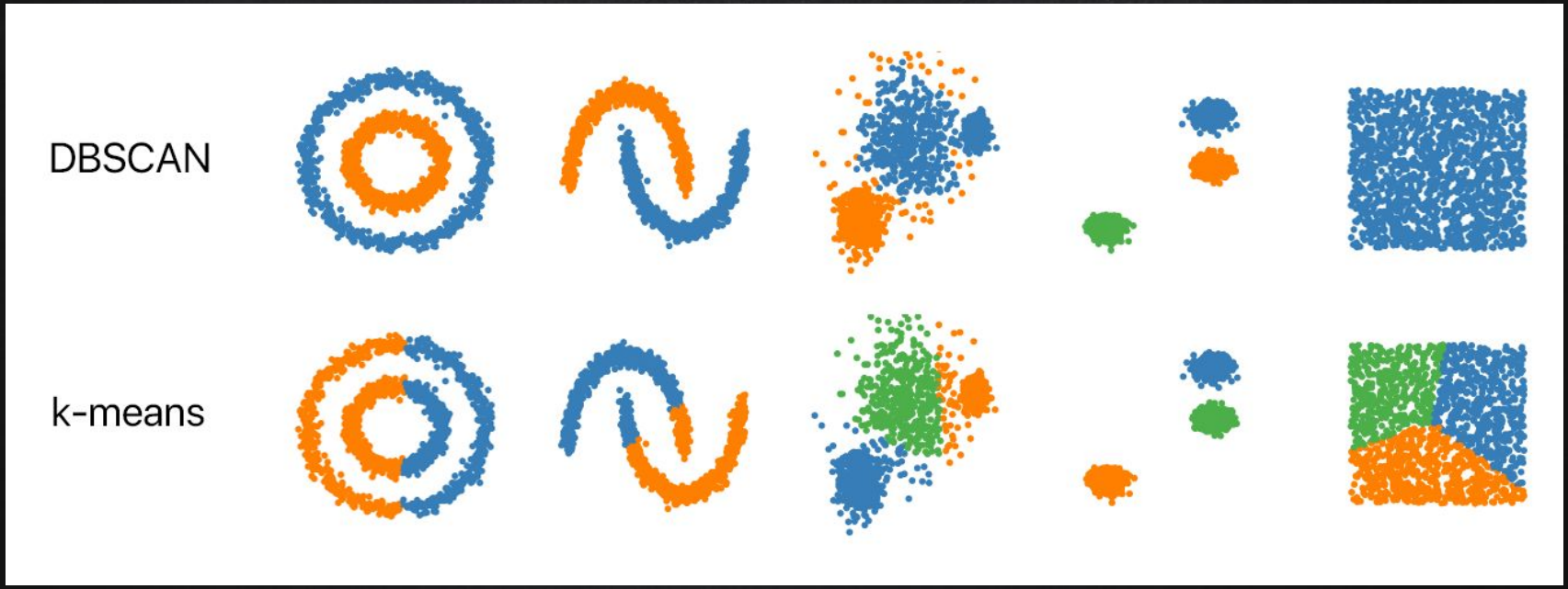
In this diagram, $\text{minPts} = 4$. Point A and the other red points are core points, because the area surrounding these points in an ε radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable.

DBSCAN

- ✘ Key parameters:
  - ε eps: specifies how close points should be to each other to be considered a part of a cluster. Points are neighbours if pointwise distance is below eps
  - minPoints: the minimum number of points to form a dense region. For example, if we set the minPoints parameter as 5, then we need at least 5 points to form a dense reg
  - Distance function
- ✘ Advantages:
  - No pre-specified number of clusters
  - Clusters can be any shape
  - Works well with high density data
  - Robust to outliers (noise component)

https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80

Comparison of two methods



Other clustering methods

Hierarchical clustering

BIRCH

Bi-clustering

Affinity propagation

Key concepts: distance metrics, similarity

Usually clustering algos are sensitive to input scale, might need to normalized

Dimensionality Reduction

Dimensionality Reduction

Dimensionality reduction techniques are widely used and versatile techniques that can be used to:

- find structure in features
- pre-processing for other ML algorithms
- aid in visualisation

The basic principle of dimensionality reduction techniques is to transform the data into a new space that summarise properties of the whole data set along a reduced number of dimensions.

PCA – Principal Component Analysis

When should I use PCA?

1. Do you want to reduce the number of variables, but aren't able to identify variables to completely remove from consideration?

2. Do you want to ensure your variables are independent of one another?

3. Are you comfortable making your independent variables less interpretable?

When your answers to all 3 questions are yes, PCA is safe to use

PCA – Principal Component Analysis

PCA is a technique that transforms the original n-dimensional data into a new n'-dimensional space.

- These new dimensions are linear combinations of the original data, i.e. they are composed of proportions of the original variables.
- Along these new dimensions, called principal components, the data expresses most of its variability along the first PC, then second, …
- Principal components are orthogonal to each other, i.e. non-correlated.

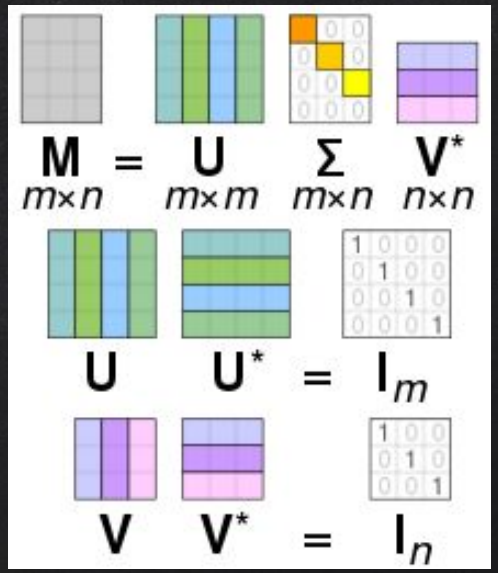
$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$$

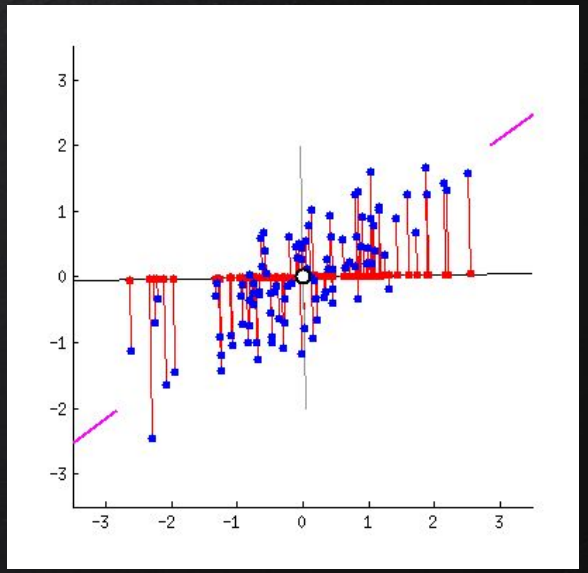
Other linear dim reduction methods: LDA, factor analysis, etc.

https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c

PCA – Principal Component Analysis

PCA is either done by singular value decomposition of a design matrix, or calculating the data covariance matrix of the original data then performing eigenvalue decomposition on the covariance matrix.



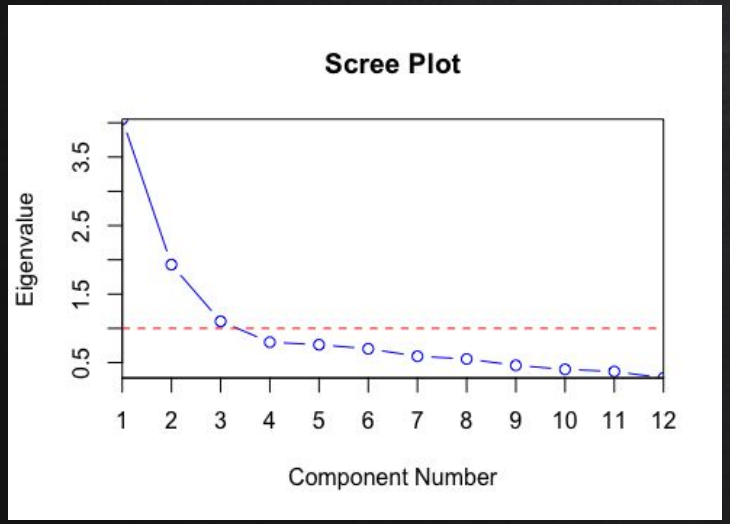


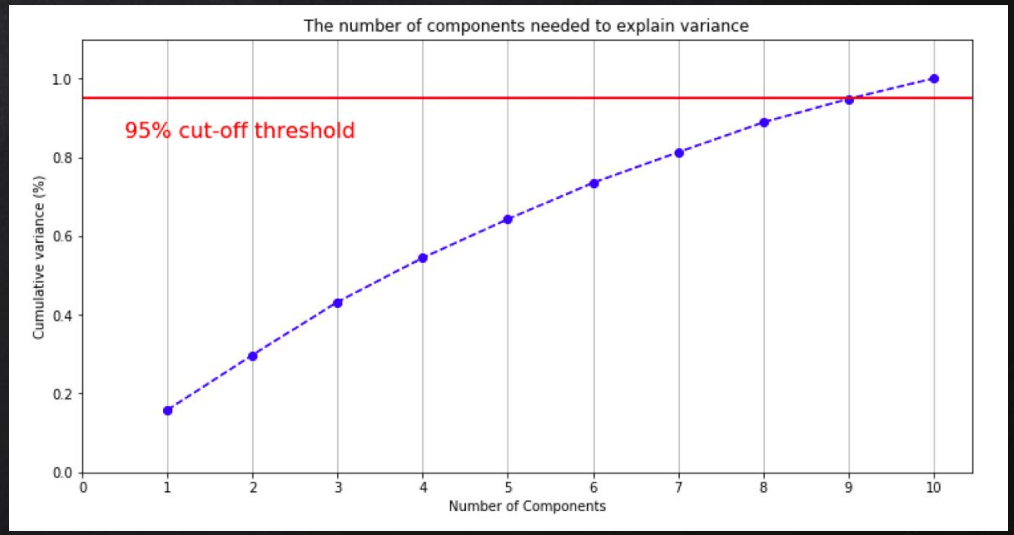
Goal: find a low-dimensional representation of the dataset that preserves as much as possible of the variation

https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

PCA – Principal Component Analysis

- ✘ Input needs standardization (scale between 0 and 1)
- ✘ How to select # of components
  - Scree plot – find the elbow
  - Cumulative % of variance explained (90%–95%)





https://www.mikulskibartosz.name/pca-how-to-choose-the-number-of-components/

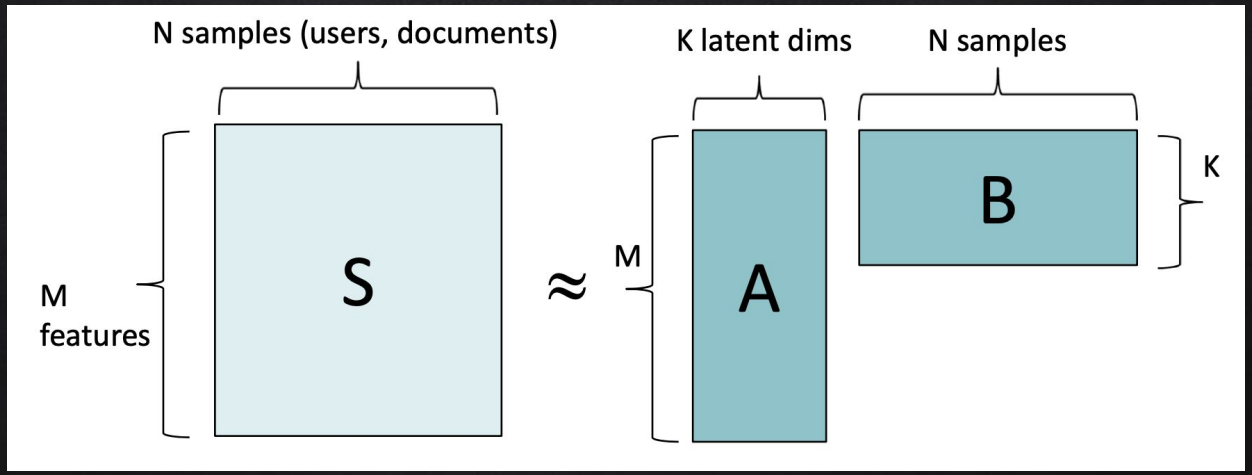
Matrix Factorization

Motivation: "curse of dimensionality"

- ✘ Data dim is nxp, as p grows, model has more d.f. than data
- ✘ Examples of high dimensional data:
  - User preference, topics in documents, etc.
- ✘ It is a popular method for sparse, linear data (ratings, kw/URL combos, CTR on ads).
- ✘ Often used in recommender systems

Matrix Factorization

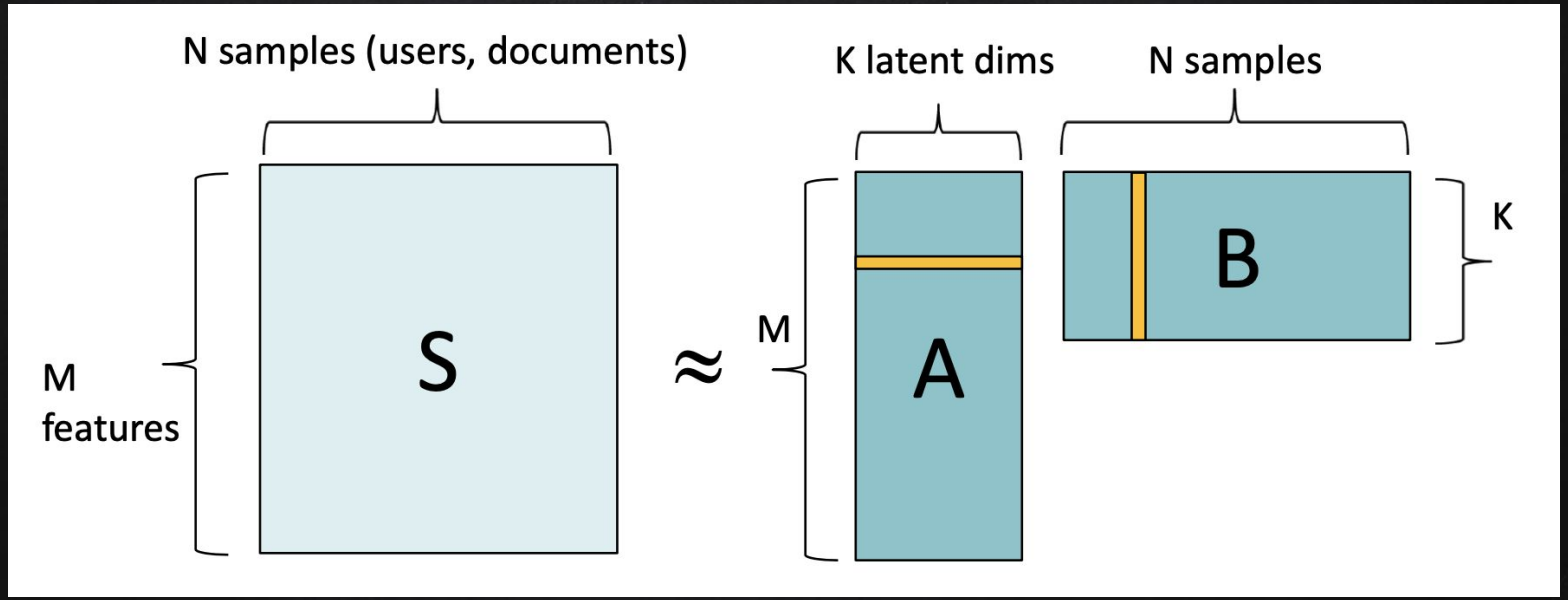
Algebraically, we have a high-dimensional data matrix (typically sparse) S, which we approximate as a product of two dense matrices A and B with low "inner" dimension.



From the factorization, we can fill in missing values of S. This problem is also called "Matrix Completion".

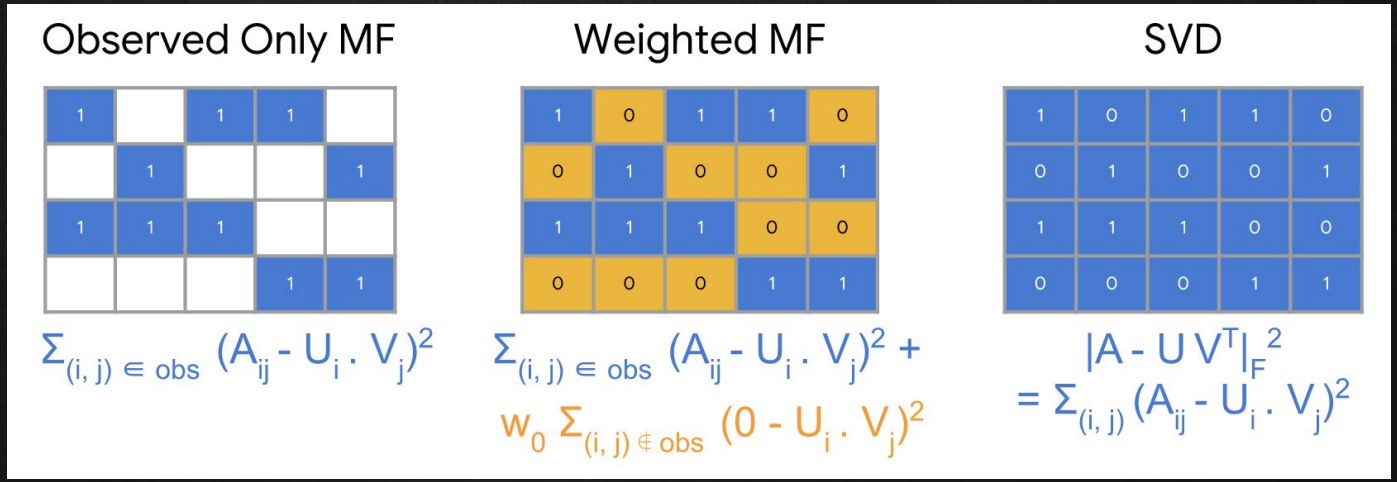
Matrix Factorization

Columns of B represent the distribution of topics the nth sample. Rows of A represent the distribution of topics in the mth feature. These weights allow us to interpret the latent dimensions.



Matrix Factorization

- Input matrix S is sparse, and we approximate it as a low-dimensional product: $A \approx U * V$
- Unknown values of S are usually initialized as 0.



- Loss could be minimized using SGD, Alternating Least Squares (ALS), or other methods.

Other dimensionality reduction methods

Factor analysis

LDA (Linear Discriminant Analysis)

t-SNE

Auto-encoder

One last note on performance

There are two different dimensions of performance:

1. Offline: Model training.

    Plenty of resources. Typical goal is one to few days training time. Best possible model accuracy.

2. Online: Model prediction.

    Limited resources (memory and machines). Goal is usually to minimize latency, and perhaps online model size.

Recommended Readings

https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html#sphx-glr-auto-examples-cluster-plot-kmeans-digits-py

https://www.youtube.com/watch?v=FgakZw6K1QQ

https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

https://developers.google.com/machine-learning/recommendation/collaborative/matrix

Reference

https://bcourses.berkeley.edu/courses/1377158/files/62044813/download?verifier=U8wZ1kcGxU7ZqpRQ7f0A28UP2uhjTZ4EzOKgWJ58&wrap=1

https://towardsdatascience.com/introduction-to-unsupervised-learning-8f1b189e9050

https://towardsdatascience.com/dimensionality-reduction-for-machine-learning-80a46c2ebb7e

https://developers.google.com/machine-learning/recommendation/collaborative/matrix