Data Preparation

Lijun Zhang
zlj@nju.edu.cn
http://cs.nju.edu.cn/zlj

Outline

- **Introduction**
- Feature Extraction and Portability
- Data Cleaning
- Data Reduction and Transformation
  - Sampling
  - Feature Subset Selection
  - Dimensionality Reduction with Axis Rotation
  - Dimensionality Reduction with Type Transformation
- Summary

Introduction

"Success depends upon previous preparation, and without such preparation there is sure to be failure."—Confucius

"凡事豫（预）则立，不豫（预）则废"——《礼记·中庸》

- Feature Extraction and Portability
  - Raw logs, documents, semistructured data
  - Data may contain heterogeneous types
- Data Cleaning
  - Missing, Erroneous, and Inconsistent
- Data Reduction, Selection, and Transformation
  - Efficiency, Effectiveness

Outline

- Introduction
- **Feature Extraction and Portability**
- Data Cleaning
- Data Reduction and Transformation
  - Sampling
  - Feature Subset Selection
  - Dimensionality Reduction with Axis Rotation
  - Dimensionality Reduction with Type Transformation
- Summary

Feature Extraction

Domain	Raw Data	Features
Sensor	Low-level signals	Wavelet or Fourier transforms
Image	Pixels	Color histograms Visual words
Web logs	Text strings	IP address Action
Network traffic	Characteristics of the network packets	Number of bytes transferred Network protocol
Document data	Text strings	Bag-of-words Entity extraction

Feature extraction is an art form that is highly dependent on the skill of the analyst

Data Type Portability (1)

Data is Often Heterogeneous

- A demographic data set may contain both numeric and mixed attributes

Possible Solutions

- Designing an algorithm with an arbitrary combination of data types
  - ✓ Time-consuming and sometimes impractical
- Converting between various data types
  - ✓ Utilize off-the-shelf tools for processing

Data Type Portability (2)

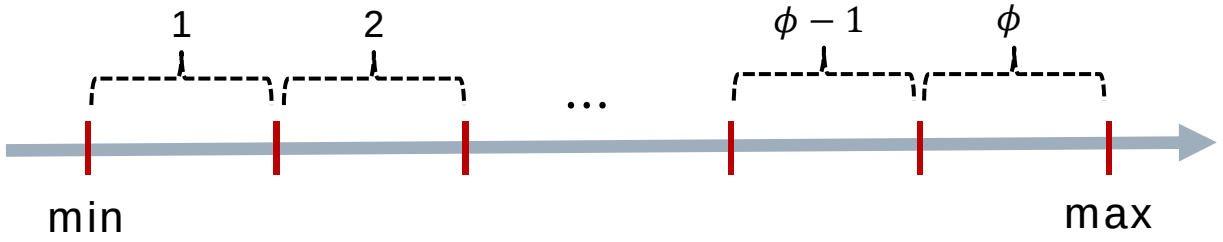
Ways of Transforming Data

Table 2.1: Portability of different data types

Source data type	Destination data type	Methods
Numeric	Categorical	Discretization
Categorical	Numeric	Binarization
Text	Numeric	Latent semantic analysis (LSA)
Time series	Discrete sequence	SAX
Time series	Numeric multidimensional	DWT, DFT
Discrete sequence	Numeric multidimensional	DWT, DFT
Spatial	Numeric multidimensional	2-d DWT
Graphs	Numeric multidimensional	MDS, spectral
Any type	Graphs	Similarity graph (Restricted applicability)

Numeric to Categorical Data: Discretization (1)

- Divides the ranges of the numeric attribute into ϕ ranges



  - Age Attribute
    - ✓ [0, 10], [21, 20], [21, 30] , ...
  - Salary
    - × [0, 10000], [10001, 20000], [20001, 30000], ...

Numeric to Categorical Data: Discretization (2)

- Equi-width Ranges
  - Each range $[a, b]$ is chosen such that $b - a$ is a constant
- Equi-log Ranges
  - Each range $[a, b]$ is chosen such that $\log b - \log a$ is a constant
  - For example, $[1, a]$, $[a, a^2]$, $[a^2, a^3]$, ...
- Equi-depth Ranges
  - Each range has an equal number of records
  - Sorting and Selecting

Categorical to Numeric Data: Binarization

- Two categories
  - $0, 1$ or $-1, 1$

- ϕ categories
  - ϕ-dimensional indicator vector
  - The position of 1 indicates the category

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{array}{l} \longrightarrow \text{ 1st Category} \\ \longrightarrow \text{ 2nd Category} \\ \longrightarrow \text{ 3rd Category} \end{array}$$

$$\phi = 3$$

Text to Numeric Data

- Tokenization, Stop Word Removal, Stemming, Weighting (TF-IDF)

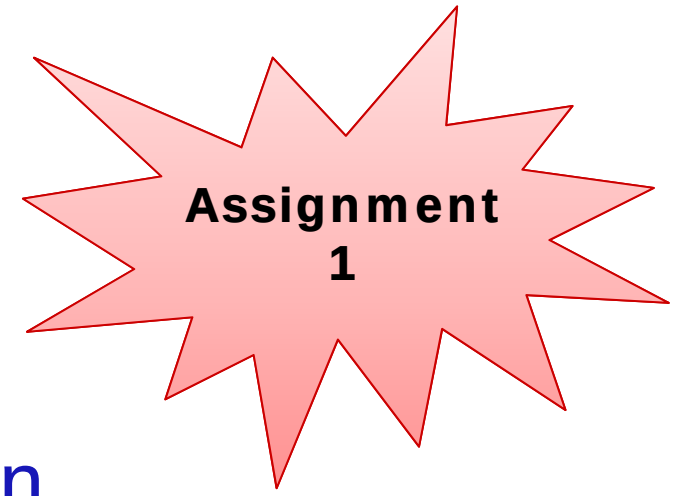
Tokenization for Chinese sentence is difficult.

“生产鞋子和服装”
“今天真热”

Text to Numeric Data

- Tokenization, Stop Word Removal, Stemming, Weighting (TF-IDF)
- Document-Term Matrix

	酒	三国	
将进酒	0.3	0	…
念奴娇·赤壁怀古	0	0.5	…
	…	…	…



- Dimensionality Reduction
  - Latent Semantic Analysis
- Normalization

Time Series to Discrete Sequence Data

- Symbolic Aggregate Approximation (SAX)
  - Window-based averaging
    - ✓ Evaluate the average value in each windows
  - Value-based discretization
    - ✓ Discretize the average value by equi-depth intervals
- How to Ensure Equi-depth?
  - Assume certain distribution, such as Gaussian
  - Estimate the distribution

Time Series to Numeric Data

- Discrete Wavelet Transform (DWT)
- Discrete Fourier transform (DFT)
- Advantages
  - Remove Dependence

Discrete Sequence to Numeric Data

- Discrete sequence to a Set of (binary) Time Series
  - ACACACTGTGACTG (4 Symbols)
  - 10101000001000 (A)
  - 01010100000100 (C)
  - 00000010100010 (T)
  - 00000001010001 (G)
- Map Each of These Time Series into a Multidimensional Vector
- Features from the Different Series are Combined

Any Type to Graphs for Similarity-Based Applications

- A Neighborhood Graph for a Set of n Points $\mathcal{O} = \{O_1, \ldots, O_n\}$
  - A Single Node is defined for each O_i
  - An edge exists between O_i and O_j, if

$$d(O_i, O_j) \leq \epsilon$$

  - The weight W_{ij} of edge (i, j) is defined as

$$W_{ij} = e^{-\frac{d(O_i, O_j)^2}{t^2}}$$

- Many Variants Exist

Other Transformations

- Spatial to Numeric Data
  - Similar to Time-series Data
- Graphs to Numeric Data
  - Multidimensional Scaling (MDS)
    - ✓ Edge represents distance
  - Spectral Transformations
    - ✓ Edge represents similarity

Outline

- Introduction
- Feature Extraction and Portability
- **Data Cleaning**
- Data Reduction and Transformation
  - Sampling
  - Feature Subset Selection
  - Dimensionality Reduction with Axis Rotation
  - Dimensionality Reduction with Type Transformation
- Summary

The Reason of Cleaning

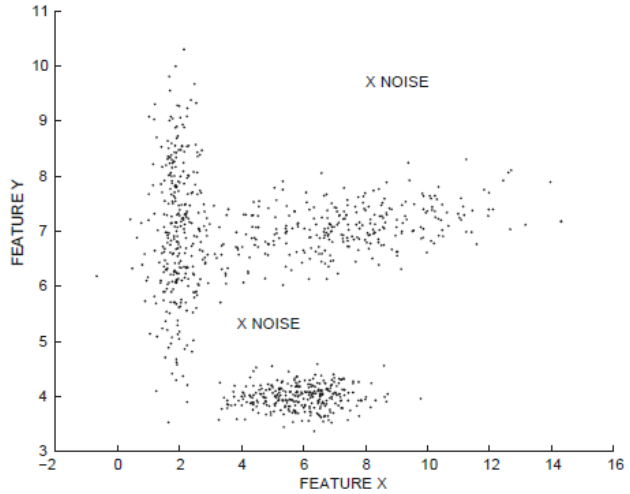
- Data Collection Technologies are Inaccurate
  - Sensors
  - Optical character recognition
  - Speech-to-text data
- Privacy Reasons
- Manual Errors
- Data Collection is Expensive
  - Medical Test

Handling Missing Entries

- Delete the Data Record Containing missing entries
  - What to do if nothing left?
- Estimate or Impute the Missing Values
  - Additional errors may be introduced
  - Good under certain conditions (e.g., Matrix Completion)
- Designing an Algorithm that Works with Missing Data

Handling Incorrect and Inconsistent Entries

- Inconsistency Detection
  - E.g., full name and abbreviation
- Domain Knowledge
  - Age cannot be 800
- Data-centric Methods



Examine before discarding

Scaling and Normalization

- Features have Different Scales
  - Age versus Salary
- Standardization
  - If the j-th attribute has mean μ_j and standard derivation σ_j

$$z_i^j = \frac{x_i^j - \mu_j}{\sigma_j}$$

- Min-Max Scaling
  - Map to $[0,1]$
  - Sensitive to noise

$$z_i^j = \frac{x_i^j - min_j}{max_j - min_j}$$

Outline

- Introduction
- Feature Extraction and Portability
- Data Cleaning
- **Data Reduction and Transformation**
  - Sampling
  - Feature Subset Selection
  - Dimensionality Reduction with Axis Rotation
  - Dimensionality Reduction with Type Transformation
- Summary

Why?

- The Advantages
  - Reduce space complexity
  - Reduce time complexity

  - Reduce noise
  - Reveal hidden structures
    - ✓ E.g., manifold learning

- The Disadvantages
  - Information loss

Outline

- Introduction
- Feature Extraction and Portability
- Data Cleaning
- Data Reduction and Transformation
  - **Sampling**
  - Feature Subset Selection
  - Dimensionality Reduction with Axis Rotation
  - Dimensionality Reduction with Type Transformation
- Summary

Sampling for Static Data

- Unbiased (Uniform) Sampling
  - Sampling without replacement
  - Sampling with replacement
    - ✓ Duplicates are possible
- Biased Sampling
  - Some parts of the data are emphasized
  - E.g., Temporal-decay bias

$$p(\bar{X}) \propto e^{-\lambda \delta t}$$

- Stratified Sampling
  - Partition data into a set of strata
  - Sample in each of stratum

An Example of Sampling

- There are 10000 people which contain 100 millionaires
- Unbiased Sampling 100 people
  - In expectation, one millionaire will be sampled
  - In practice, maybe no millionaires are sampled
- Stratified Sampling
  - Unbiased Sampling 1 from 100 millionaires
  - Unbiased Sampling 99 from remaining

Reservoir Sampling for Data Streams

- The Setting
  - Data arrive sequentially
  - We want sample k of them uniformly
    - ✓ There is a reservoir that can hold k data points
- The Algorithm
  - The first k data points are kept
  - Insert the n-th data point with probability k/n
    - ✓ If the n-th data is inserted, then drop one of the existing k data points uniformly

Reservoir Sampling for Data Streams

- The Setting
  - Data arrive sequentially
  - We want sample k of them uniformly

data

After n stream points have arrived, the probability of any stream point being included in the reservoir is the same, and is equal to k/n.

- T
  -
  - Insert the n-th data point with probability k/n
    - ✓ If the n-th data is inserted, then drop one of the existing k data points uniformly

Outline

- Introduction
- Feature Extraction and Portability
- Data Cleaning
- Data Reduction and Transformation
  - Sampling
  - **Feature Subset Selection**
  - Dimensionality Reduction with Axis Rotation
  - Dimensionality Reduction with Type Transformation
- Summary

Feature Subset Selection

Unsupervised Feature Selection

- Using the performance of unsupervised learning (e.g, clustering) to guide the selection

Supervised Feature Selection

- Using the performance of supervised learning (e.g., classification) to guide the selection

Outline

- Introduction
- Feature Extraction and Portability
- Data Cleaning
- Data Reduction and Transformation
  - Sampling
  - Feature Subset Selection
  - **Dimensionality Reduction with Axis Rotation**
  - Dimensionality Reduction with Type Transformation
- Summary

Dimensionality Reduction with Axis Rotation (1)

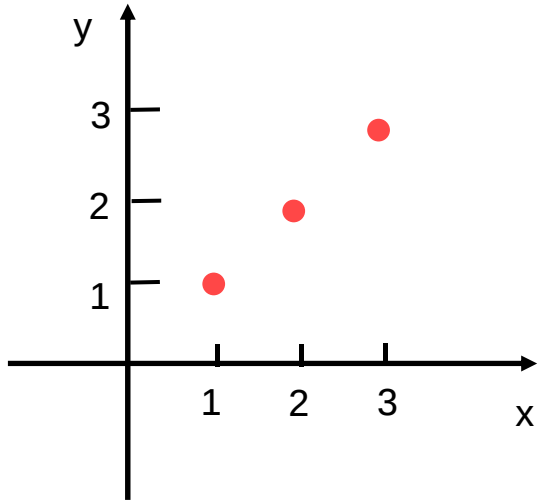
Motivations (Perfect Case)

- Consider the following 3 points in a 2-dimensional space

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$



Dimensionality Reduction with Axis Rotation (2)

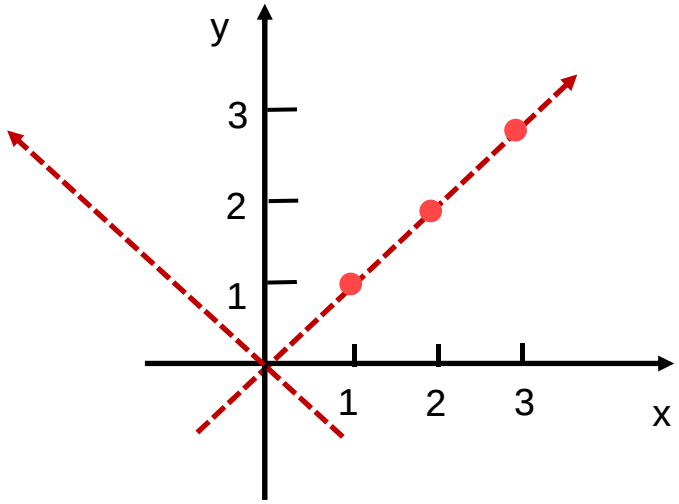
Motivations (Perfect Case)

- What is the new coordinates if we rotate the axis

$$\mathbf{x}_1 = \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2\sqrt{2} \\ 0 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 3\sqrt{2} \\ 0 \end{bmatrix}$$



Dimensionality Reduction with Axis Rotation (2)

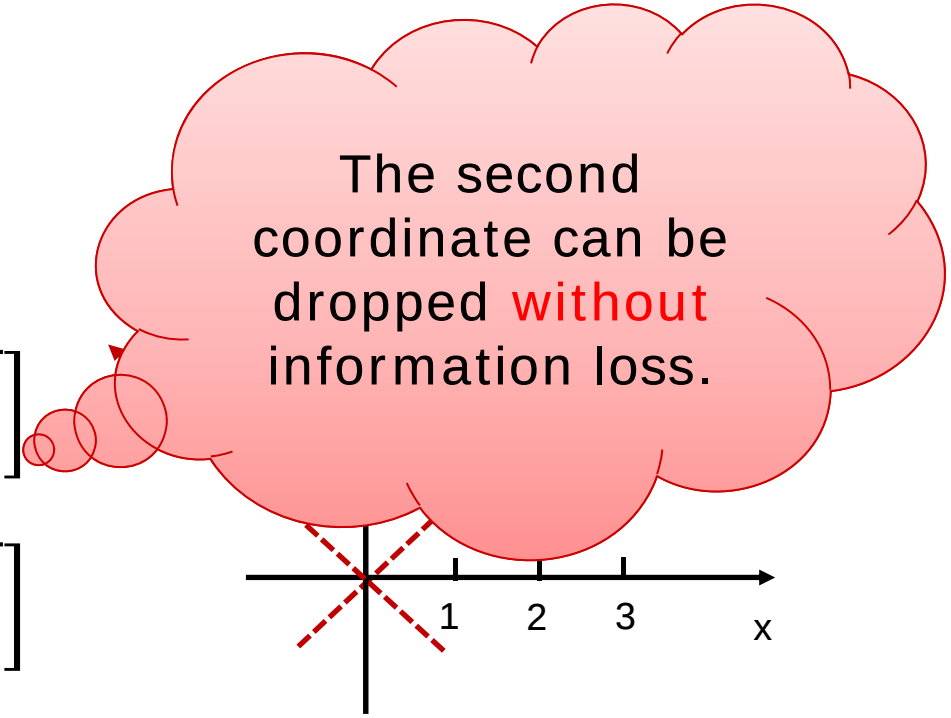
Motivations (Perfect Case)

- What is the new coordinates if we rotate the axis

$$\mathbf{x}_1 = \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2\sqrt{2} \\ 0 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 3\sqrt{2} \\ 0 \end{bmatrix}$$



Dimensionality Reduction with Axis Rotation (3)

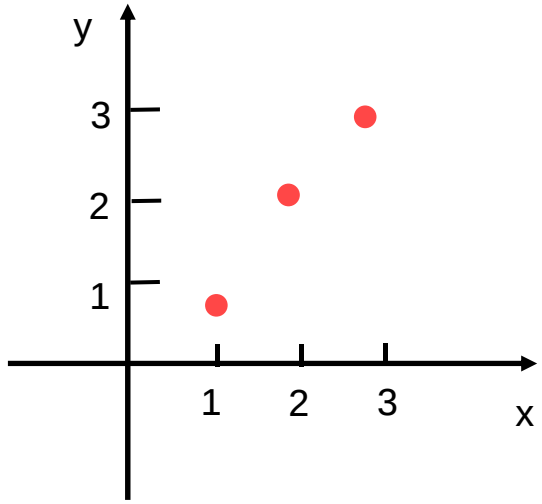
Motivations (Noisy Case)

- Consider the following 3 points in a 2-dimensional space

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0.9 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2.1 \\ 2 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 2.9 \\ 3.1 \end{bmatrix}$$



Dimensionality Reduction with Axis Rotation (4)

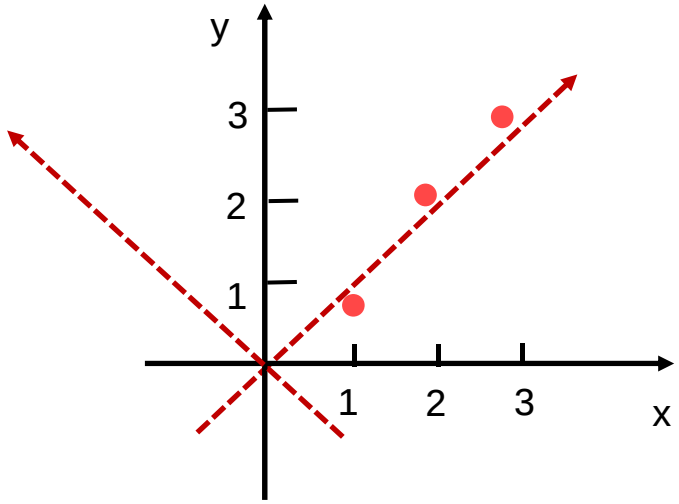
Motivations (Noisy Case)

- What is the new coordinates if we rotate the axis

$$\mathbf{x}_1 = \begin{bmatrix} 1.34 \\ 0.07 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2.89 \\ 0.07 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 4.24 \\ -0.14 \end{bmatrix}$$



Dimensionality Reduction with Axis Rotation (4)

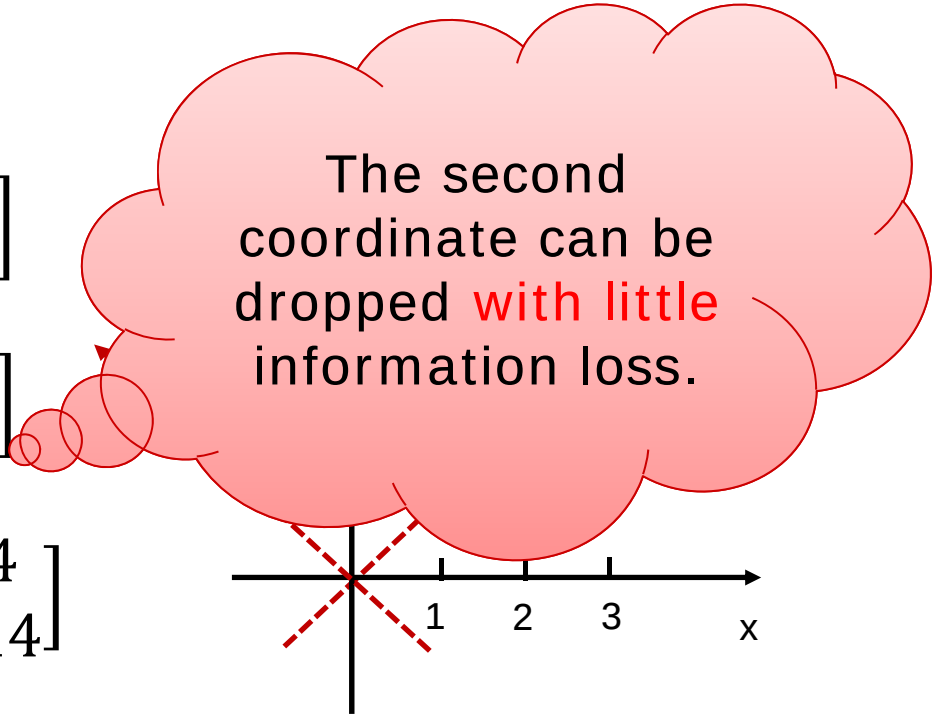
Motivations (Noisy Case)

- What is the new coordinates if we rotate the axis

$$\mathbf{x}_1 = \begin{bmatrix} 1.34 \\ 0.07 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2.89 \\ 0.07 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 4.24 \\ -0.14 \end{bmatrix}$$



Dimensionality Reduction with Axis Rotation (5)

- When does it Work?
  - Correlations exist among features
- Axis Rotation
  - Remove correlations
  - Reduce dimensionality
- How to Determine such Axis System?
  - Principal component analysis (PCA)
  - Singular value decomposition (SVD)

Axis Rotation—Mathematical Formulation (1)

- By default, the Original Coordinates are Defined with respect to the Standard Basis

$$\mathbf{x} = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^d \end{bmatrix} \in \mathbb{R}^d \Longleftrightarrow \mathbf{x} = x^1\mathbf{e}_1 + x^2\mathbf{e}_2 + \cdots + x^d\mathbf{e}_d$$

Axis Rotation—Mathematical Formulation (2)

- The New Coordinates with respect to a Orthonormal Basis $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d\}$
  - $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d]$ is a orthonormal matrix

$$\mathbf{x} = WW^{\top}\mathbf{x} = \left(\sum_{i=1}^{d} \mathbf{w}_i \mathbf{w}_i^{\top}\right)\mathbf{x} = \sum_{i=1}^{d} \mathbf{w}_i\left(\mathbf{w}_i^{\top}\mathbf{x}\right)$$

$$= (\mathbf{w}_1^{\top}\mathbf{x})\mathbf{w}_1 + (\mathbf{w}_2^{\top}\mathbf{x})\mathbf{w}_2 + \cdots + \left(\mathbf{w}_d^{\top}\mathbf{x}\right)\mathbf{w}_d$$

  - Thus, the new coordinates are

$$\mathbf{y} = \begin{bmatrix} \mathbf{w}_1^{\top}\mathbf{x} \\ \mathbf{w}_2^{\top}\mathbf{x} \\ \vdots \\ \mathbf{w}_d^{\top}\mathbf{x} \end{bmatrix} \in \mathbb{R}^d$$

Axis Rotation—Mathematical Formulation (2)

- The New Coordinates with respect to a Orthonormal Basis $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d\}$
  - $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d]$ is a orthonormal matrix

$$\mathbf{x} = WW^{\top}\mathbf{x} = \left(\sum_{i=1}^{d} \mathbf{w}_i \mathbf{w}_i^{\top}\right)\mathbf{x} = \sum_{i=1}^{d} \mathbf{w}_i \left(\mathbf{w}_i^{\top}\mathbf{x}\right)$$

$$= (\mathbf{w}_1^{\top}\mathbf{x})\mathbf{w}_1 + (\mathbf{w}_2^{\top}\mathbf{x})$$

  - Thus, the new c

$$\mathbf{y} = \begin{bmatrix} \mathbf{w}_1^{\top}\mathbf{x} \\ \mathbf{w}_2^{\top}\mathbf{x} \\ \vdots \\ \mathbf{w}_d^{\top}\mathbf{x} \end{bmatrix} \in \mathbb{R}^d$$

Dimensionality reduction is achieved by dropping some of the new coordinates.

Terminology

- $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d\}$
  - Basis
  - Directions

- $\mathbf{w}_i^\top \mathbf{x} = \langle \mathbf{w}_i, \mathbf{x} \rangle$
  - New coordinates
  - Projection of $\mathbf{x}$ along the direction $\mathbf{w}_i$

Principal Component Analysis (PCA)

- Given a set of Data Points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ where $\mathbf{x}_i \in \mathbb{R}^d$

- Finding a set of directions $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ such that the variance of

$$\left\{ \mathbf{y}_1 = \begin{bmatrix} \mathbf{w}_1^\top \mathbf{x}_1 \\ \mathbf{w}_2^\top \mathbf{x}_1 \\ \vdots \\ \mathbf{w}_k^\top \mathbf{x}_1 \end{bmatrix}, \mathbf{y}_2 = \begin{bmatrix} \mathbf{w}_1^\top \mathbf{x}_2 \\ \mathbf{w}_2^\top \mathbf{x}_2 \\ \vdots \\ \mathbf{w}_k^\top \mathbf{x}_2 \end{bmatrix}, \cdots, \mathbf{y}_n = \begin{bmatrix} \mathbf{w}_1^\top \mathbf{x}_n \\ \mathbf{w}_2^\top \mathbf{x}_n \\ \vdots \\ \mathbf{w}_k^\top \mathbf{x}_n \end{bmatrix} \right\}$$

are maximized

P nt Analysis (PCA)

For the purpose of dimensionality reduction, PCA only learns k directions.

- ta Points
  here $\mathbf{x}_i \in \mathbb{R}^d$

- Finding a set of directions $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ such that the variance of

$$\left\{ \mathbf{y}_1 = \begin{bmatrix} \mathbf{w}_1^\top \\ \\ \mathbf{w} \end{bmatrix} \right. \qquad \left. \begin{matrix} \mathbf{w}_1^\top \mathbf{x}_n \\ \mathbf{w}_2^\top \mathbf{x}_n \\ \vdots \\ \mathbf{w}_k^\top \mathbf{x}_n \end{matrix} \right\}$$

PCA uses variances to measure the quality of new coordinates.

are maximized

PCA—One-dimensional Case (1)

- New Coordinates of $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

$$\mathbf{w}_1^\top \mathbf{x}_1, \mathbf{w}_1^\top \mathbf{x}_2, \ldots, \mathbf{w}_1^\top \mathbf{x}_n$$

- Variance is

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}_1^\top \mathbf{x}_i - \mu)^2$$

where $\mu = \frac{1}{n}\sum_{i=1}^{n} \mathbf{w}_1^\top \mathbf{x}_i$ is the mean of new coordinates

PCA—One-dimensional Case (2)

- Let $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ be the mean vector
- Then, $\mu = \mathbf{w}_1^\top \bar{\mathbf{x}}$

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}_1^\top \mathbf{x}_i - \mu)^2 = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}_1^\top \mathbf{x}_i - \mathbf{w}_1^\top \bar{\mathbf{x}})^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{w}_1^\top (\mathbf{x}_i - \bar{\mathbf{x}})\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\mathbf{w}_1^\top (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{w}_1$$

$$= \mathbf{w}_1^\top \left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top\right)\mathbf{w}_1$$

PCA—One-dimensional Case (3)

The Optimization Problem of PCA

$$\max_{\mathbf{w}\in\mathbb{R}^d} \quad \mathbf{w}^\top C\mathbf{w}$$

$$s.t. \quad \|\mathbf{w}\|_2^2 = 1$$

where $C = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$ is the covariance matrix

The Solution (Rayleigh Quotient)

- Lagrangian: $-\mathbf{w}^\top C\mathbf{w} + \boldsymbol{\lambda}(\|\mathbf{w}\|_2^2 - \mathbf{1})$
- Set the gradient of $\mathbf{w}$ be zero

  $-2C\mathbf{w} + 2\lambda\mathbf{w} = 0 \Leftrightarrow C\mathbf{w} = \lambda\mathbf{w}$

PCA—One-dimensional Case (4)

- $(\mathbf{w}, \lambda)$ is eigenvector and eigenvalue of C
- The objective becomes

$$\mathbf{w}^\top C\mathbf{w} = \lambda \mathbf{w}^\top \mathbf{w} = \lambda$$

- Thus, we select the largest eigenvector and eigenvalue of C

The Algorithm

1. Calculate the mean vector $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$
2. Calculate the covariance matrix $C = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$
3. Calculate the largest eigenvector of C

Property of the Covariance Matrix

$$C = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$$

- C is symmetric
- C is positive semidefinite (PSD)
  - All the eigenvalues are non-negative
- The rank of C is at most $n-1$
  - Let $\bar{X} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \dots, \mathbf{x}_n - \bar{\mathbf{x}}] \in \mathbb{R}^{d\times n}$

    $$\text{rank}(C) = \text{rank}(\bar{X}\bar{X}^\top) = \text{rank}(\bar{X}) \le n-1$$

  - It has at most $n-1$ positive eigenvalues

PCA—k-dimensional Case (1)

- The Optimization Problem of PCA

$$\max_{W \in \mathbb{R}^{d \times k}} \quad \text{trace}(W^\top C W)$$

$$s.t. \qquad W^\top W = I$$

where $C = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$

- The Solution (Rayleigh Quotient)
  - $W = [\mathbf{w}_1, \ldots, \mathbf{w}_k]$, where $\mathbf{w}_1, \ldots, \mathbf{w}_k$ are the k largest eigenvectors of C
  - Section 5.2.2.(6) of [Lütkepohl 1996]
- Can also be defined in an incremental fashion

PCA—k-dimensional Case (2)

The Algorithm

1. Calculate the mean vector $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$
2. Calculate the covariance matrix $C = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^{\mathsf{T}}$
3. Calculate the k largest eigenvectors of C

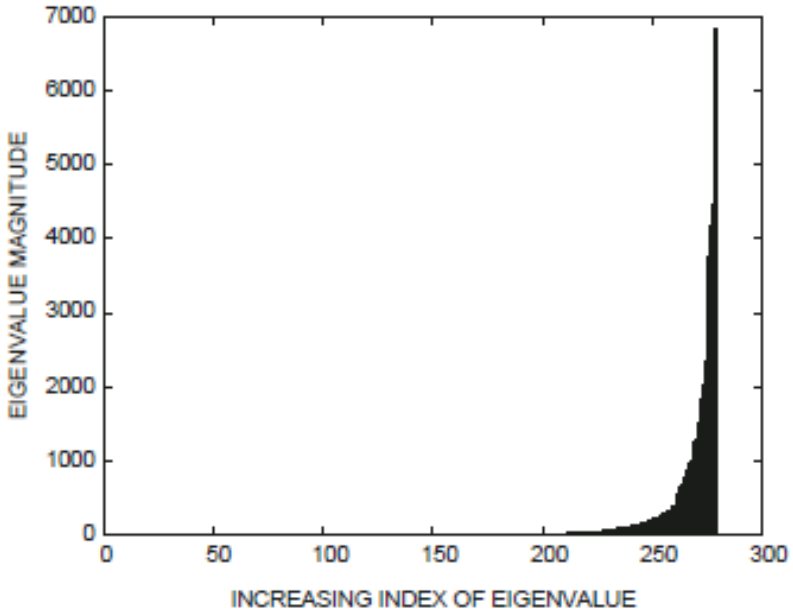
Eigenvalue

- λ_i is the variance of the i-th coordinate
- Measure the quality of PCA

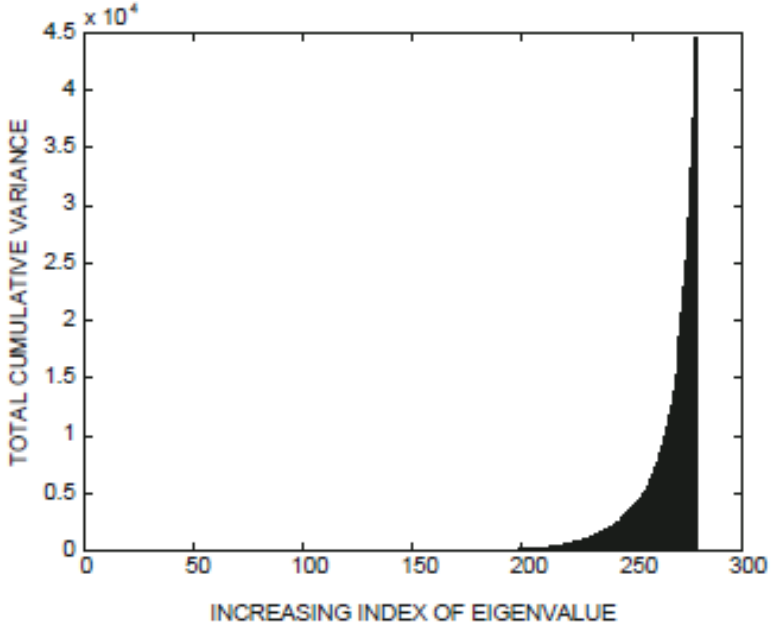
Captured $\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$ $\frac{\sum_{i=k+1}^{d}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$ Lost

An Example

- Arrythmia data set from the UCI



(a) Magnitude of Eigenvalues (Increasing Index): *Arrythmia*



(b) Variance in smallest k Eigenvalues: *Arrythmia*

Discussions of PCA

- The Key Operation
  - Eigendecomposition of C
- PCA can also be derived from the perspective of projection error minimization
  - Section 12.1.2 of [Bishop 2007]
- PCA is Linear Since

$$\mathbf{x} \in \mathbb{R}^d \rightarrow W^\top \mathbf{x} \in \mathbb{R}^k$$

  where $W = [\mathbf{w}_1, \ldots, \mathbf{w}_k] \in \mathbb{R}^{d \times k}$

- PCA is Unsupervised

Singular Value Decomposition (SVD)

- SVD of $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ with $d \leq n$

$$X = U\Sigma V^\top = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

  - $U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d] \in \mathbb{R}^{d \times d}$, $U^\top U = UU^\top = I$
  - $V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d] \in \mathbb{R}^{n \times d}$, $V^\top V = I$
  - $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_d) \in \mathbb{R}^{d \times d}$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d \geq 0$

Compact SVD

- SVD of $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ with $\text{rank(r)} < \min(d, n)$

$$X = U_r \Sigma_r V_r^\top = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

  - $U_r = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r] \in \mathbb{R}^{d \times r}$, $U_r^\top U_r = I$
  - $V_r = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r] \in \mathbb{R}^{n \times r}$, $V_r^\top V_r = I$
  - $\Sigma_r = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$

Dimensionality Reduction by SVD

- The Algorithm
  1. Calculate the k largest left singular vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ of $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$
- The New Coordinates of $\mathbf{x}$ are

$$U_k^\top \mathbf{x} = \begin{bmatrix} \mathbf{u}_1^\top \mathbf{x} \\ \mathbf{u}_2^\top \mathbf{x} \\ \vdots \\ \mathbf{u}_k^\top \mathbf{x} \end{bmatrix} \in \mathbb{R}^k$$

  - $U_k = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k] \in \mathbb{R}^{d \times k}$
- The New Coordinates of X is

$$U_k^\top \mathrm{X} = U_k^\top U_r \Sigma_r V_r^\top = \Sigma_k V_k^\top$$

SVD—A Energy-preserving Interpretation

The Optimization Problem of SVD

- 1-dimensional

$$\max_{\mathbf{w}\in\mathbb{R}^d} \quad \mathbf{w}^\top (XX^\top)\mathbf{w}$$
$$s.t. \quad \|\mathbf{w}\|_2^2 = 1$$

- k-dimensional

$$\max_{W\in\mathbb{R}^{d\times k}} \quad \mathrm{trace}(W^\top (XX^\top)W)$$
$$s.t. \quad W^\top W = I$$

Left (right) singular vectors of X are the eigenvectors of $XX^\top$ ($X^\top X$).

PCA by SVD

- Old Algorithm
  1. Calculate the mean vector $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$
  2. Calculate the covariance matrix $C = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^{\mathsf{T}}$
  3. Calculate the k-largest eigenvectors of C
- New Algorithm
  1. Calculate the mean vector $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$
  2. Calculate the k largest left singular vectors of $\bar{X} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots, \mathbf{x}_n - \bar{\mathbf{x}}]$

PCA by SVD

- Old Algorithm
  1. Calculate the mean vector $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$
  2. Calculate the covariance matrix $C = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i -$
  3. Calcu f C
- New Al
  1. Calculate $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$
  2. Calculate the k largest left singular vectors of $\bar{X} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots, \mathbf{x}_n - \bar{\mathbf{x}}]$

PCA is equivalent to SVD if $X = \bar{X}$, that is, if data are zero-mean.

Outline

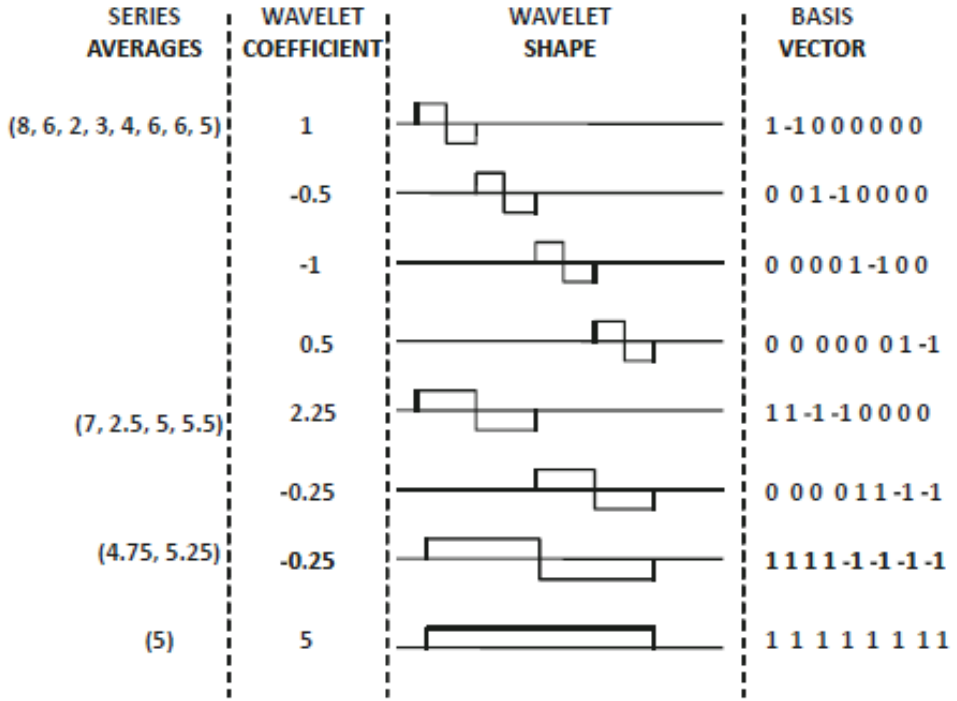
- Introduction
- Feature Extraction and Portability
- Data Cleaning
- Data Reduction and Transformation
  - Sampling
  - Feature Subset Selection
  - Dimensionality Reduction with Axis Rotation
  - **Dimensionality Reduction with Type Transformation**
- Summary

Dimensionality Reduction with Type Transformation

- Time Series to Multidimensional
  - Can also be viewed as a rotation of an axis system
  - Haar wavelet transform
- Weighted graphs to multidimensional
  - Multidimensional Scaling (MDS)
    - ✓ Edge represents distance
  - Spectral Transformations
    - ✓ Edge represents similarity

Haar Wavelet Transform (1)

- A New Basis for time series data
  - Each element basis is a time series (wavelets)
  - Coefficients can be calculated efficiently
  - Coefficients have nice interpretations



Haar Wavelet Transform (2)

- Given a Time Series $\mathbf{t}$ with length d

$$\mathbf{t} = \alpha^1 \mathbf{w}_1 + \alpha^2 \mathbf{w}_2 + \cdots + \alpha^d \mathbf{w}_d$$

where $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d$ are wavelets, and they are orthogonal to each other

- Normalization

$$\mathbf{t} = \alpha^1 \|\mathbf{w}_1\|_2 \frac{\mathbf{w}_1}{\|\mathbf{w}_1\|_2} + \alpha^2 \|\mathbf{w}_2\|_2 \frac{\mathbf{w}_2}{\|\mathbf{w}_2\|_2} + \cdots + \alpha^d \|\mathbf{w}_d\|_2 \frac{\mathbf{w}_d}{\|\mathbf{w}_d\|_2}$$

  - $\frac{\mathbf{w}_1}{\|\mathbf{w}_1\|_2}, \frac{\mathbf{w}_2}{\|\mathbf{w}_2\|_2}, \cdots, \frac{\mathbf{w}_d}{\|\mathbf{w}_d\|_2}$ are orthonormal to each other

Haar Wavelet Transform (3)

The New Coordinates

$$\mathbf{y} = \begin{bmatrix} \alpha^1 \|\mathbf{w}_1\|_2 \\ \alpha^2 \|\mathbf{w}_2\|_2 \\ \vdots \\ \alpha^d \|\mathbf{w}_d\|_2 \end{bmatrix} \in \mathbb{R}^d$$

Dimensionality Reduction

$$Y = \begin{bmatrix} \alpha_1^1 \|\mathbf{w}_1\|_2 & \alpha_2^1 \|\mathbf{w}_1\|_2 & \cdots & \alpha_n^1 \|\mathbf{w}_1\|_2 \\ \alpha_1^2 \|\mathbf{w}_2\|_2 & \alpha_2^2 \|\mathbf{w}_2\|_2 & \cdots & \alpha_n^2 \|\mathbf{w}_2\|_2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^d \|\mathbf{w}_d\|_2 & \alpha_2^d \|\mathbf{w}_d\|_2 & \cdots & \alpha_n^d \|\mathbf{w}_d\|_2 \end{bmatrix} \in \mathbb{R}^{d \times n}$$

- Feature Selection, PCA, SVD
- Sparse Representation

Multidimensional Scaling (MDS)

- Input
  - A graph with n nodes
  - $\delta_{ij} = \delta_{ji}$ be the distance between nodes i and j
- Output
  - A set of coordinates that fits the distance
- Metric MDS

$$\min_{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^k} \sum_{i,j:i<j} \left(\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \delta_{ij} \right)^2$$

Assume the specified distance matrix is Euclidean

The Algorithm

1. Calculate the dot-product matrix

$$S = -\frac{1}{2}\left(I - \frac{\mathbf{1}\mathbf{1}^\top}{n}\right)\Delta\left(I - \frac{\mathbf{1}\mathbf{1}^\top}{n}\right)$$

2. Eigen decompose S

$$S = U\Lambda U^\top = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$$

3. The new coordinates are

$$U_k \Lambda_k^{-1/2} \in \mathbb{R}^{n\times k}$$

$$U_k = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{n\times k},\ \Lambda_k = \mathrm{diag}(\lambda_1, \ldots, \lambda_k) \in \mathbb{R}^{k\times k}$$

Assume the specified distance matrix is Euclidean

The Algorithm

1. Calculate the dot-product matrix

$$S = -\frac{1}{2}\left(I - \frac{\mathbf{1}\mathbf{1}^{\mathsf{T}}}{n}\right)$$

2. Eig

3. Th re

$$U_k \Lambda_k^{-1/2} \in \mathbb{R}^{n \times k}$$

$$U_k = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{n \times k},\ \Lambda_k = \mathrm{diag}(\lambda_1, \ldots, \lambda_k) \in \mathbb{R}^{k \times k}$$

Metric MDS is equivalent to PCA, if the distance matrix is Euclidean.

Spectral Transformations (1)

- Input
  - A graph with n nodes
  - $w_{ij} = w_{ji}$ be the similarity between nodes i and j
- Output
  - A set of coordinates that preserves the similarity
- The Objective

$$\min_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n \in \mathbb{R}^k} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2$$

Spectral Transformations (2)

The Optimization Problem

$$\min_{Y \in \mathbb{R}^{n \times k}} \quad \text{trace}(Y^\top L Y)$$
$$s.t. \quad Y^\top D Y = I$$

- $Y = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times k}$, $L = D - W$
- D is a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} w_{ij}$

Generalized Eigenproblem

$$L\mathbf{y} = \lambda D\mathbf{y}$$

The Solution [Luxburg 2007]

- $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_k] \in \mathbb{R}^{n \times k}$, where $\mathbf{y}_i$ is the i-th smallest eigenvector

Outline

- Introduction
- Feature Extraction and Portability
- Data Cleaning
- Data Reduction and Transformation
  - Sampling
  - Feature Subset Selection
  - Dimensionality Reduction with Axis Rotation
  - Dimensionality Reduction with Type Transformation
- **Summary**

Summary

- Feature Extraction and Portability
- Data Cleaning
- Data Reduction by Sampling
- Dimensionality Reduction with Axis Rotation
  - PCA, SVD
- Dimensionality Reduction with Type Transformation
  - Haar Wavelet Transform, MDS, Spectral Transformation

Reference

- Bishop 2007
  - Christopher Bishop. Pattern Recognition and Machine Learning. Springer, 2007.
- Lütkepohl 1996
  - H. Lütkepohl. Handbook of Matrices. Wiley, 1996.
- Luxburg 2007
  - Ulrike von Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17(4): 395-416, 2007.