Data Mining:

Concepts and Techniques

(3rd ed.)

— Chapter 3 —

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

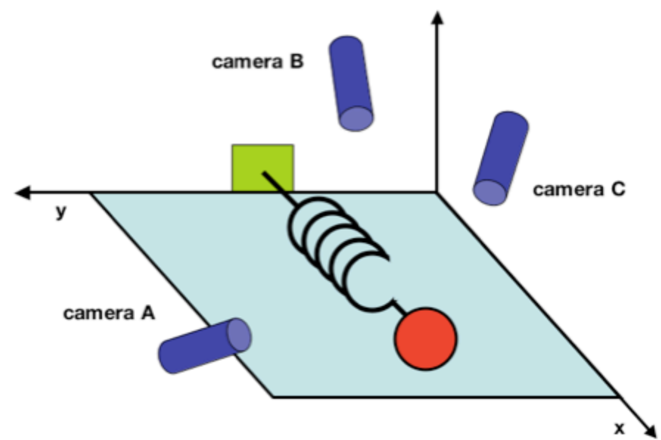
Focused Topic:
Principal Component Analysis

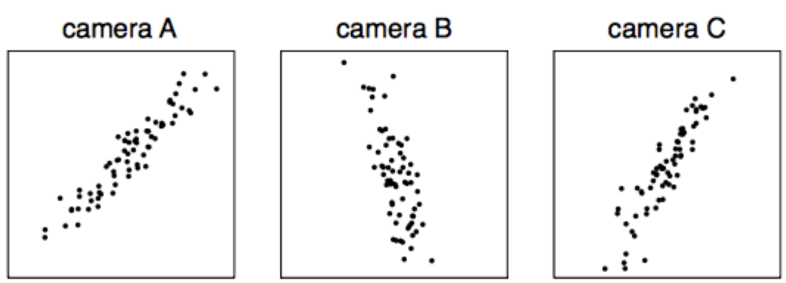
A dimensionality reduction method

Motivation: Data Compression for Feature Selection and Visualiation

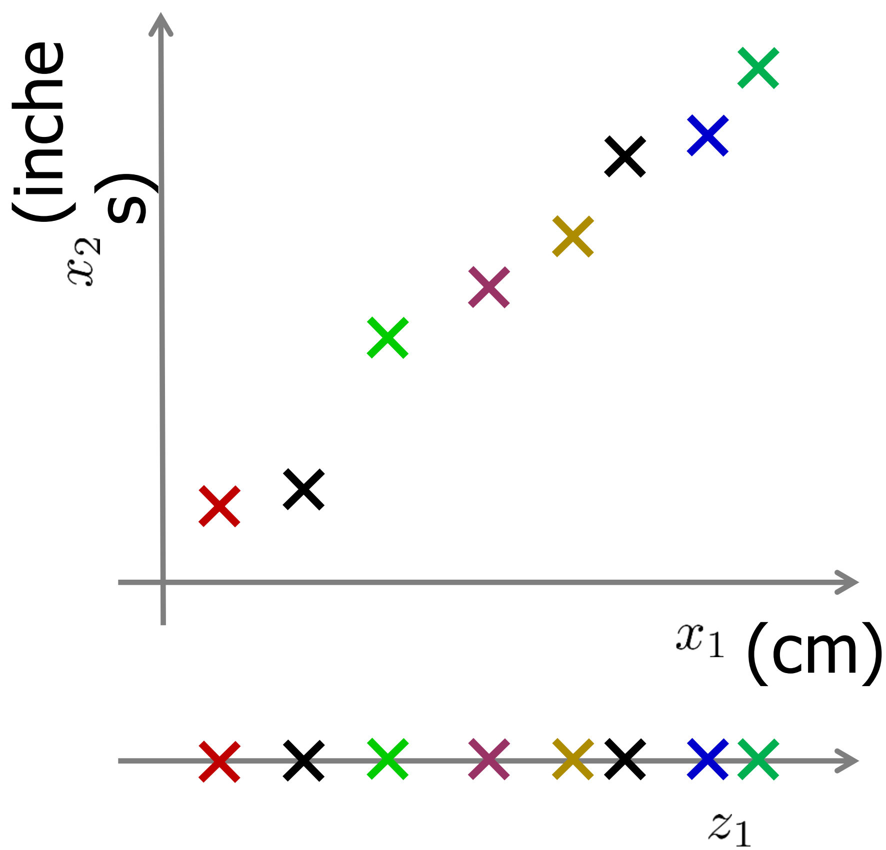
Data Compression Example 1

- Ball of mass m attached to massless, frictionless spring
- Ball moved away from equilibrium results in spring
- oscillating indefinitely along x-axis
- Three cameras ~ three dimensions
- However, all dynamics can be a function of only a single variable x





Data Compression Example 2



Reduce data from 2D to 1D

$$x^{(1)} \rightarrow z^{(1)}$$

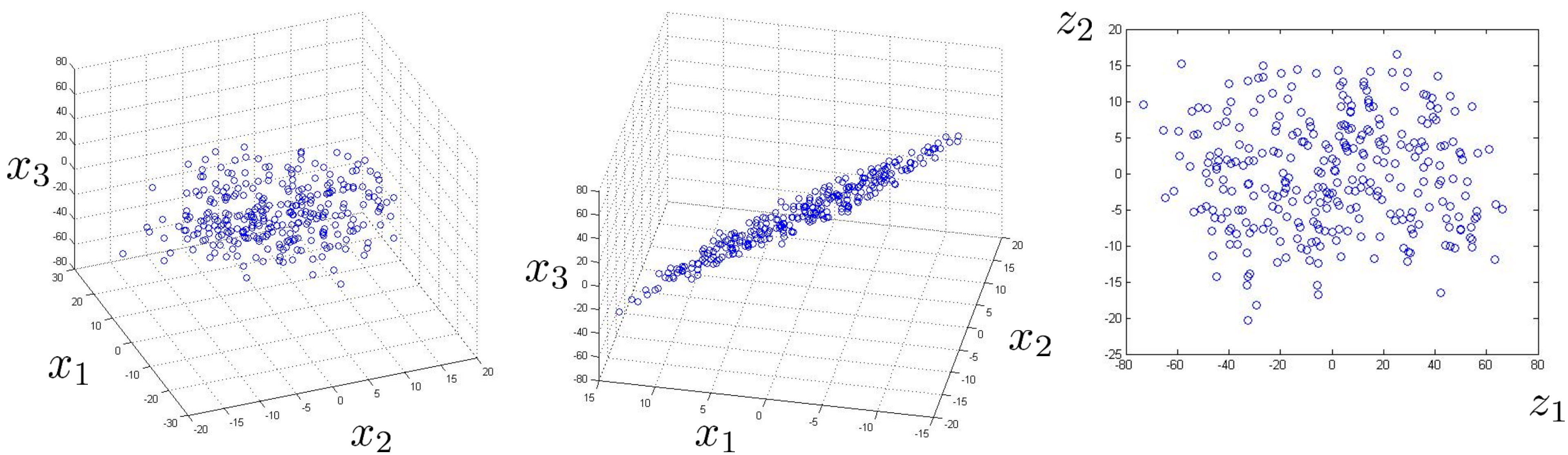
$$x^{(2)} \rightarrow z^{(2)}$$

$$\vdots$$

$$x^{(m)} \rightarrow z^{(m)}$$

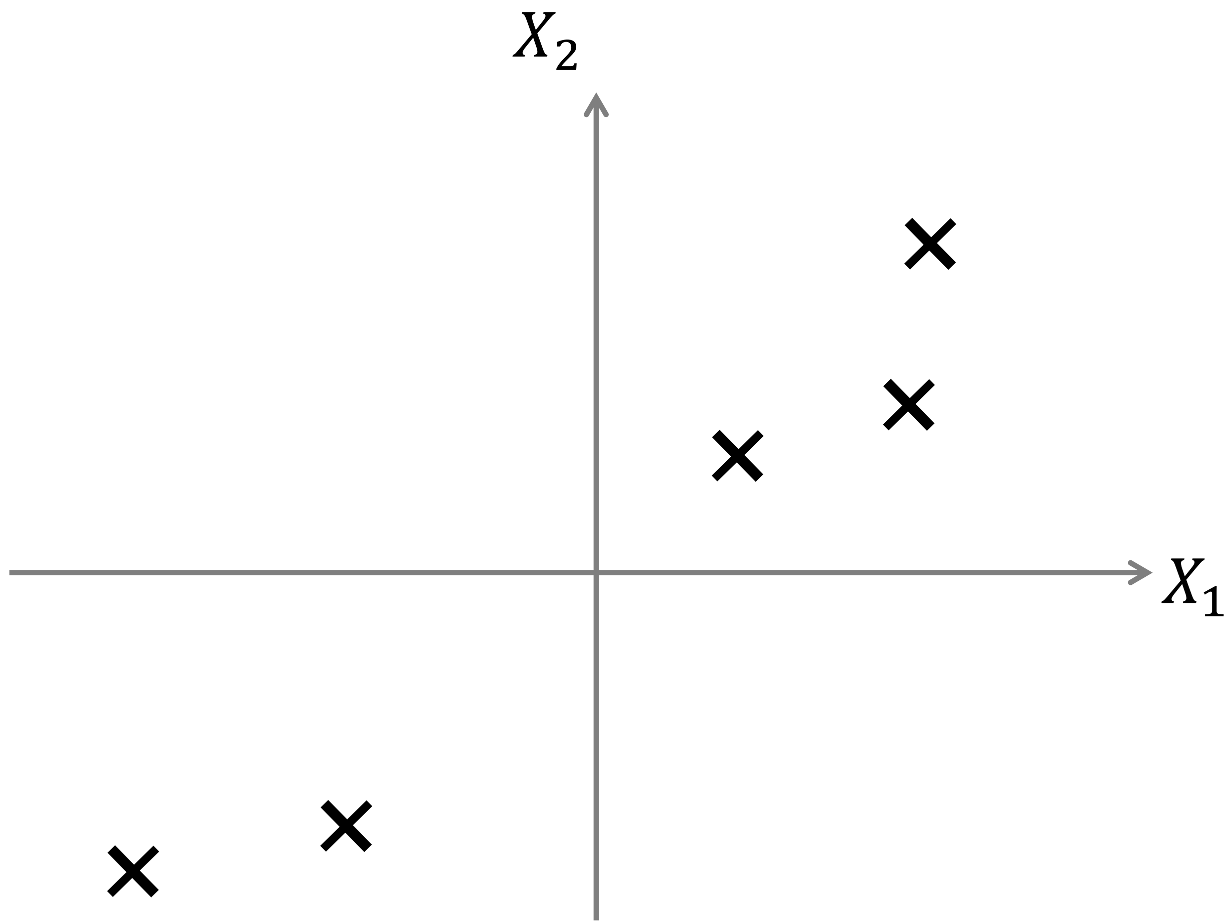
Data Compression Example 3

Reduce data from 3D to 2D

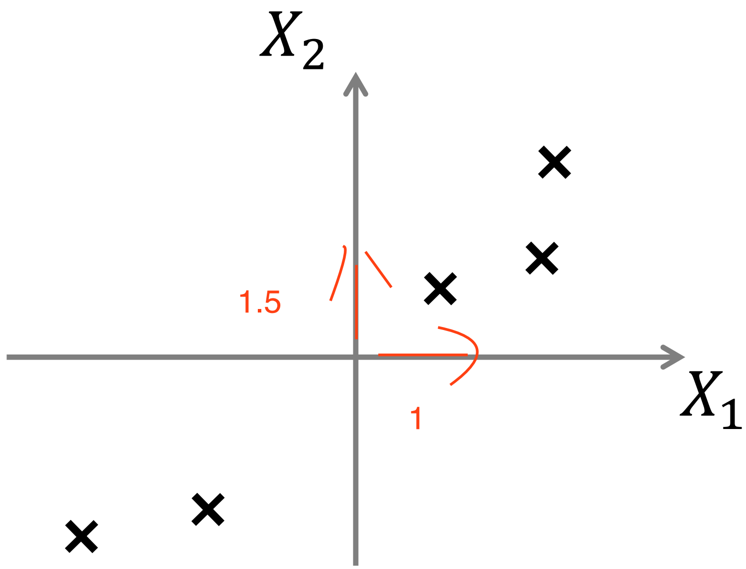


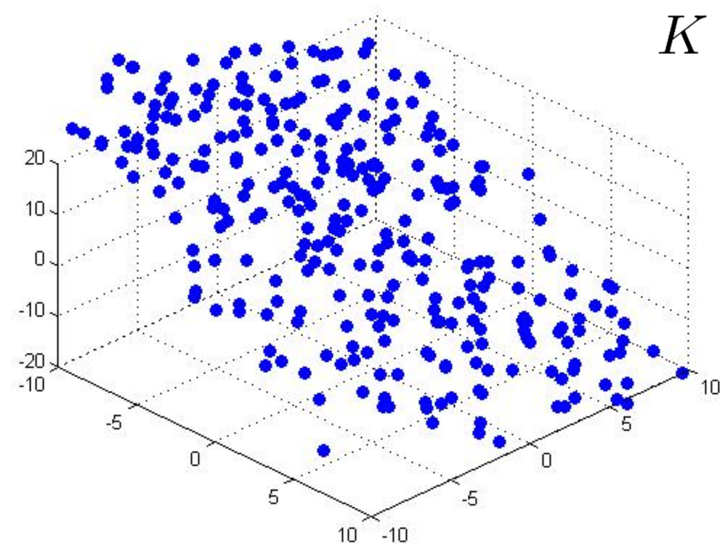
PCA PROBLEM FORMULATION

Principal Component Analysis (PCA)
Problem formulation



Principal Component Analysis (PCA)
Problem formulation





Orig Base

Reduce from 2-dimension to 1-dimension: Find a direction (a vector $P_1 \in R^2$) onto which to project the data so as to minimize the projection error.

Reduce from n-dimension to k-dimension: Find k vectors $P_1, P_2, \ldots, P_k$ onto which to project the data, so as to minimize the projection error.

In summary, the goal of PCA is…

- Compute the most meaningful basis to re-express a noisy data set
- Hope that this new basis will filter out the noise and reveal hidden structure
- In the toy example:
  - Determine that the dynamics are along the x-axis

PCA ALGORITHM

Naïve Basis: formed directly from the method used to gather data

- At each point in time, record 2 coordinates of ball position in each of the 3 images
- After 10 minutes at 120Hz, we have 10×60×120=7200 6-dimensional vectors
- These vectors can be represented in arbitrary coordinate systems

$$\vec{X} = \begin{bmatrix} x_A \\ y_A \\ x_B \\ y_B \\ x_C \\ y_C \end{bmatrix}$$

6*7200

$X =$

PCA: Changing basis to express the data better

- PCA: Is there another basis, which is a linear combination of the original basis, that best re-expresses our data set?
- Assumption: linearity
  - Restricts set of potential bases
  - Simplifies the characterization of a complex system

Change of basis expressed in linear algebra

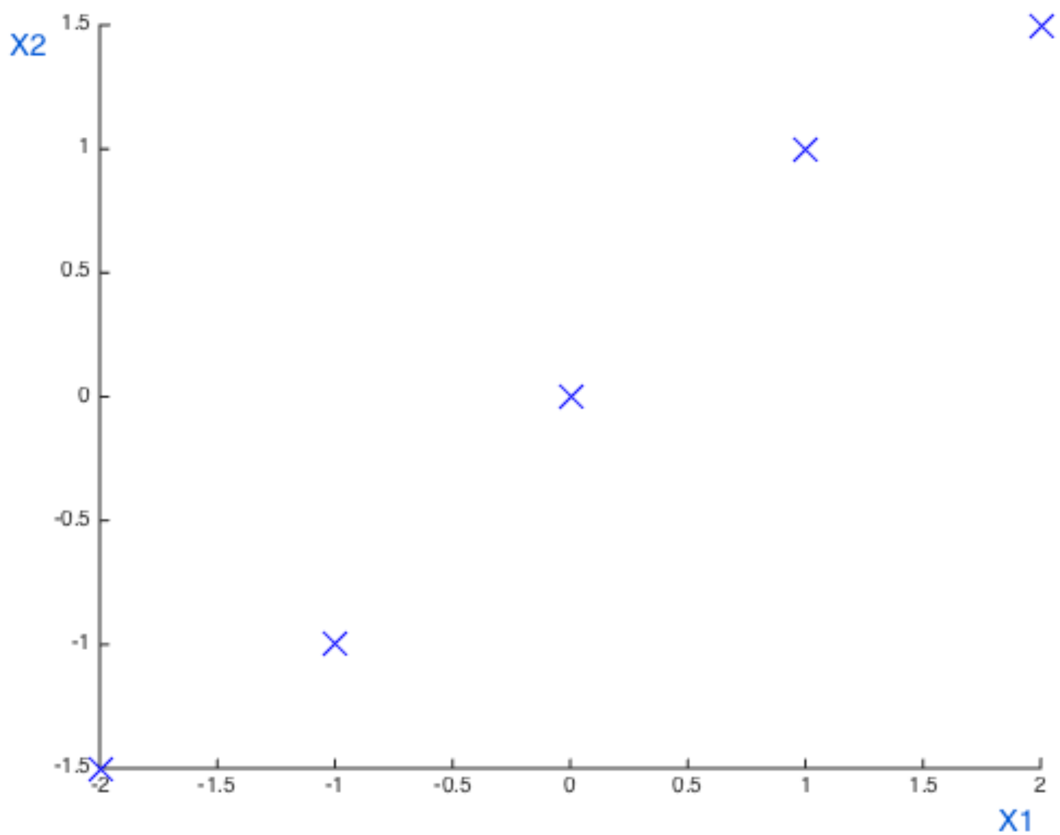
- X is original data (m×n, m=6, n=7200)
- Let Y be another m×n matrix such that Y=PX (or, k x n if first k dimensions).
- P is a matrix that transforms X into Y
  - What is the size of P?
  - Geometrically it is a rotation and stretch
  - The rows of P {p1,..., pm} are the new basis vectors for the columns of X
    - Called principal components of X
  - Each element of yi is a dot product of xi with the corresponding row of P (a projection of xi onto pj)

$$\mathbf{PX} = \begin{bmatrix} \mathbf{p_1} \\ \vdots \\ \mathbf{p_m} \end{bmatrix} \begin{bmatrix} \mathbf{x_1} & \cdots & \mathbf{x_n} \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{p_1 \cdot x_1} & \cdots & \mathbf{p_1 \cdot x_n} \\ \vdots & \ddots & \vdots \\ \mathbf{p_m \cdot x_1} & \cdots & \mathbf{p_m \cdot x_n} \end{bmatrix} \qquad \mathbf{y}_i = \begin{bmatrix} \mathbf{p_1 \cdot x_i} \\ \vdots \\ \mathbf{p_m \cdot x_i} \end{bmatrix}$$

Running example for two measurements

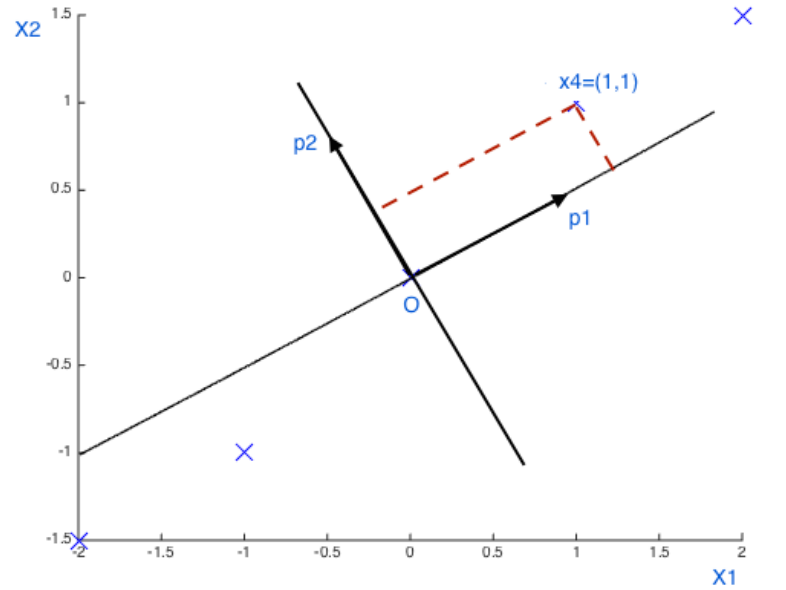
	x1	x2	x3	x4	x5
X1	-2	-1	0	1	2
X2	-1.5	-1	0	1	1.5



Running example: Changing basis

- Assume that by someway we know the new basis is a 2-d space, i.e., specified by
  - Vector p1=(0.9, 0.45)
  - Vector p2=(-0.45, 0.9)
- Matrix P: (P is orthonormal)

$$P = \begin{bmatrix} 0.9 & 0.45 \\ -0.45 & 0.9 \end{bmatrix}$$



- Change basis for x4 (1,1): two dot products: A·v1 and A·v2 equivalent to two projections

$$x4' = P * x4 = \begin{bmatrix} 0.9 & 0.45 \\ -0.45 & 0.9 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.35 \\ 0.45 \end{bmatrix}$$

	x1	x2	x3	x4	x5
X1	-2	-1	0	1	2
X2	-1.5	-1	0	1	1.5

- Change basis for entire dataset X:

$$Y = PX = \begin{bmatrix} 0.9 & 0.45 \\ -0.45 & 0.9 \end{bmatrix} * \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -1.5 & -1 & 0 & 1 & 1.5 \end{bmatrix}$$

$$Y = \begin{bmatrix} -2.48 & -1.35 & 0 & 1.35 & 2.48 \\ -0.45 & -0.45 & 0 & 0.45 & 0.45 \end{bmatrix}$$

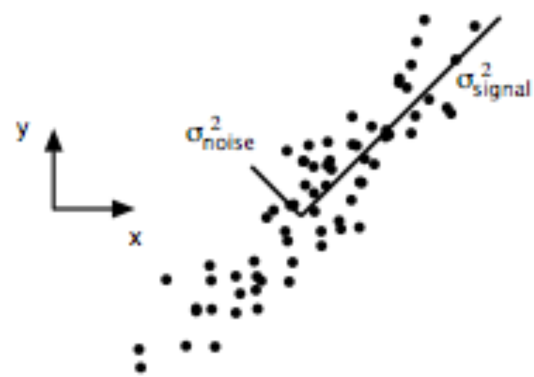
Finding an appropriate change of basis by minimizing noise and redundancy

- Finding change of basis = Finding matrix P
- What is the best way to re-express X? What features would we like Y to exhibit?
- If we call X "garbled data", garbling in a linear system can refer to two things:
  - Noise
  - Redundancy

Signal and noise can be measured by variance

- Measurement noise in any reasonable data set should be low.
- In the toy example: ball travels in straight line
  - Any deviation must be noise.
  - The variance due to the signal and noise are indicated by each line in the diagram.
- Assumption: directions with largest variances in our measurement space contain the dynamics of interest.
  - Signal-to-Noise Ratio (SNR)

$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}.$$



1. → Goal 1: **Maximize variances** of new dimensions.

Redundancy happens when we have little info about the dimensions of interest when collecting data

- Is it necessary to record 2 variables for the ball-spring system?
- Is it necessary to use 3 cameras?
- Covariance Cov(X1, X2) (or correlation) reveals redundancy between X1 and X2.
- Redundancy should be removed.
  - the new basis can use smaller number of dimensions than the naïve basis does.

1. → Goal 2: **Minimize co-variance** between new dimensions.

Covariance Matrix tells us about signals and redundancies

$$\Sigma = \begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}$$

- Diagonal elements of C_X : variances of dimensions
  - Large → interesting dynamics
  - Small → noise
- Off-diagonal elements of C_X : covariances of two different dimensions
  - Large → high redundancy
  - Small → low redundancy
- **Goal 1+2**: Covariance matrix of the new space C_Y should be diagonal!

$$\begin{bmatrix} 128 & 0 \\ 0 & \end{bmatrix}$$

If all dimensions have zero-mean, we have a simpler formula for covariance matrix

$$
\mathbf{C_X} = \begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}
$$

- If all dimensions have zero-mean, covariance matrix $\mathbf{C_X} = \frac{1}{n}\mathbf{X}\mathbf{X}^{\mathbf{T}}$
- In the toy example, because both measures are **zero-mean normalized**:

Signal

$$
C_x = \frac{1}{5} * \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -1.5 & -1 & 0 & 1 & 1.5 \end{bmatrix} * \begin{bmatrix} -2 & -1.5 \\ -1 & -1 \\ 0 & 0 \\ 1 & 1 \\ 2 & 1.5 \end{bmatrix} = \begin{bmatrix} 2.0 & 1.6 \\ 1.6 & 1.3 \end{bmatrix}
$$

Signal

- **First**: Zero-mean normalize each dimension, to simply computation.

Solving PCA (1): Using eigenvector decomposition

- The objective

  Find some orthonormal matrix $\mathbf{P}$ in $\mathbf{Y} = \mathbf{PX}$ such that $\mathbf{C_Y} \equiv \frac{1}{n}\mathbf{YY}^T$ is a diagonal matrix. The rows of $\mathbf{P}$ are the *principal components* of $\mathbf{X}$.

- New assumption: P must be orthonormal
  - There are many possible P in the search space
  - But if P is orthonormal, there is an efficient solution to find P:
    - Eigenvector decomposition

Running example: C_Y is not always diagonal

- In the running example, we chose an orthonormal P:
  - p1 and p2 are unit vectors: |p1|=|p2|=1
  - p1 and p2 are orthogonal:

  p1·p2 =(0.9, 0.45) · (-0.45, 0.9) = 0
- However, C_Y is not diagonal

$$C_y = \frac{1}{5} * Y * Y^T = \begin{bmatrix} 3.18 & 0.69 \\ 0.69 & 0.16 \end{bmatrix}$$

- How to find P to make C_Y diagonal?

Solving PCA (2): Rewrite C_Y in terms of P and C_X

$$
\begin{aligned}
\mathbf{C_Y} &= \frac{1}{n}\mathbf{Y}\mathbf{Y}^T \\
&= \frac{1}{n}(\mathbf{PX})(\mathbf{PX})^T \\
&= \frac{1}{n}\mathbf{P}\mathbf{X}\mathbf{X}^T\mathbf{P}^T \\
&= \mathbf{P}(\frac{1}{n}\mathbf{X}\mathbf{X}^T)\mathbf{P}^T \\
\mathbf{C_Y} &= \mathbf{P}\mathbf{C_X}\mathbf{P}^T
\end{aligned}
$$

Solving PCA (3): As we want C_Y diagonal, put a diagonal matrix to the formula of C_Y by eigenvector decomposition

Covariance matrix C_x is always symmetric
E: Eigenvectors of C_x
D: Diagonal matrix
(Theorem 4) When C_x is symmetric, $C_x = EDE^T$

$$\mathbf{C_Y} = \mathbf{P\ C_X P^T}$$
$$= \mathbf{P(E\ D\ E^T)P^T}$$

Solving PCA (4): Canceling out anything other than the diagonal matrix in C_Y by selecting P as a matrix where each row is an eigenvector of C_X

Select $E = P^T$

P = E^t

$$
\begin{aligned}
\mathbf{C_Y} &= \mathbf{P\,C_X P^T} \\
&= \mathbf{P\left(E\,D\,E^T\right)P^T} \\
&= \mathbf{P\left(P^T D\,P\right)P^T} \\
&= \mathbf{\left(P\,P^T\right)D\left(PP^T\right)}
\end{aligned}
$$

E = P^t = P^ -1

(Theorem 1) The inverse of an orthogonal matrix is its transpose. As P is an orthogonal matrix, C_Y turns out to be diagonal

$$
\begin{aligned}
\mathbf{C_Y} &= \mathbf{\left(PP^{-1}\right)D\left(PP^{-1}\right)} \\
&= \mathbf{D}
\end{aligned}
$$

PCA by eigenvalue decomposition in the running example

X is already zero-mean

Eigenvectors of C_x (using Matlab): [0.627 -0.779] and [-0.779 -0.627]

Each row of P is an eigenvector

$$P = \begin{bmatrix} 0.627 & -0.779 \\ -0.779 & -0.627 \end{bmatrix}$$

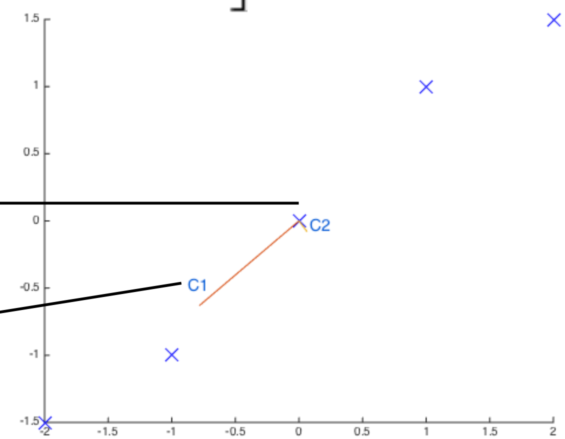
Cx * e = lamda * e

Project X to the new space to obtain Y

X4

$$Y = PX = \begin{bmatrix} 0.627 & -0.779 \\ -0.779 & -0.627 \end{bmatrix} * \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -1.5 & -1 & 0 & 1 & 1.5 \end{bmatrix}$$

$$Y = \begin{bmatrix} -0.0855 & 0.1520 & 0 & -0.1520 & 0.0855 \\ 2.4985 & 1.4060 & 0 & -1.4060 & -2.4985 \end{bmatrix}$$



Y4

Guess what, C_Y is diagonal!

$$C_Y = \frac{1}{5} * Y * Y^T = \begin{bmatrix} 0.012 & 0 \\ 0 & 3.288 \end{bmatrix}$$

If we want to use only the most important principal component

$$Y = \begin{bmatrix} -0.779 & -0.627 \end{bmatrix} * \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -1.5 & -1 & 0 & 1 & 1.5 \end{bmatrix} = \begin{bmatrix} 2.4985 & 1.4060 & 0 & -1.4060 & -2.4985 \end{bmatrix}$$

Summary: PCA by Eigenvector Decomposition

- Make $\mathbf{X}$ a mean-normalized data matrix with m dims x n points.
- Find covariance matrix $\mathbf{C_X} = \frac{1}{n}\mathbf{XX^T}$, an m x m symmetric matrix.
- For $\mathbf{C_X}$: $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_m]$ (eigenvectors) and $\mathbf{D}$ ($\mathbf{C_Y} = \mathbf{D}$, new covar), sorted by variance each -- Use eigenvector decomposition or SVD.
- Find $\mathbf{P} = \begin{bmatrix} \mathbf{e_1^T} \\ \vdots \\ \vdots \\ \mathbf{e}_{\boldsymbol{m}}^{\mathbf{T}} \end{bmatrix}$ as principle components.
- Select $\mathbf{P}_{\boldsymbol{k}}$ as first k principle components.
- New data $\mathbf{Y} = \mathbf{P}_k\mathbf{X}.$

- What k to choose? Large enough to preserve sufficient covariance.

Singular Value Decomposition (SVD)

- An alternative to finding eigenvectors for the covariance matrix
- More numerically stable than directly finding eigenvectors
- Widely used in practice for PCA

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values s.t. each old value can be identified with one of the new values
- Methods
  - Smoothing: Remove noise from data
  - Attribute/feature construction
    - New attributes constructed from the given ones
  - Aggregation: Summarization, data cube construction
  - Normalization: Scaled to fall within a smaller, specified range
    - min-max normalization
    - z-score normalization
    - normalization by decimal scaling
  - Discretization: Concept hierarchy climbing

Normalization

- **Min-max normalization**: to [new_min_A, new_max_A]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,000 is mapped to $\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ: mean, σ: standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

  - Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

Discretization

- Three types of attributes
  - Nominal—values from an unordered set, e.g., color, profession
  - Ordinal—values from an ordered set, e.g., military or academic rank
  - Numeric—real numbers, e.g., integer or real numbers
- Discretization: Divide the range of a continuous attribute into intervals
  - Interval labels can then be used to replace actual data values
  - Reduce data size by discretization
  - Supervised vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute
  - Prepare for further analysis, e.g., classification

Data Discretization Methods

- Typical methods: All the methods can be applied recursively
  - Binning
    - Top-down split, unsupervised
  - Histogram analysis
    - Top-down split, unsupervised
  - Clustering analysis (unsupervised, top-down split or bottom-up merge)
  - Decision-tree analysis (supervised, top-down split)
  - Correlation (e.g., χ^2) analysis (unsupervised, bottom-up merge)

Simple Discretization: Binning

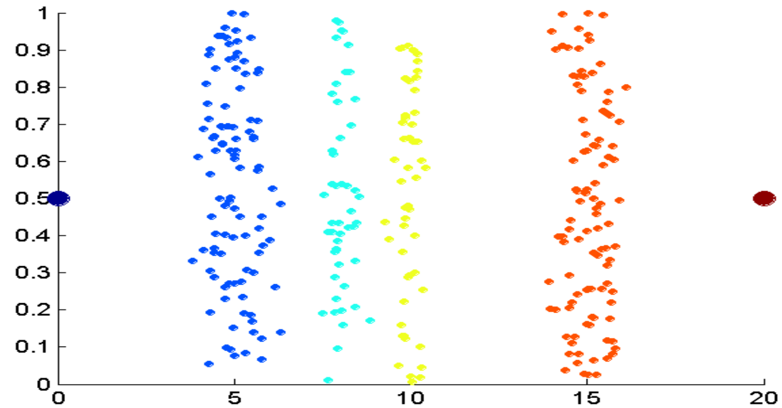
- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples

Binning Methods for Data Smoothing

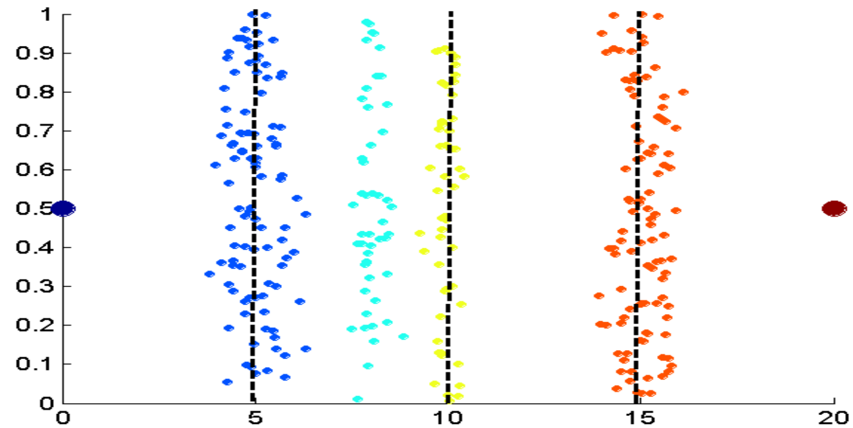
- Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into equal-frequency (**equi-depth**) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Smoothing by **bin means**:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Smoothing by **bin boundaries**:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

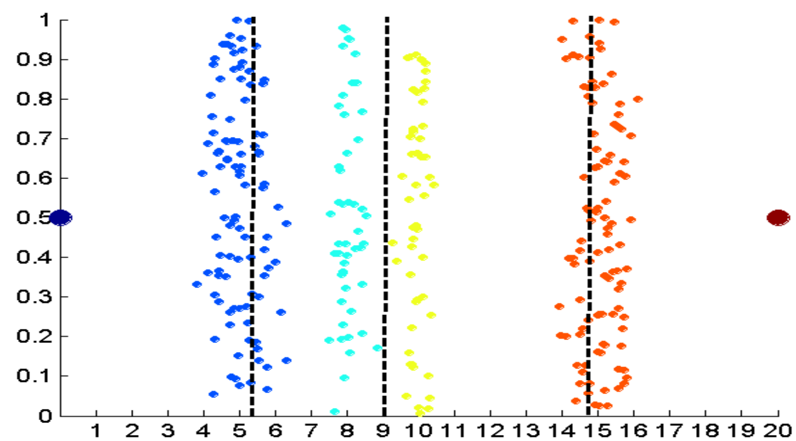
Discretization Without Using Class Labels (Binning vs. Clustering)



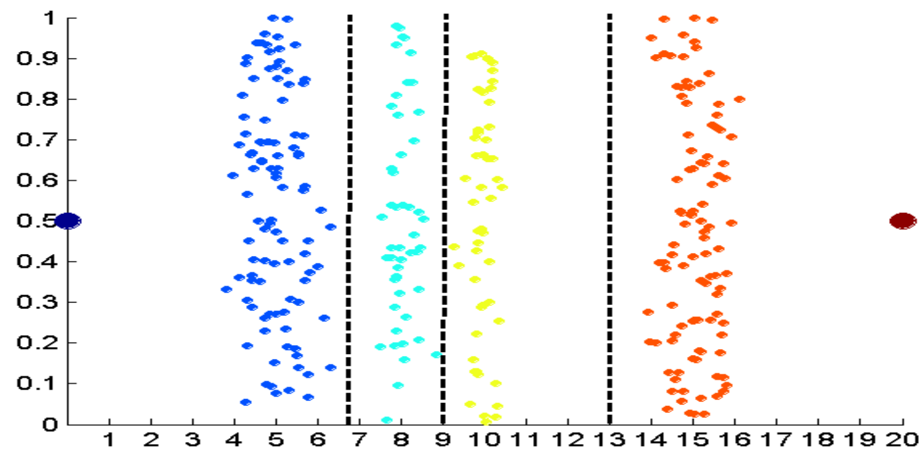
Data



Equal-width binning



Equal-frequency binning



K-means clustering leads to better results

Discretization by Classification & Correlation Analysis

- Classification (e.g., decision tree analysis)
  - Supervised: Given class labels, e.g., cancerous vs. benign
  - Using *entropy* to determine split point (discretization point)
  - Top-down, recursive split
  - Details to be covered in Chapter "Classification"
- Correlation analysis (e.g., Chi-merge: χ^2-based discretization)
  - Supervised: use class information
  - Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
  - Merge performed recursively, until a predefined stopping condition

Concept Hierarchy Generation for Nominal Data

- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
  - *street* < *city* < *state* < *country*
- Specification of a hierarchy for a set of values by explicit data grouping
  - {Urbana, Champaign, Chicago} < Illinois
- Specification of only a partial set of attributes
  - E.g., only *street* < *city*, not others
- Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
  - Country 15 distinct values
  - State 365 distinct values
  - City 3567 distinct values
  - Street 674339 distinct values

Chapter 3: Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

Summary

- **Data quality**: accuracy, completeness, consistency, timeliness, believability, interpretability
- **Data cleaning**: e.g. missing/noisy values, outliers
- **Data integration** from multiple sources:
  - Entity identification problem; Remove redundancies; Detect inconsistencies
- **Data reduction**
  - Dimensionality reduction; Numerosity reduction; Data compression
- **Data transformation and data discretization**
  - Normalization; Concept hierarchy generation