INTRODUCTION TO MACHINE LEARNING

AARHUS UNIVERSITY
DEPARTMENT OF BIOSCIENCE

XIAODONG DUAN
POSTDOC

OUTLINE

1. Introduction
2. Basics of linear algebra
3. SVM
4. PCA
5. K-means clustering

AARHUS UNIVERSITY
DEPARTMENT OF BIOSCIENCE

9 MAY 2019

XIAODONG DUAN
POSTDOC

BASICS OF LINEAR ALGEBRA

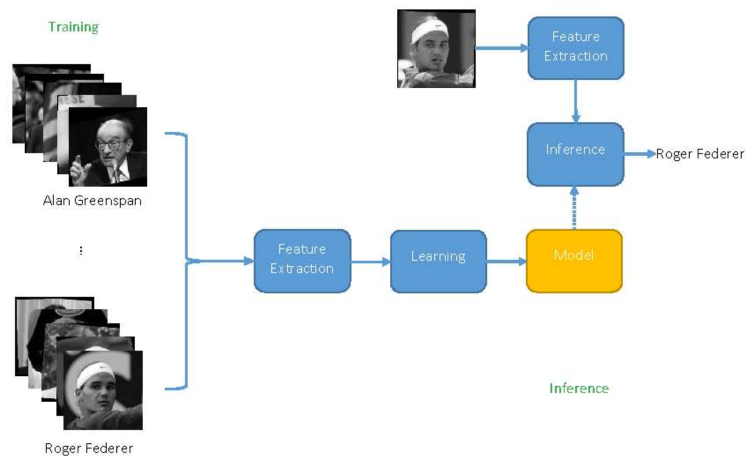
1. Scalar, Vector, Matrix
2. Matrix-Scalar multiplication
3. Matrix-Vector multiplication
4. Matrix-Matrix addition, subtraction and multiplication
   - Vector-Vector multiplication
5. Transpose

MACHINE LEARNING

1. Machine learning is a technology for computers to learn how to perform a task from data without the need of thoroughly understanding it.

2. Components:
   - Data
   - Model
   - Training methods
   - Inference methods

AN EXAMPLE



Face images are from LFW: http://vis-www.cs.umass.edu/lfw/

CATEGORIES OF MACHINE LEARNING

1. Supervised learning
   - Classification
   - Regression
2. Unsupervised learning
   - Dimensionality reduction
   - Clustering

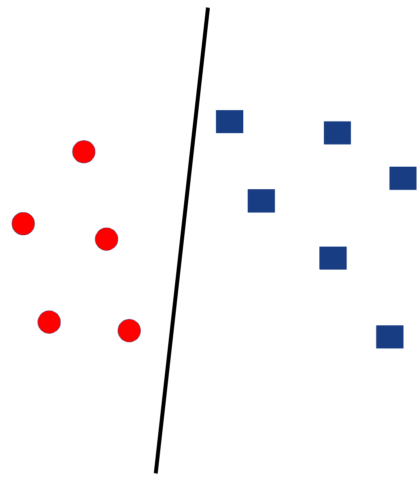
PIPELINE

1. Gather data.
2. Annotate data if needed.
3. Separate the data into training, validation and testing subsets.
4. Define the feature for representation of the data.
5. Choose models.
6. Train the model using the training data.
7. Choose the proper model or hyper-parameters using validation data.
8. Test the chosen model using the testing data.

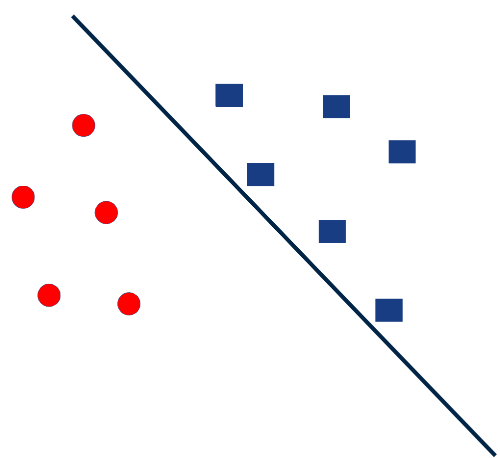
SUPPORT VECTOR MACHINE (SVM)

1. Supervised learning
2. Construct a hyperplane to do classification task
3. Training set
   - $\{\boldsymbol{x}_i, y_i\}, i = 1, \ldots, N$
   - $y_i \in \{-1, 1\}$
4. Hyperplane
   - $\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b} = 0$

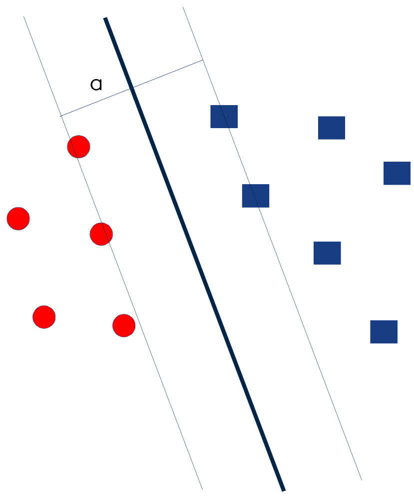
SVM



SVM



SVM



SVM

1. Objective function
   - $\min_{w} \frac{1}{2} \|\boldsymbol{w}\|^2$
   - s.t. $y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 \geq 0, i = 1, \ldots, N$

2. Decision function
   - $f(\boldsymbol{x}) = \sum_{i=1}^{N} a_i \, y_i \boldsymbol{x}^T \boldsymbol{x}_i + b$

SVM

Kernel trick

- We map the feature into space with increased dimension, when the data are not separable.
- Mapping function: $\varphi(\boldsymbol{x})$
- With kernel function, we do not need to calculate $\varphi(\boldsymbol{x})$. Instead, through the kernel function, we can get the dot product between $\varphi(x_i)$ and $\varphi(x_j)$, which is needed for the decision function.

PRINCIPAL COMPONENT ANALYSIS (PCA)

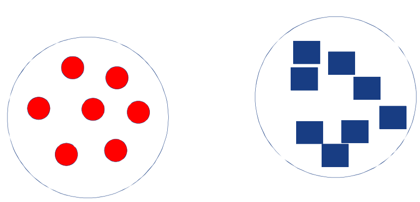
1. Project the original feature into a lower dimensional space
   - Speeding up.
   - Noise reduction.
2. Linear mapping: $\boldsymbol{x}_n = \boldsymbol{A}\boldsymbol{x}$
3. Maximizing the variance of the projected data.

PCA

1. Training data
   - $\boldsymbol{x}_n, n = 1, \ldots, N$
2. Form the covariance matrix for the training data
   - $\boldsymbol{C} = \frac{1}{N}\sum_{n=1}^{N}\{(\boldsymbol{x}_n - \boldsymbol{m})(\boldsymbol{x}_n - \boldsymbol{m})^T\}$
3. Calculate the eigenvectors for the covariance matrix
   - $\boldsymbol{Cw} = \lambda\boldsymbol{C}$
4. Rank the eigenvectors based the corresponding eigenvalues.
5. Use the first K eigenvectors to form **A**, then we can map the feature to a K dimensional space.

K-MEANS CLUSTERING

1. Form K reference vectors that best represent the data.
2. The data can be segmented into K clusters based on the K reference vectors.
3. Unsupervised learning.



K-MEANS CLUSTERING

1. Training data
   - $\boldsymbol{x}_n, n = 1, \ldots, N$
2. Randomly choose K data point to initialize the reference vectors $\boldsymbol{m}_k, k = 1, \ldots, K$
3. Initialize a K-dimension vector $\boldsymbol{b}_n$ for each $\boldsymbol{x}_n$
4. For each $\boldsymbol{x}_n$
   - $\boldsymbol{b}_n^i = \begin{cases} 1 & if\ \|\boldsymbol{x}_n - \boldsymbol{m}_i\| = \min_k \|\boldsymbol{x}_n - \boldsymbol{m}_k\| \\ 0 & else \end{cases}$
5. For each $\boldsymbol{m}_k$
   - $\boldsymbol{m}_k = \sum_n \boldsymbol{b}_n^k \boldsymbol{x}_n / \sum_n \boldsymbol{b}_n^k$
6. If $\boldsymbol{m}_k$ converges, stop. Otherwise, go to 4.

RESOURCES

1. Books:
   - Bishop, Christopher M. *Pattern recognition and machine learning*. springer, 2006.
   - Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
2. Softwares:
   - Python
   - scikit-learn, tensorflow, keras, CNTK, PyTorch, Github
3. Papers:
   - ICML, NeurIPS(NIPS), ICLR, CVPR, ICCV, ECCV, IJCAI, AAAI, arXiv

AARHUS
UNIVERSITY