CST8506 ADVANCED MACHINE LEARNING

Week 1

Data Preprocessing

Professor: Dr. Anu Thomas
Email: thomasa@algonquincollege.com
Office: T315

Agenda

- Recap – Machine Learning
- Feature Selection vs Feature Extraction
- Dimensionality Reduction
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)

Recap – Machine Learning

- CRISP-DM: CRoss-Industry Standard Process for Data Mining
- Learning
  - Supervised Learning
    - Classification – kNN, Decision Tree, Random Forest, Logistic Regression
    - Regression – Simple, multiple, multivariate
  - Unsupervised Learning
    - Clustering - kMeans
    - Outlier Detection – Local Outlier Factor, Isolation Forest

CRISP-DM

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives	Collect Initial Data	Select Data	Select Modeling Technique	Evaluate Results	Plan Deployment
Assess Situation	Describe Data	Clean Data	Generate Test Design	Review Process	Plan Monitering & Maintenance
Determine Data Mining Goals	Explore Data	Construct Data	Build Model	Determine Next Steps	Produce Final Report
Produce Project Plan	Verify Data Quality	Integrate Data	Assess Model		Review Project
		Format Data			

Preprocessing

- Data cleaning – handling missing & duplicate data, handling noise etc.
- Data integration – Combine data from multiple sources
- Data transformation
- Data reduction
  - Dimensionality reduction

Preprocessing (Contd.)

- Data transformation (Format data)
  - Normalization: change a continuous feature to fall within 0 and 1
  - Range Normalization: change a continuous feature to fall within a range
  - Standardization: Rescales data to have a mean of 0 and SD of 1
  - Binning: converting a continuous feature into a categorical feature.
    - equal-width binning - splits the range of the feature values into b bins each of size $\frac{range}{b}$
    - Equal-frequency binning - first sorts values into ascending order and then places an equal number of instances into each bin
  - Sampling – top sampling, random sampling, stratified sampling

Dimensionality Reduction (DR)

- Objective:
  - Reduce the number of features while retaining essential information
  - Solve the problem in low dimensions

Drawbacks of High dimensionality

- Time consuming
- High memory consumption
- Complex models
- Hard to create visualizations
- Curse of dimensionality
  - too many dimensions causes every observation in the dataset to appear equidistant from all the others
  - Distance metrics lose meaning
  - Models require more data to generalize

Types of Dimensionality Reduction

- Feature Selection – keeps a subset of the original features
- Feature Extraction – transforms the data onto a new feature space

Both are used to reduce the number of features – i.e. reduce the number of dimensions → Dimensionality Reduction

Feature Extraction

- Can construct new features by combining existing features
- Reduce dimensionality to $d<k$, where k is the total number of dimensions (features)

How can we extract new features?

Common Approaches for DR

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

Principal Component Analysis (PCA)

- Reduce the number of features in a dataset by preserving as much information as possible (by creating new synthetic features by linearly combining the original features)
- Idea is to trade a little accuracy for simplicity
- Unsupervised technique

How does PCA work?

- Identify the directions in which the data varies the most
- Project the data onto a new set of axes (principal components) aligned with these directions
- Rank the principal components by the amount of variance they explain

How can we do PCA?

- Standardize data
- Calculate covariance matrix to identify correlations
- Find eigen values and vectors to identify principal components
- Create a feature vector to decide which of the principal components to be used (sort eigen vectors by their corresponding eigen values in decreasing order and then select the top k eigen vectors)
- Recast the data along the principal components' axes

Step 1 – Standardize data

- Standardize the data by transforming the features to have mean of 0 and SD of 1
- Each feature will contribute equally
- In Python, StandardScaler() will standardize the data

Step 2 – Calculate Covariance Matrix

- To find the correlation between attributes
  - If positive, those variables increase or decrease together
  - If negative, then when one increases, the other decreases

Covariance Matrix of Iris Dataset

Covariance Matrix of Iris Standardized Dataset

Attributes	SL	SW	PL	PW
SL	1.000	-0.109	0.872	0.818
SW	-0.109	1.000	-0.421	-0.357
PL	0.872	-0.421	1.000	0.963
PW	0.818	-0.357	0.963	1.000

Eigen Values and Eigen Vectors

- Eigen Vectors: direction of the axes where there is the most variance (principal components)
- Eigen Values: coefficients attached to eigen vectors, which give the amount of variance carried in each principal component
- By ranking the eigen vectors in order of their eigen values, highest to lowest, we get the principal components in order of significance

Eigen Values and Eigen Vectors of Iris Dataset

- Eigen Values: [2.94 0.92 0.15 0.02]
- Eigen Vectors:

$$\begin{bmatrix} 0.52 & -0.38 & -0.72 & 0.26 \\ -0.27 & -0.92 & 0.24 & -0.12 \\ 0.58 & -0.02 & 0.14 & -0.80 \\ 0.56 & -0.07 & 0.63 & 0.52 \end{bmatrix}$$

- Variances: [0.73, 0.23, 0.04, 0.005]

$$2.94/(2.94 + 0.92 + 0.15 + 0.02) = 0.73$$

$$0.92/(2.94 + 0.92 + 0.15 + 0.02) = 0.23$$

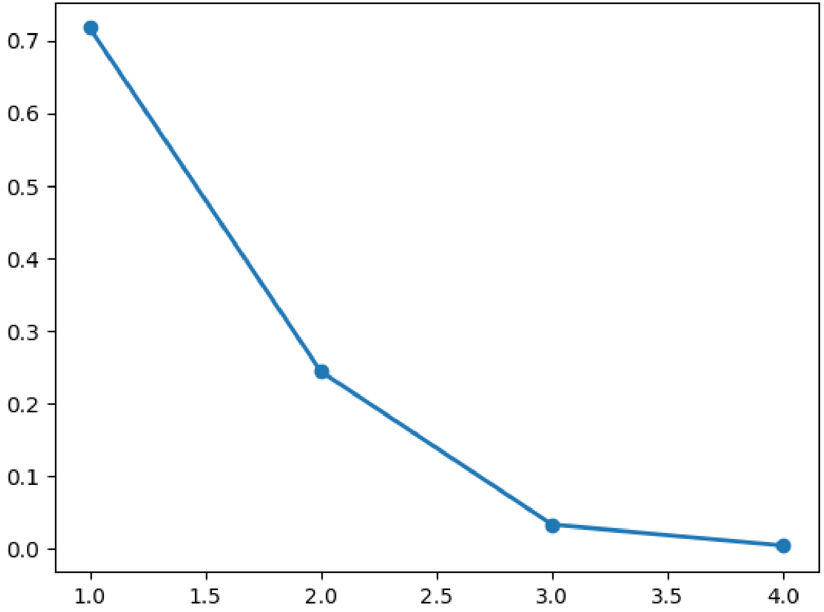
Based on the variances, we can see 96% (73 + 23) of information is compressed in first two principal components

Principal Components

- the new features created as linear combinations of initial features
  - New features will be uncorrelated
  - Maximum possible information will be included in the first component, then the maximum of the remaining will be in the second component and so on.
  - We can discard the components with minimal info

Optimal Number of Principal Components

- Scree Plot
- A line plot of eigen values of principal components
- Here, 3 is the best number



Results – Iris Dataset

With two principal components:

Sum of variance: 0.958

- Confusion Matrix before PCA

  [[50 0 0]
  [0 47 3]
  [0 4 46]]

- Confusion matrix after PCA:

  [[50 0 0]
  [0 44 2]
  [0 6 48]]

- Accuracy before PCA: 95.33%
- Accuracy after PCA: 94.67%

With three principal components:

- Sum of variance: 0.995
- Confusion Matrix before PCA:

  [[50 0 0]
  [0 47 3]
  [0 4 46]]

- Confusion Matrix after PCA:

  [[50 0 0]
  [0 48 2]
  [0 2 48]]

- Accuracy before PCA : 95.33%
- Accuracy after PCA : 97.33%

In-class Activity - Excel

Linear Discriminant Analysis

- Projects a dataset onto a lower-dimensional space by maximizing class-separability
- Similar to PCA, but additionally interested in the axes that maximize the separation between classes
- Supervised technique

How can we do LDA?

- Find the means of various classes of the dataset
- Create new axis such that:
  - Maximize the distance between means
  - Minimize the variation (or the scatter) within each category

How does LDA work?

- Find the d-dimensional mean vectors for the various classes of the dataset
- Calculate the scatter matrices (Between class and Within-class scatter matrix)
- Calculate the eigen vectors and the corresponding eigen values for the scatter matrix
- Sort eigen vectors by their corresponding eigen values in decreasing order and then select the top k eigen vectors to form a d x k matrix
- Use this d x k matrix to transform the samples onto the new subspace

PCA vs LDA

Similarities

Both rank the new axes in the order of importance

- PC1 accounts for the most variation in the data, PC2 will be the next one that holds the maximum of the remaining info and so on
- LD1 accounts for the most variation between the categories, and then LD2 and so on

Differences

PCA	LDA
Unsupervised learning algorithm	Supervised learning algorithm
Finds directions of maximum variance regardless of class labels	Finds directions of maximum class separability
n_components <= min(n_samples, n_features)	n_components <= min(n_classes - 1, n_features)

References

- https://builtin.com/data-science/step-step-explanation-principal-component-analysis
- https://sebastianraschka.com/Articles/2014_python_lda.html