CSCI 556 Data Analysis & Visualization

SVM, Feature Selection, Data Projection

Instructor: Dr. Jinoh Kim

Topics

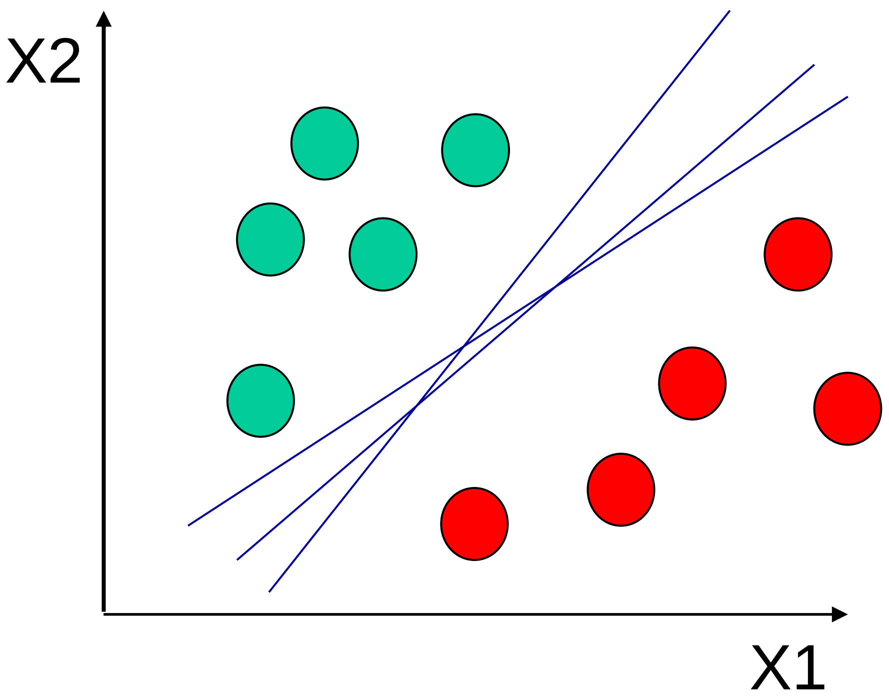
- Support vector machine (SVM)
- Feature selection (attribute selection)
  - Can we use a subset of features (instead of using the entire features)?
  - Scheme-independent and scheme-specific
- Projections
  - Transforming data into a lower-dimensional space
  - principal component analysis (PCA), autoencoder
  - Visualization using t-SNE and UMAP

Support vector machines (SVMs)

- Algorithms for learning linear classifiers
- Finds a special kind of linear model: the maximum margin hyperplane
  - Hyperplane is a subspace whose dimension is one less than that of its ambient space
- Resilient to overfitting because they learn a particular linear decision boundary (Maximum margin hyperplane)
- Can use for non-linear classification
  - Non-linear transformation: mapping data instances to a higher dimension where it is linearly separable

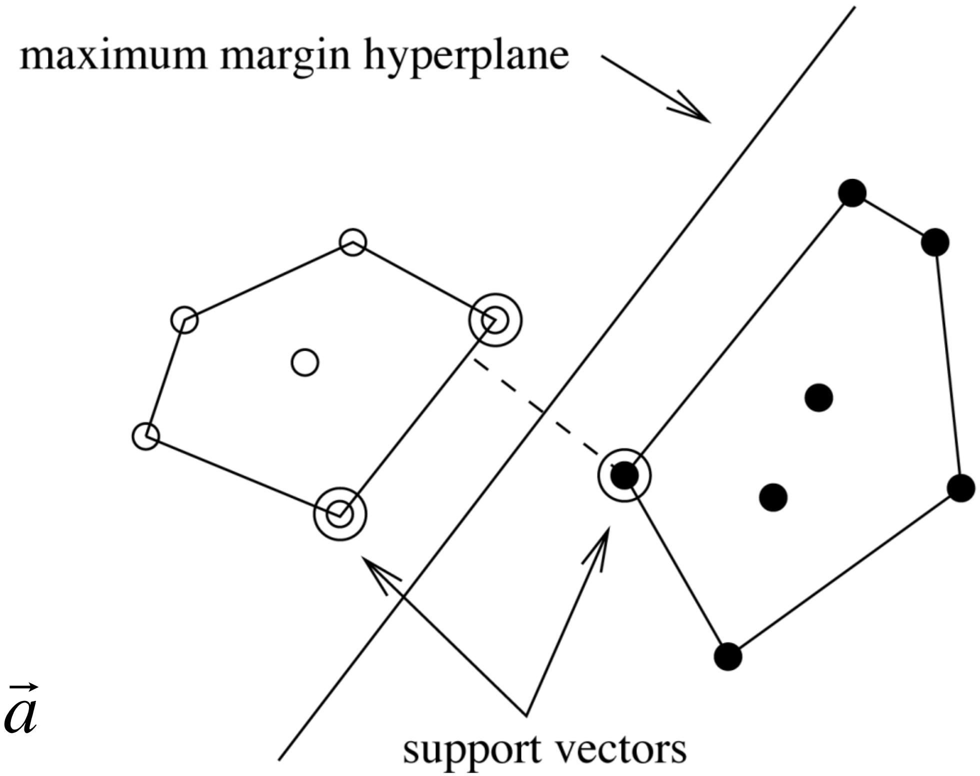
Linearly separable

- Two variables X1 and X2



Support vectors

- The support vectors define the maximum margin hyperplane
- All other instances can be deleted without changing its position and orientation



- Assume two attributes of a1 and a2
- The hyperplane:

$$x = w_0 + w_1 a_1 + w_2 a_2$$

- The maximum margin hyperplane can be written as:

$$x = b + \sum_{i \text{ is a supp. vector}} \alpha_i y_i \vec{a}(i) \cdot \vec{a}$$

Support vectors example

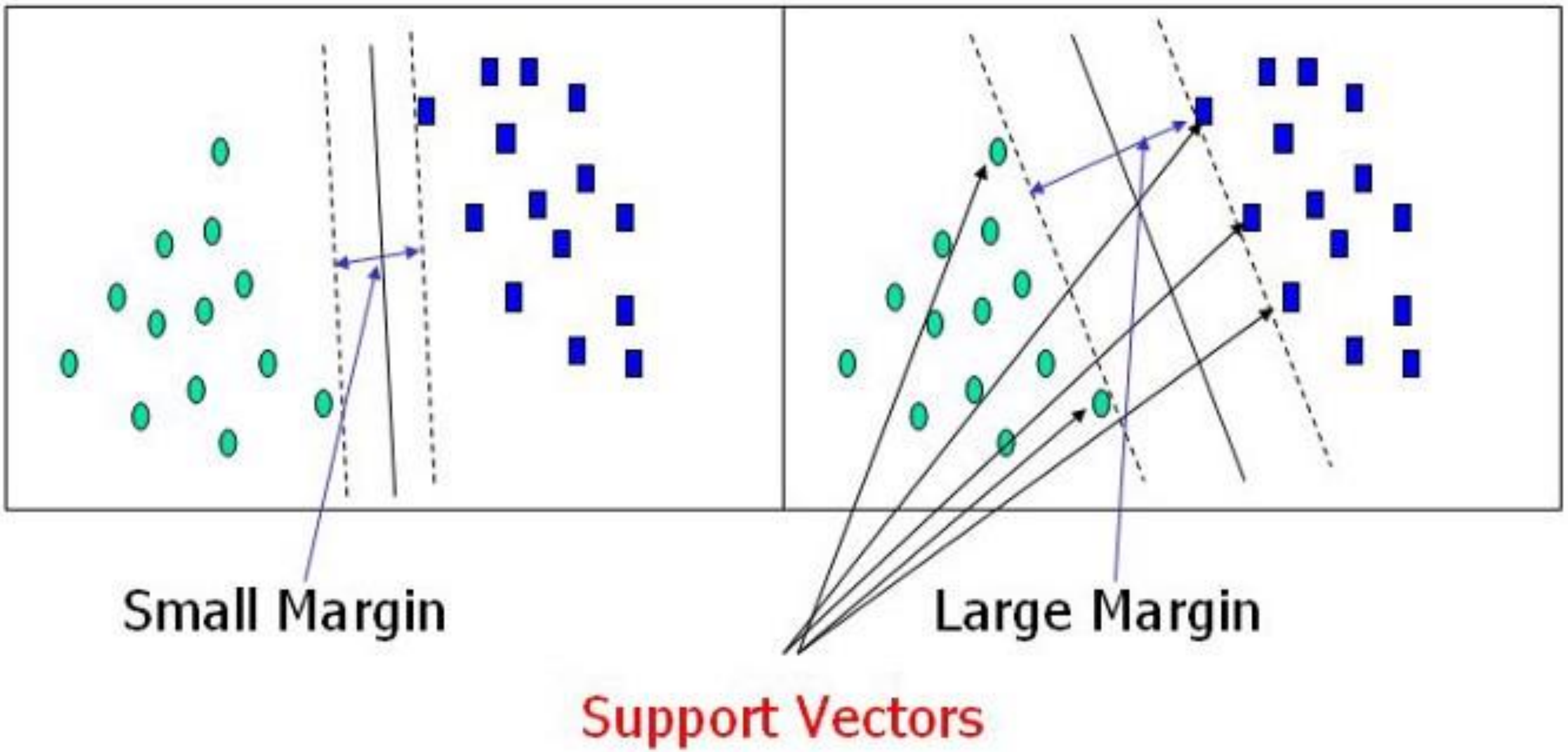


Image from towardsdatascience.com

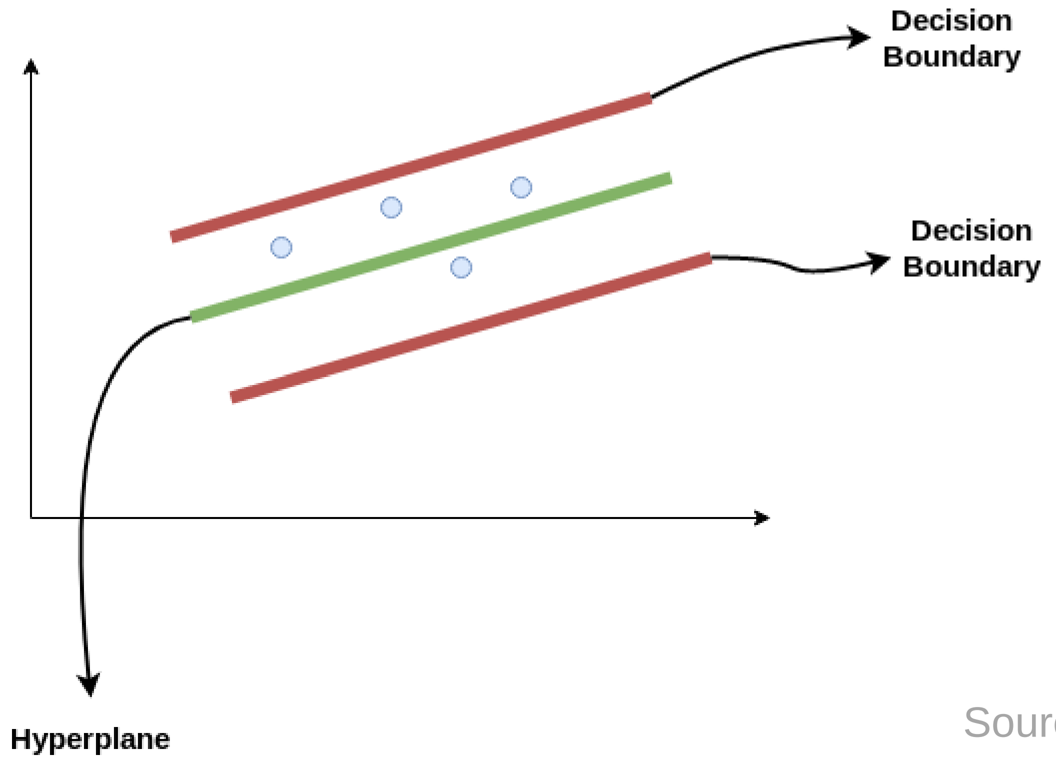
Non-linear SVMs

- We can create a non-linear classifier by creating new "pseudo" attributes from the original attributes in the data
  - "Pseudo" attributes represent attribute combinations
  - E.g.: all polynomials of degree 2 that can be formed from the original attributes
- We can learn a linear SVM from this extended data
- The linear SVM in the extended space is a non-linear classifier in the original attribute space
- Overfitting often not a significant problem with this approach because the maximum margin hyperplane is stable
  - There are often comparatively few support vectors relative to the size of the training set
- Computation time still an issue
  - Each time the dot product is computed, all the "pseudo attributes" must be included

Support vector regression (SVR)

- Maximum margin hyperplane only applies to classification
  - However, idea of support vectors and kernel functions can be used for regression
- Basic method is the same as in linear regression: want to minimize error
- Difference: ignore errors smaller than ε and use absolute error instead of squared error
- User-specified parameter ε defines decision boundary

SVR Example



ε = width between hyperplane and decision boundary

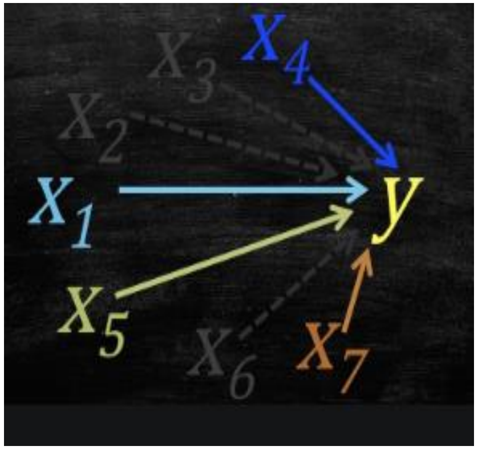
Source: analyticsvidhya.com

Aim: decide a decision boundary at ε distance from the original hyperplane such that data points closest to the hyperplane or the support vectors are within that boundary line.

Feature Selection

- A.k.a feature selection
  - Use "attribute" and "feature" interchangeably
- Selection of attributes to be included in the learning model

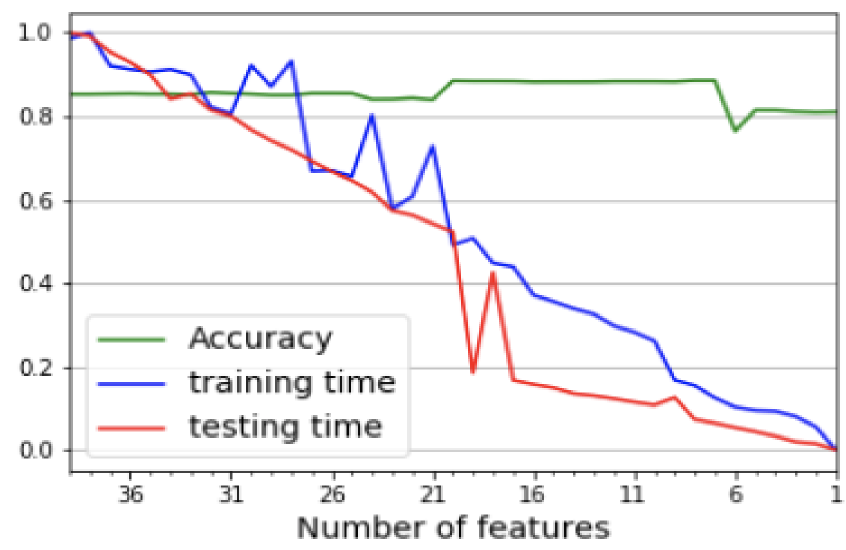
- Example: Student data
  - Features: height, weight, address, hours studied, previous exam grade
  - To predict if a student will pass the exam, most likely height, weight features are less important than previous grade and how many hours study



Why feature selection is important?

- Chance to improve accuracy
  - Eliminating misleading data may enhance the accuracy
- Reduce time
  - Fewer number of attributes may reduce algorithm complexity

- Accuracy slightly decreases with a smaller number of features
- Training and testing time decreases almost linearly across the feature reduction



Feature selection approaches

- Scheme-independent selection (”filter”)
  - Independent from target ML algorithm
- Scheme-specific selection (”wrapper”)
  - Dependent on target ML algorithm

Scheme-independent feature selection

- Filter approach to attribute selection: assess attributes based on general characteristics of the data
- The attributes are selected in a manner that is independent of the target machine learning scheme
- Features can be reduced based on distance, consistency, similarity, and statistical measures
- E.g., Pearson's Correlation, Chi-square, information gain

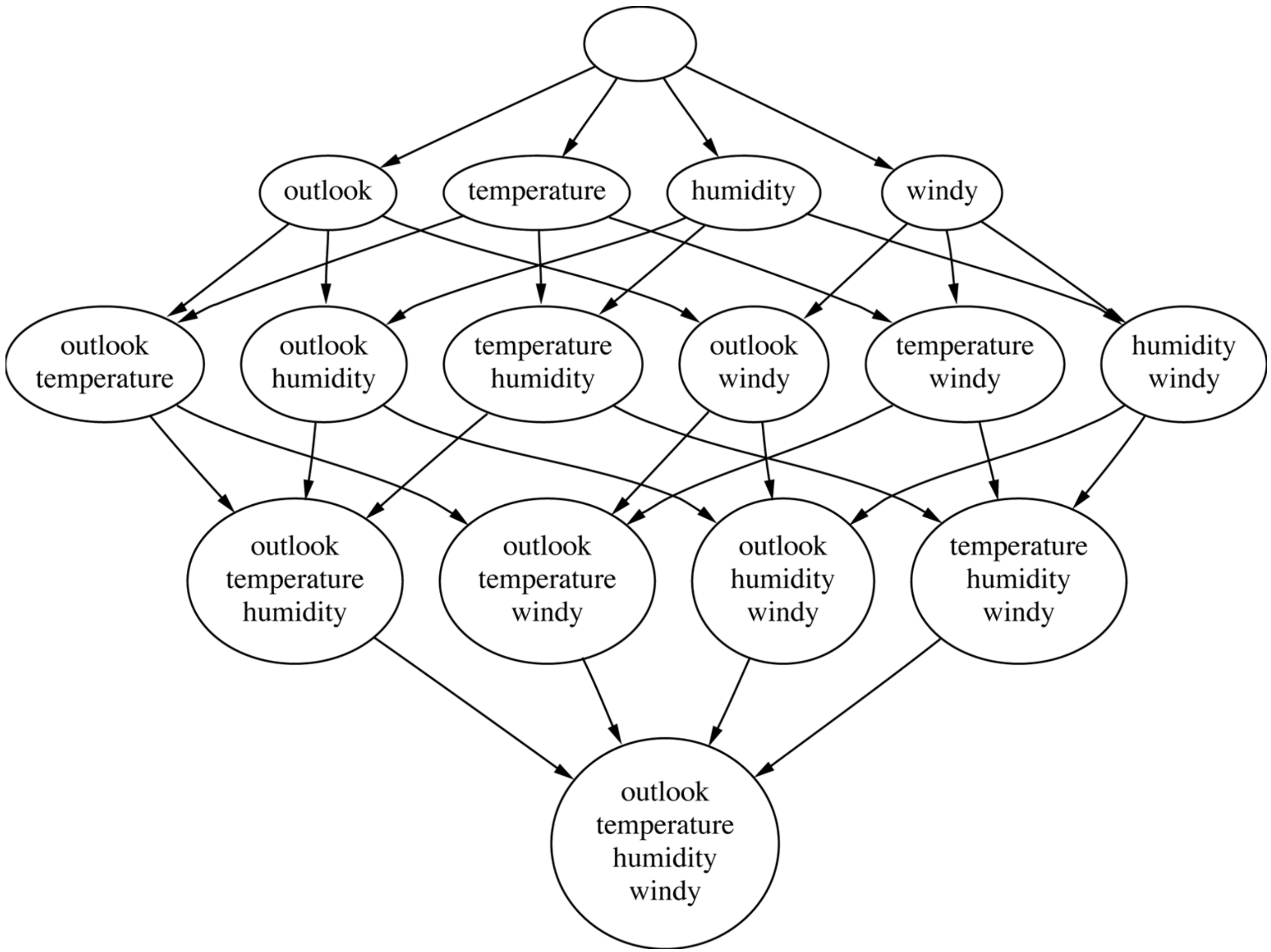
Scheme-specific selection

- Wrapper approach to attribute selection: attributes are selected with target scheme in the loop
- Implement "wrapper" around learning scheme

  Evaluation criterion: cross-validation performance
- In the wrapper method, it is decided to add or remove features to/from the feature subset
  - Hence, the problem is reduced to a search problem: top-down, bottom-up, etc
  - Time consuming and computationally expensive

Attribute subsets for weather data

- Number of attribute subsets is exponential in the number of attributes



Searching the attribute space

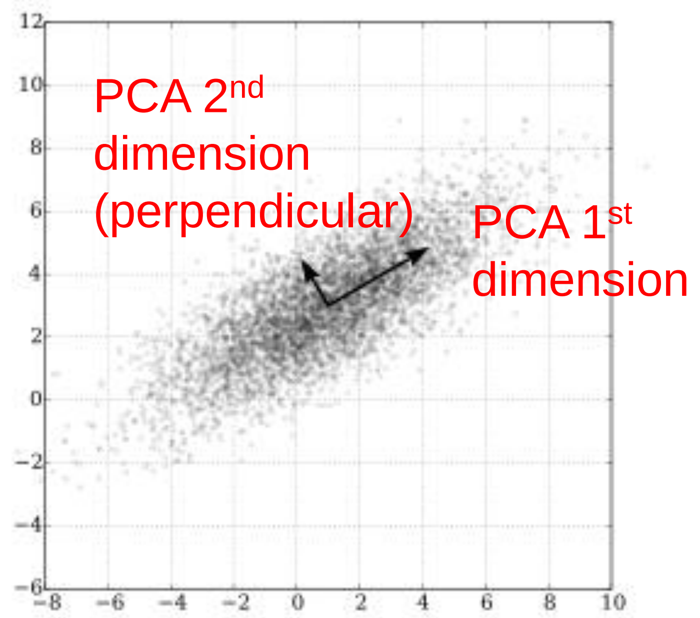
- Number of attribute subsets is exponential in the number of attributes
- Common greedy approaches:
  - forward selection: searches in a downward direction in the attribute space (thus added)
  - backward elimination: upward direction in the attribute space (thus eliminated)

Projections and dimensionality reduction

- Simple transformations can often make a large difference in performance
- Data projection: A kind of function of mapping that transforms data in some ways.
- Dimensionality reduction: transformation of data from a high-dimensional space into a low-dimensional space
  - low-dimensional representation retains some meaningful properties of the original data
  - Example: Principal Component Analysis (PCA), autoencoder (using neural networks), etc
- Curse of dimensionality: problem caused by the exponential increase in complexity associated with adding extra dimensions

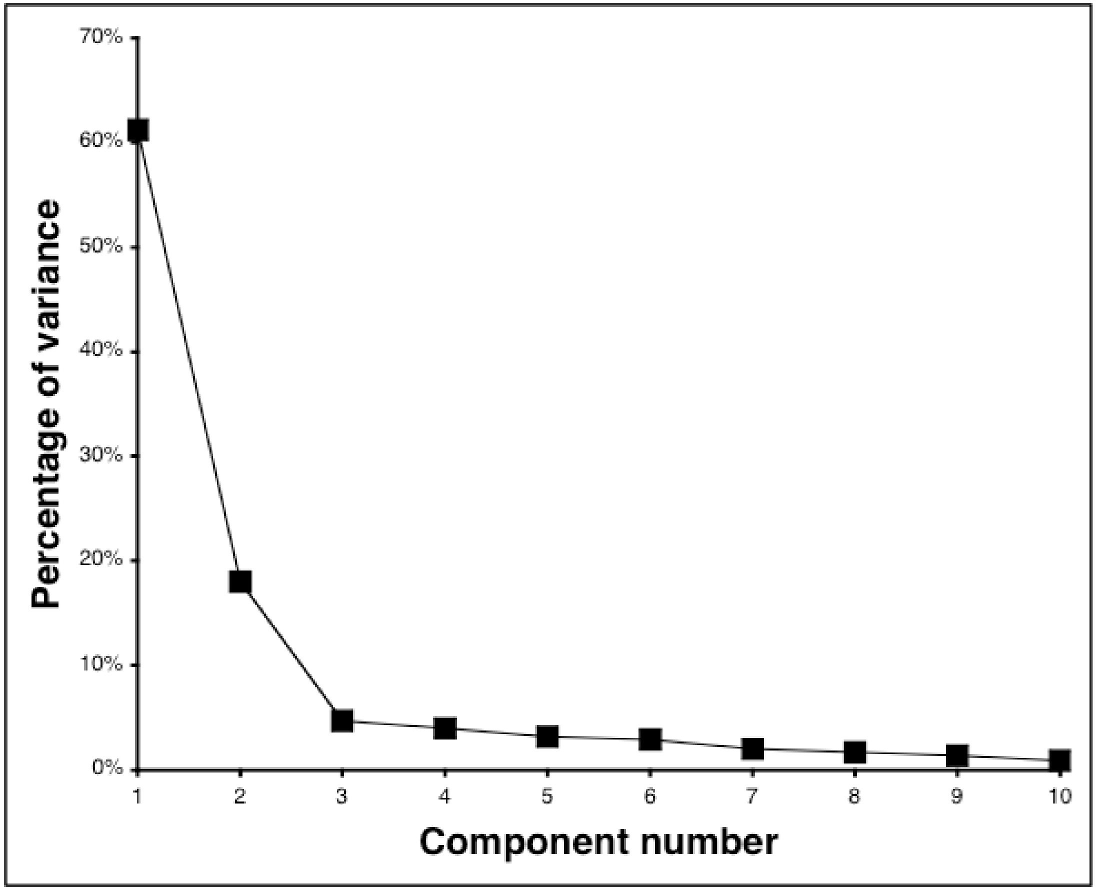
Principal component analysis

- PCA is a method for *dimensionality reduction*
  - Unsupervised method for identifying the important directions in a dataset
  - We can then rotate the data into the (reduced) coordinate system that is given by those directions
- Algorithm:
  1. Find direction (axis) of greatest variance
  2. Find direction of greatest variance that is perpendicular to previous direction and repeat
- Implementation: find eigenvectors of the covariance matrix of the data
  - Eigenvectors (sorted by eigenvalues) are the directions



Example: 10-dimensional data

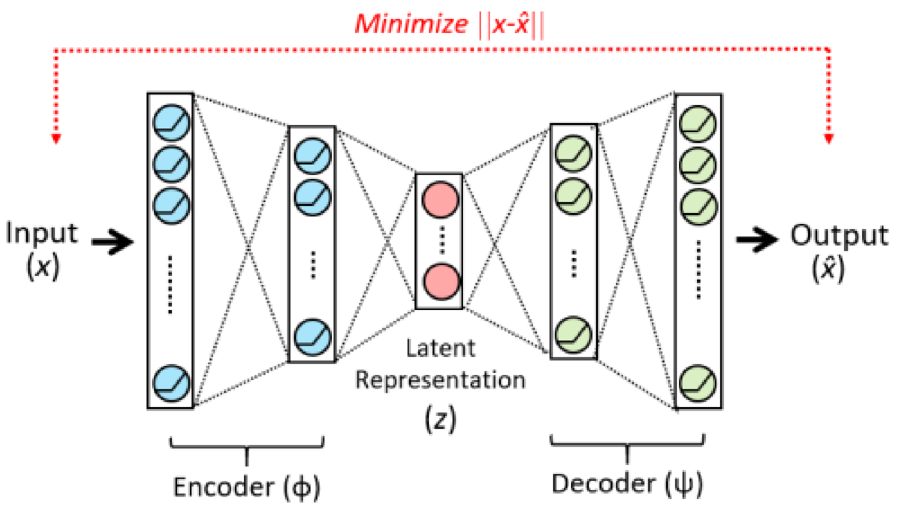
Axis	Variance	Cumulative
1	61.2%	61.2%
2	18.0%	79.2%
3	4.7%	83.9%
4	4.0%	87.9%
5	3.2%	91.1%
6	2.9%	94.0%
7	2.0%	96.0%
8	1.7%	97.7%
9	1.4%	99.1%
10	0.9%	100.0%



- Data is normally standardized or mean-centered for PCA
- In the table, three principal components account for 83.9% of the variance in the dataset

AutoEncoder (AE)

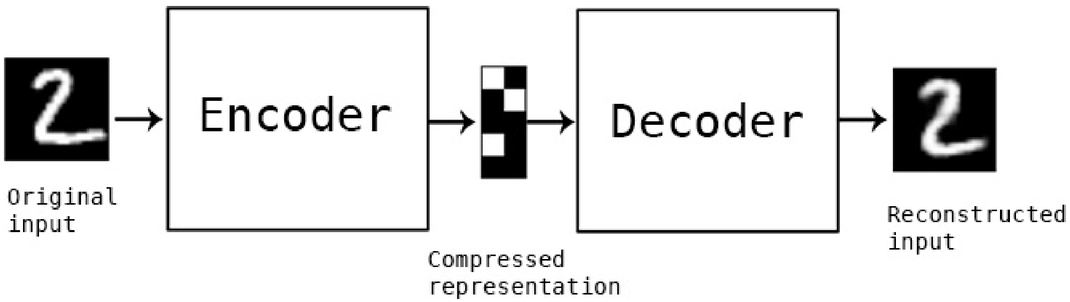
- A type of neural network to learn efficient coding of unlabeled data
- Components: encoder and decoder
  - Encoder compresses and decoder decompresses



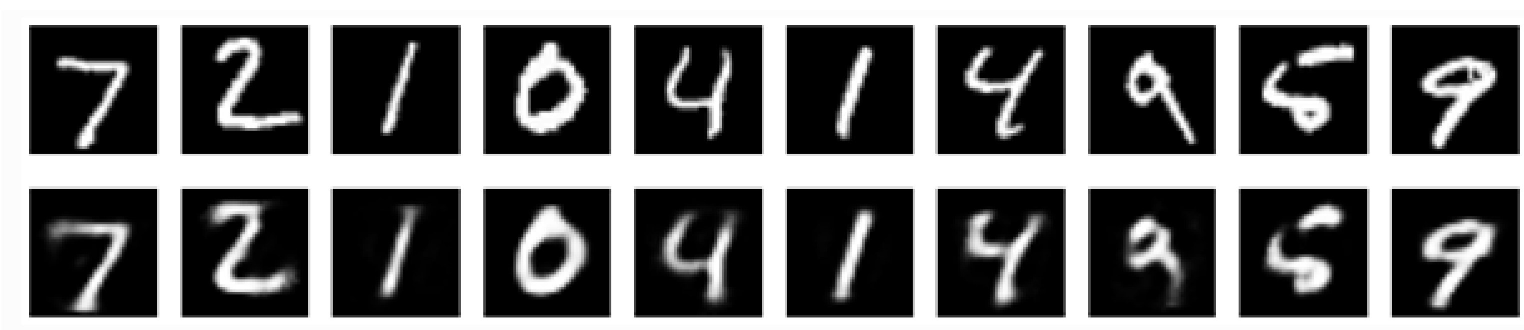
- Goal: minimize reconstruction error
- Reconstruction error = || input – output ||

Autoencoder example

- Autoencoder to learn handwritten digits (MNIST)



- Top row: original digits
- Bottom row: reconstructed digits



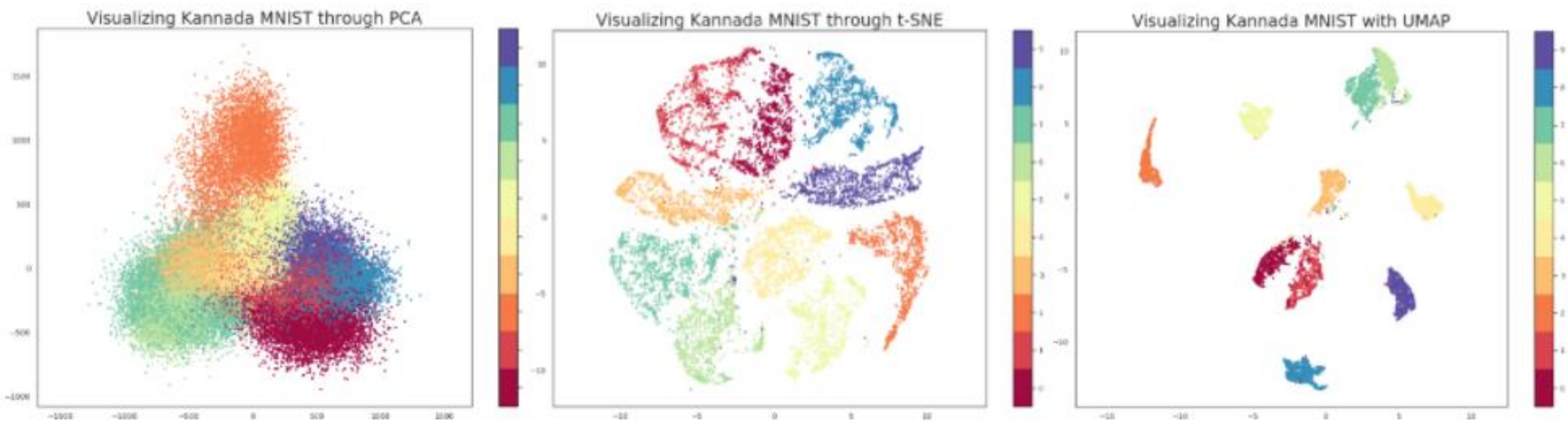
Source: https://blog.keras.io/building-autoencoders-in-keras.html

Visualization

- Dimensionality reduction is essential for visualizing data
- Definitely can use PCA and autoencoder
- t-SNE (t-distributed Stochastic Neighbor Embedding)
  - Aims to solve the problem of PCA
  - Non-linear scaling to represent
- UMAP (Uniform Manifold Approximation and Projection for Dimension Reduction)
  - Similar but quicker than t-SNE

Visualization example

- Visualize MNIST data using PCA, t-SNE, UMAP
- MNIST data contains hand-written digits from ‘0’ to ‘9’



Source: https://www.kaggle.com/parulpandey/part3-visualising-kannada-mnist-with-umap

Summary

- Support vector machine
- Feature selection
  - Scheme-independent (filter) and scheme-specific (wrapper)
- Projections
  - principal component analysis (PCA), autoencoder
  - Visualization using t-SNE and UMAP