

---

# Dimensionality Reduction for Data Mining

## - Techniques, Applications and Trends

---

**Lei Yu**  
**Binghamton University**



**Jieping Ye, Huan Liu**  
**Arizona State University**



# Outline

- Introduction to dimensionality reduction
- Feature selection (part I)
  - Basics
  - Representative algorithms
  - Recent advances
  - Applications
- Feature extraction (part II)
- Recent trends in dimensionality reduction

---

# Why Dimensionality Reduction?

- It is so easy and convenient to collect data
  - An experiment
- Data is not collected only for data mining
- Data accumulates in an unprecedented speed
- Data preprocessing is an important part for *effective* machine learning and data mining
- Dimensionality reduction is an effective approach to downsizing data

# Why Dimensionality Reduction?

- Most machine learning and data mining techniques may not be effective for high-dimensional data
  - ❑ **Curse of Dimensionality**
  - ❑ Query accuracy and efficiency degrade rapidly as the dimension increases.
- The **intrinsic** dimension may be small.
  - ❑ For example, the number of genes responsible for a certain type of disease may be small.

# Why Dimensionality Reduction?

- **Visualization**: projection of high-dimensional data onto 2D or 3D.
- **Data compression**: efficient storage and retrieval.
- **Noise removal**: positive effect on query accuracy.

# Application of Dimensionality Reduction

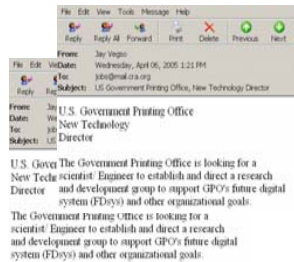
- Customer relationship management
- Text mining
- Image retrieval
- Microarray data analysis
- Protein classification
- Face recognition
- Handwritten digit recognition
- Intrusion detection

# Document Classification

## Web Pages



## Emails



## Internet



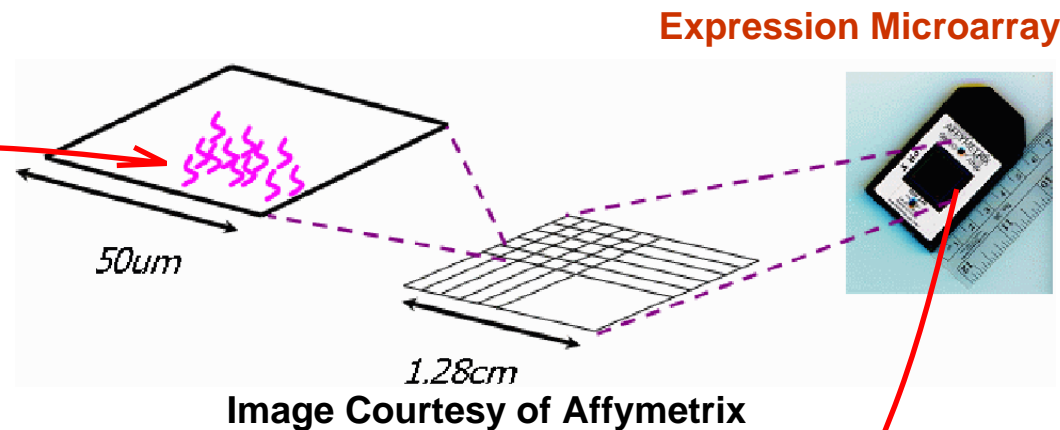
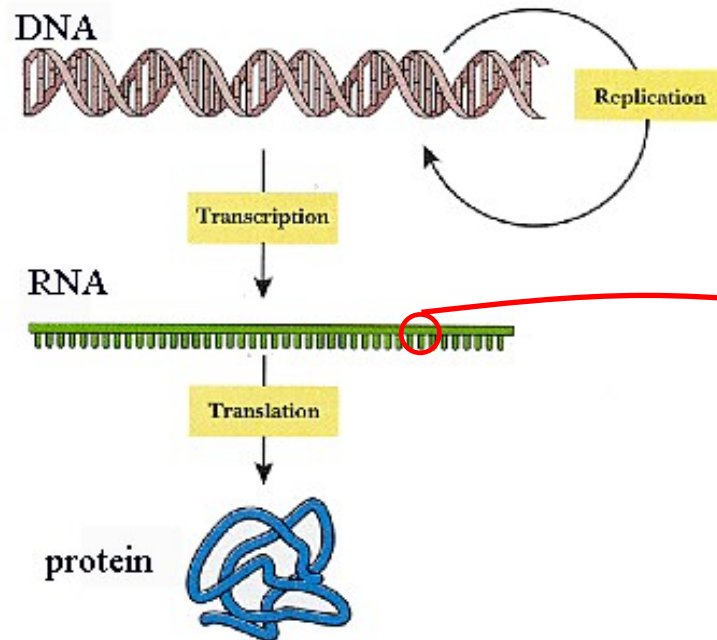
## Digital Libraries

## Terms

	$T_1$	$T_2$	.....	$T_N$	$C$
$D_1$	12	0	.....	6	Sports
$D_2$	3	10	.....	28	Travel
$\vdots$	$\vdots$			$\vdots$	$\vdots$
$D_M$	0	11	.....	16	Jobs

- **Task:** To classify unlabeled documents into categories
- **Challenge:** thousands of terms
- **Solution:** to apply dimensionality reduction

# Gene Expression Microarray Analysis



- **Task:** To classify novel samples into known disease types (disease diagnosis)
- **Challenge:** thousands of genes, few samples
- **Solution:** to apply dimensionality reduction

Gene \ Sample	M23197_at	U66497_at	M92287_at	...	Class
Sample 1	261	88	4778	...	ALL
Sample 2	101	74	2700	...	ALL
Sample 3	1450	34	498	...	AML
.	.	.	.	...	.
.	.	.	.	...	.
.	.	.	.	...	.

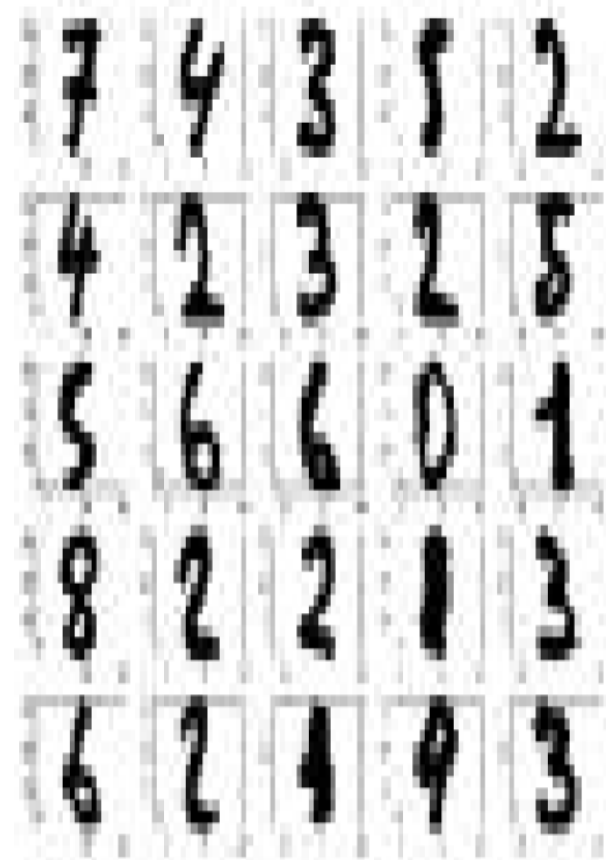
Expression Microarray Data Set



# Other Types of High-Dimensional Data



Face images



Handwritten digits

# Major Techniques of Dimensionality Reduction

- Feature selection
  - Definition
  - Objectives
- Feature Extraction (reduction)
  - Definition
  - Objectives
- Differences between the two techniques

# Feature Selection

## ■ Definition

- A process that chooses an optimal subset of features according to a objective function

## ■ Objectives

- To reduce dimensionality and remove noise
- To improve mining performance
  - Speed of learning
  - Predictive accuracy
  - Simplicity and comprehensibility of mined results

# Feature Extraction

- Feature reduction refers to the mapping of the original high-dimensional data onto a lower-dimensional space
- Given a set of data points of  $p$  variables  $\{x_1, x_2, \dots, x_n\}$   
Compute their low-dimensional representation:

$$x_i \in \mathbb{R}^d \rightarrow y_i \in \mathbb{R}^p \ (p \ll d)$$

- Criterion for feature reduction can be different based on different problem settings.
  - Unsupervised setting: minimize the information loss
  - Supervised setting: maximize the class discrimination

# Feature Reduction vs. Feature Selection

- Feature reduction
  - All original features are used
  - The transformed features are linear combinations of the original features
- Feature selection
  - Only a subset of the original features are selected
- Continuous versus discrete

# Outline

- Introduction to dimensionality reduction
- Feature selection (part I)
  - Basics
  - Representative algorithms
  - Recent advances
  - Applications
- Feature extraction (part II)
- Recent trends in dimensionality reduction

# Basics

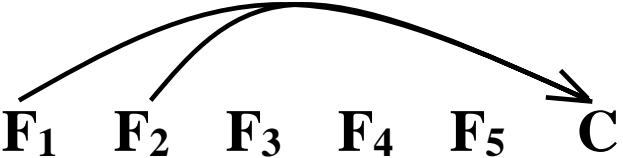
- Definitions of subset optimality
- Perspectives of feature selection
  - Subset search and feature ranking
  - Feature/subset evaluation measures
  - Models: filter vs. wrapper
  - Results validation and evaluation

# Subset Optimality for Classification

- A minimum subset that is sufficient to construct a hypothesis consistent with the training examples (*Almuallim and Dietterich, AAAI, 1991*)
  - Optimality is based on training set
  - The optimal set may overfit the training data
- A minimum subset  $G$  such that  $P(C|G)$  is equal or as close as possible to  $P(C|F)$  (*Koller and Sahami, ICML, 1996*)
  - Optimality is based on the entire population
  - Only training part of the data is available



# An Example for Optimal Subset



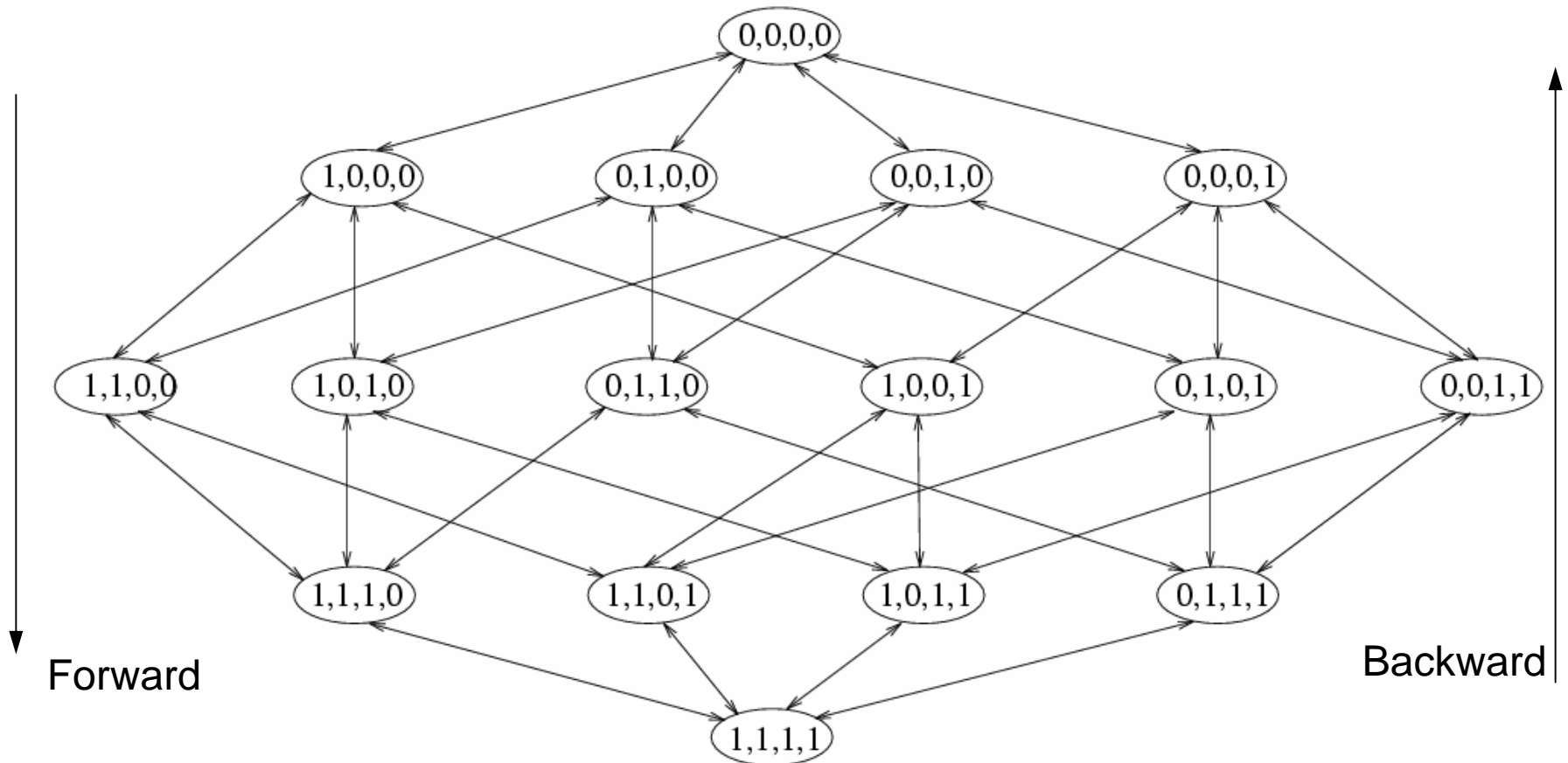
A diagram above the table shows three curved arrows pointing from the feature columns  $F_1$ ,  $F_2$ , and  $F_5$  to the target column  $C$ , indicating that these features are used to determine the value of  $C$ .

$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$C$
0	0	1	0	1	0
0	1	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
0	0	1	1	0	0
0	1	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1

- Data set (whole set)
  - Five Boolean features
  - $C = F_1 \vee F_2$
  - $F_3 = \neg F_2$ ,  $F_5 = \neg F_4$
  - Optimal subset:  
 $\{F_1, F_2\}$  or  $\{F_1, F_3\}$
- Combinatorial nature of searching for an optimal subset

# A Subset Search Problem

- An example of search space (*Kohavi & John 1997*)



# Different Aspects of Search

- Search starting points
  - Empty set
  - Full set
  - Random point
- Search directions
  - Sequential forward selection
  - Sequential backward elimination
  - Bidirectional generation
  - Random generation

# Different Aspects of Search (Cont'd)

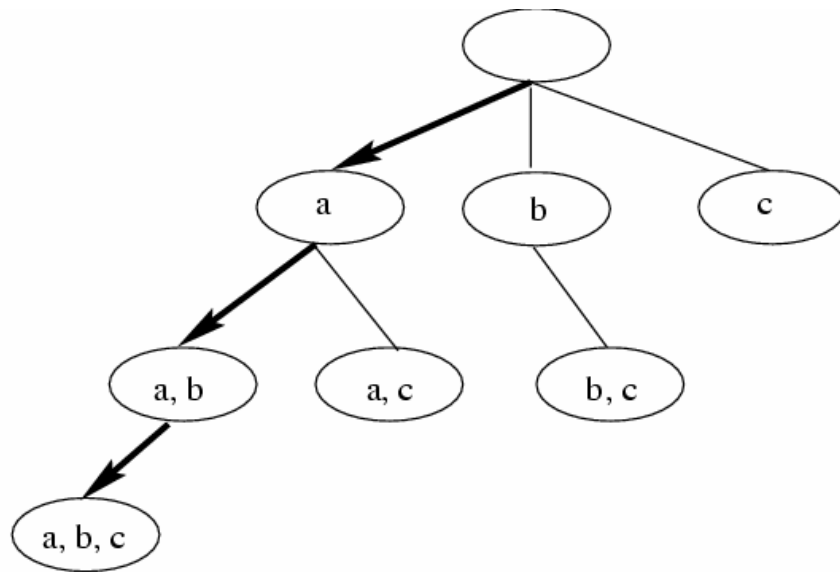
## ■ Search Strategies

- ❑ Exhaustive/complete search
- ❑ Heuristic search
- ❑ Nondeterministic search

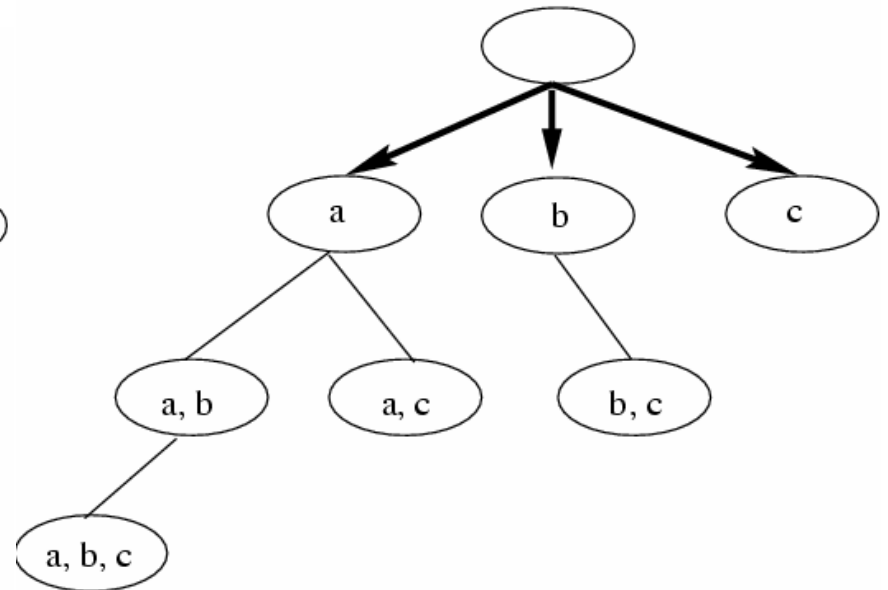
## ■ Combining search directions and strategies

Search Direction	Search Strategy		
	Complete	Heuristic	Nondeterministic
SFG	✓	✓	✗
SBG	✓	✓	✗
BG	✓	✓	✗
RG	✗	✓	✓

# Illustrations of Search Strategies



**Depth-first search**



**Breadth-first search**

# Feature Ranking

- Weighting and ranking individual features
- Selecting top-ranked ones for feature selection
- Advantages
  - Efficient:  $O(N)$  in terms of dimensionality  $N$
  - Easy to implement
- Disadvantages
  - Hard to determine the threshold
  - Unable to consider correlation between features

# Evaluation Measures for Ranking and Selecting Features

- The goodness of a feature/feature subset is dependent on measures
- Various measures
  - Information measures (Yu & Liu 2004, Jebara & Jaakkola 2000)
  - Distance measures (Robnik & Kononenko 03, Pudil & Novovicov 98)
  - Dependence measures (Hall 2000, Modrzejewski 1993)
  - Consistency measures (Almuallim & Dietterich 94, Dash & Liu 03)
  - Accuracy measures (Dash & Liu 2000, Kohavi&John 1997)

# Illustrative Data Set

	Hair	Height	Weight	Lotion	Result
$i_1$	1	2	1	0	1
$i_2$	1	3	2	1	0
$i_3$	2	1	2	1	0
$i_4$	1	1	2	0	1
$i_5$	3	2	3	0	1
$i_6$	2	3	3	0	0
$i_7$	2	2	3	0	0
$i_8$	1	1	1	1	0

**Sunburn data**

	Result (Sunburn)	
	No	Yes
$P(\text{Result})$	$5/8$	$3/8$
$P(\text{Hair}=1 \text{Result})$	$2/5$	$2/3$
$P(\text{Hair}=2 \text{Result})$	$3/5$	0
$P(\text{Hair}=3 \text{Result})$	0	$1/3$
$P(\text{Height}=1 \text{Result})$	$2/5$	$1/3$
$P(\text{Height}=2 \text{Result})$	$1/5$	$2/3$
$P(\text{Height}=3 \text{Result})$	$2/5$	0
$P(\text{Weight}=1 \text{Result})$	$1/5$	$1/3$
$P(\text{Weight}=2 \text{Result})$	$2/5$	$1/3$
$P(\text{Weight}=3 \text{Result})$	$2/5$	$1/3$
$P(\text{Lotion}=0 \text{Result})$	$2/5$	$3/3$
$P(\text{Lotion}=1 \text{Result})$	$3/5$	0

**Priors and class conditional probabilities**



# Information Measures

- Entropy of variable  $X$

$$H(X) = - \sum_i P(x_i) \log_2(P(x_i))$$

- Entropy of  $X$  after observing  $Y$

$$H(X|Y) = - \sum_j P(y_j) \sum_i P(x_i|y_j) \log_2(P(x_i|y_j))$$

- Information Gain

$$IG(X|Y) = H(X) - H(X|Y)$$

# Consistency Measures

- Consistency measures
  - Trying to find a minimum number of features that separate classes as consistently as the full set can
  - An inconsistency is defined as two instances having the same feature values but different classes
    - E.g., one inconsistency is found between instances i4 and i8 if we just look at the first two columns of the data table (Slide 24)

# Accuracy Measures

- Using classification accuracy of a classifier as an evaluation measure
- Factors constraining the choice of measures
  - Classifier being used
  - The speed of building the classifier
- Compared with previous measures
  - Directly aimed to improve accuracy
  - Biased toward the classifier being used
  - More time consuming

# Models of Feature Selection

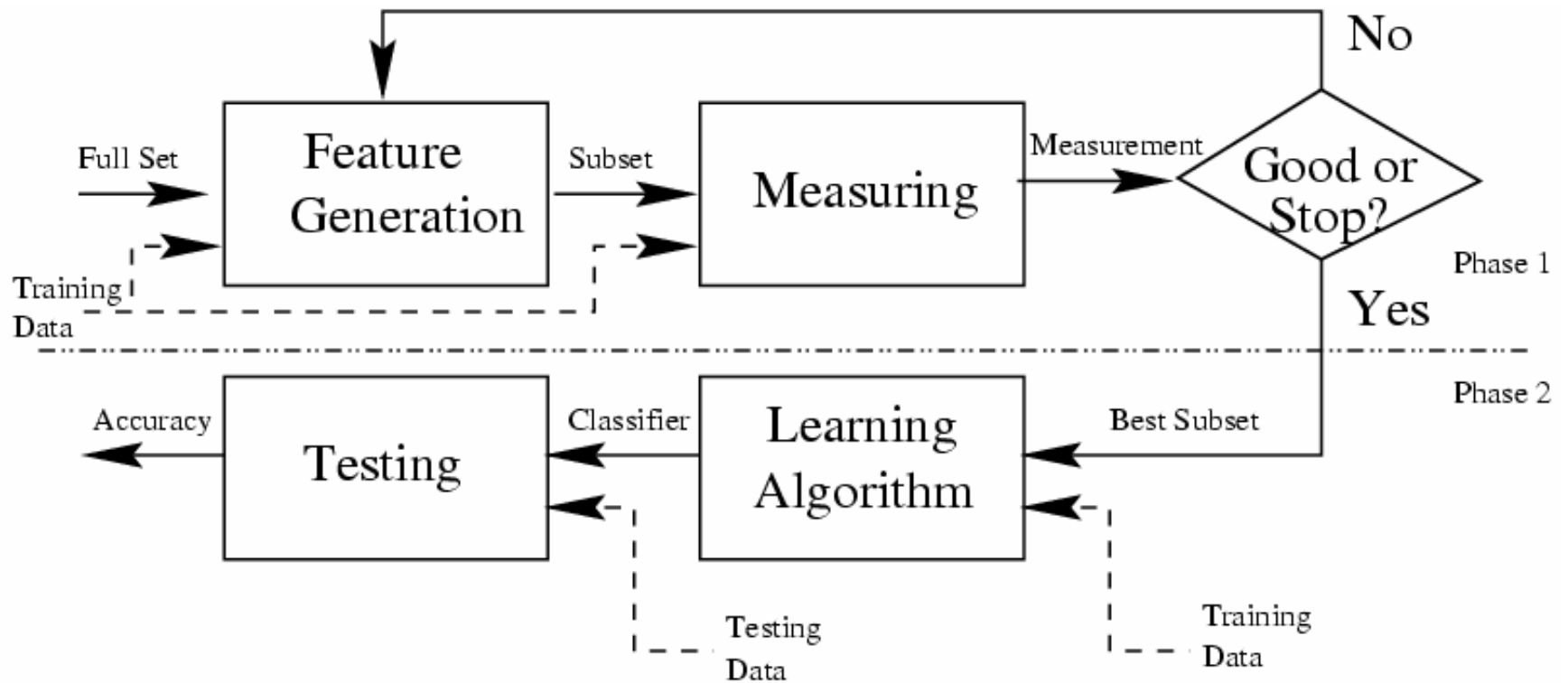
## ■ Filter model

- ❑ Separating feature selection from classifier learning
- ❑ Relying on general characteristics of data (*information, distance, dependence, consistency*)
- ❑ No bias toward any learning algorithm, fast

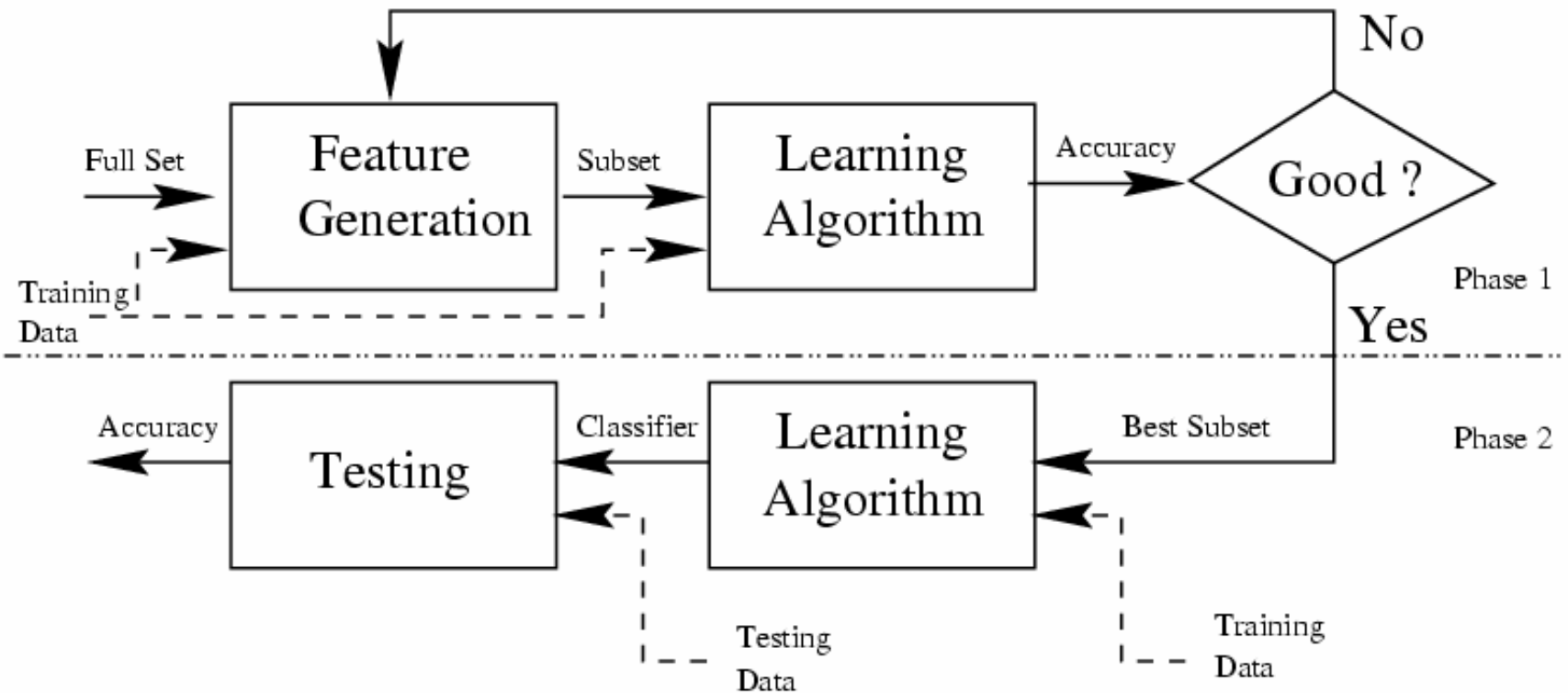
## ■ Wrapper model

- ❑ Relying on a predetermined classification algorithm
- ❑ Using predictive accuracy as goodness measure
- ❑ High accuracy, computationally expensive

# Filter Model



# Wrapper Model

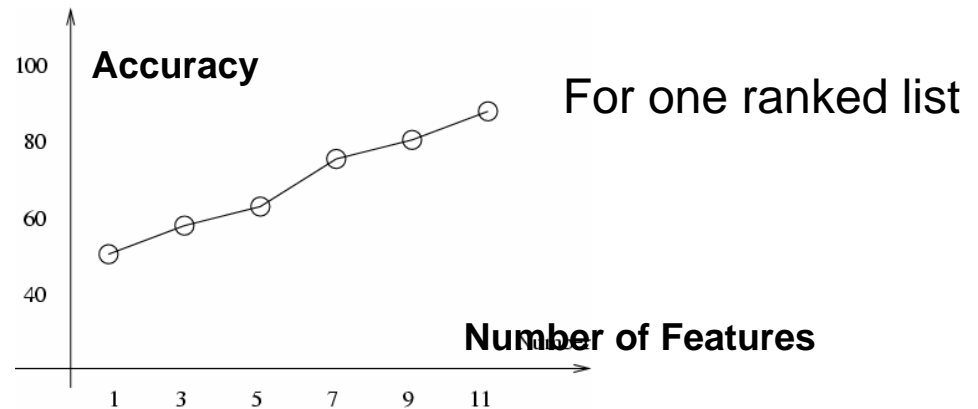


# How to Validate Selection Results

- Direct evaluation (if we know *a priori* ...)
  - Often suitable for artificial data sets
  - Based on prior knowledge about data
- Indirect evaluation (if we don't know ...)
  - Often suitable for real-world data sets
  - Based on **a)** number of features selected, **b)** performance on selected features (e.g., predictive accuracy, goodness of resulting clusters), and **c)** speed

(Liu & Motoda 1998)

# Methods for Result Evaluation



- Learning curves
  - For results in the form of a ranked list of features
- Before-and-after comparison
  - For results in the form of a minimum subset
- Comparison using different classifiers
  - To avoid learning bias of a particular classifier
- Repeating experimental results
  - For non-deterministic results



# Representative Algorithms for Classification

## ■ Filter algorithms

### □ Feature ranking algorithms

- Example: Relief (*Kira & Rendell 1992*)

### □ Subset search algorithms

- Example: consistency-based algorithms
  - Focus (*Almuallim & Dietterich, 1994*)

## ■ Wrapper algorithms

### □ Feature ranking algorithms

- Example: SVM

### □ Subset search algorithms

- Example: RFE

# Relief Algorithm

## Relief

**Input:**  $\mathbf{x}$  - features

$m$  - number of instances sampled

$\tau$  - adjustable relevance threshold

**initialize:**  $\mathbf{w} = 0$

**for**  $i = 1$  to  $m$

**begin**

    randomly select an instance  $I$

    find nearest-hit  $H$  and nearest-miss  $J$

**for**  $j = 1$  to  $N$

$\mathbf{w}(j) = \mathbf{w}(j) - \text{diff}(j, I, H)^2/m + \text{diff}(j, I, J)^2/m$

**end**

**Output:**  $\mathbf{w}$  greater than  $\tau$

# Focus Algorithm

## Focus

**Input:**  $F$  - all features  $x$  in data  $D$   
 $U$  - inconsistency rate as evaluation measure

**initialize:**  $S = \{\}$

**for**  $i = 1$  to  $N$

**for** each subset  $S$  of size  $i$

**if**  $\text{Cal}U(S, D) = 0$     */\* CalU(S, D) returns inconsistency\*/*

**return**  $S$

**Output:**  $S$  - a minimum subset that satisfies  $U$

# Representative Algorithms for Clustering

- Filter algorithms

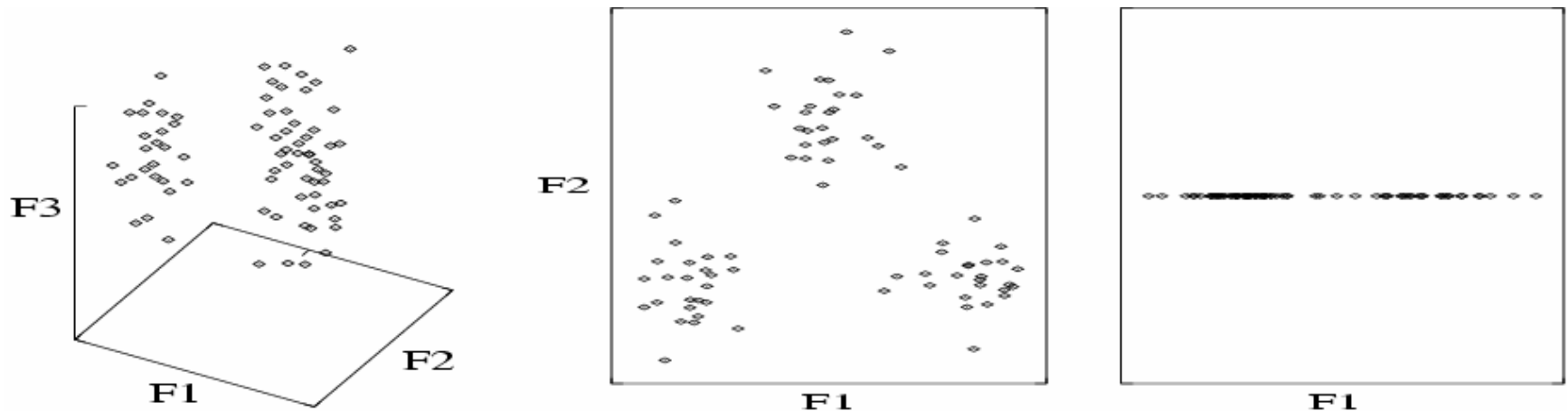
- Example: a filter algorithm based on entropy measure (*Dash et al., ICDM, 2002*)

- Wrapper algorithms

- Example: FSSEM – a wrapper algorithm based on EM (expectation maximization) clustering algorithm (*Dy and Brodley, ICML, 2000*)

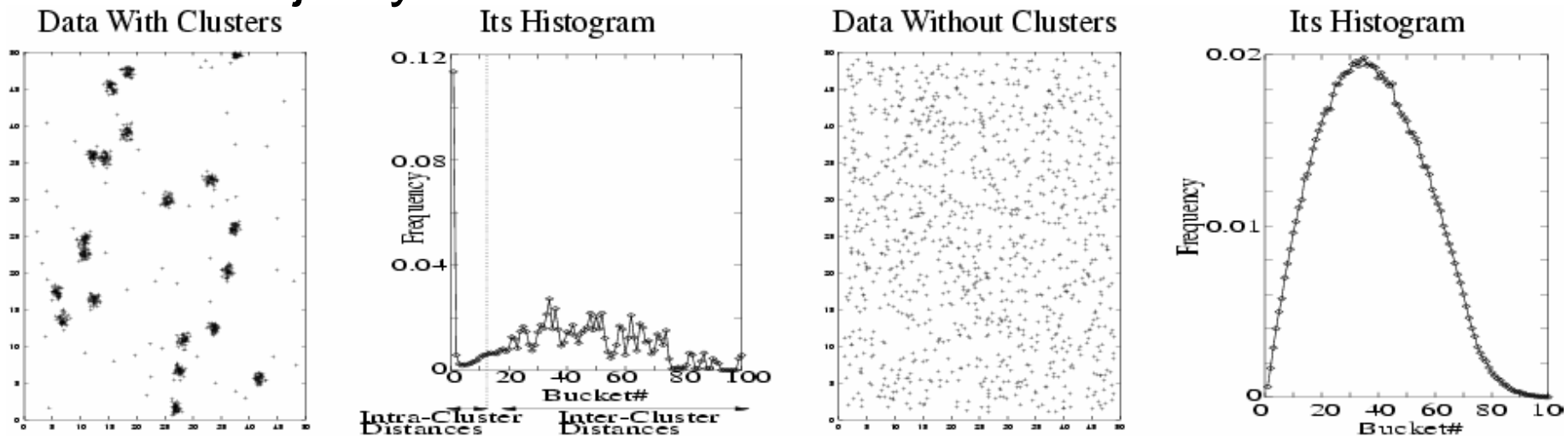
# Effect of Features on Clustering

- Example from (*Dash et al., ICDM, 2002*)
- Synthetic data in (3,2,1)-dimensional spaces
  - 75 points in three dimensions
  - Three clusters in F1-F2 dimensions
  - Each cluster having 25 points



# Two Different Distance Histograms of Data

- Example from (*Dash et al., ICDM, 2002*)
- Synthetic data in 2-dimensional space
  - Histograms record point-point distances
  - For data with 20 clusters (left), the majority of the intra-cluster distances are smaller than the majority of the inter-cluster distances



# An Entropy based Filter Algorithm

## ■ Basic ideas

- When clusters are very distinct, intra-cluster and inter-cluster distances are quite distinguishable
- Entropy is low if data has distinct clusters and high otherwise

## ■ Entropy measure

- Substituting probability with distance  $D_{ij}$
- Entropy is 0.0 for minimum distance 0.0 or maximum 1.0 and is 1.0 for the mean distance 0.5

$$E = - \sum_{X_i} \sum_{X_j} [D_{ij} \log D_{ij} + (1 - D_{ij}) \log(1 - D_{ij})]$$

# FSSEM Algorithm

## ■ EM Clustering

- ❑ To estimate the maximum likelihood mixture model parameters and the cluster probabilities of each data point
- ❑ Each data point belongs to every cluster with some probability

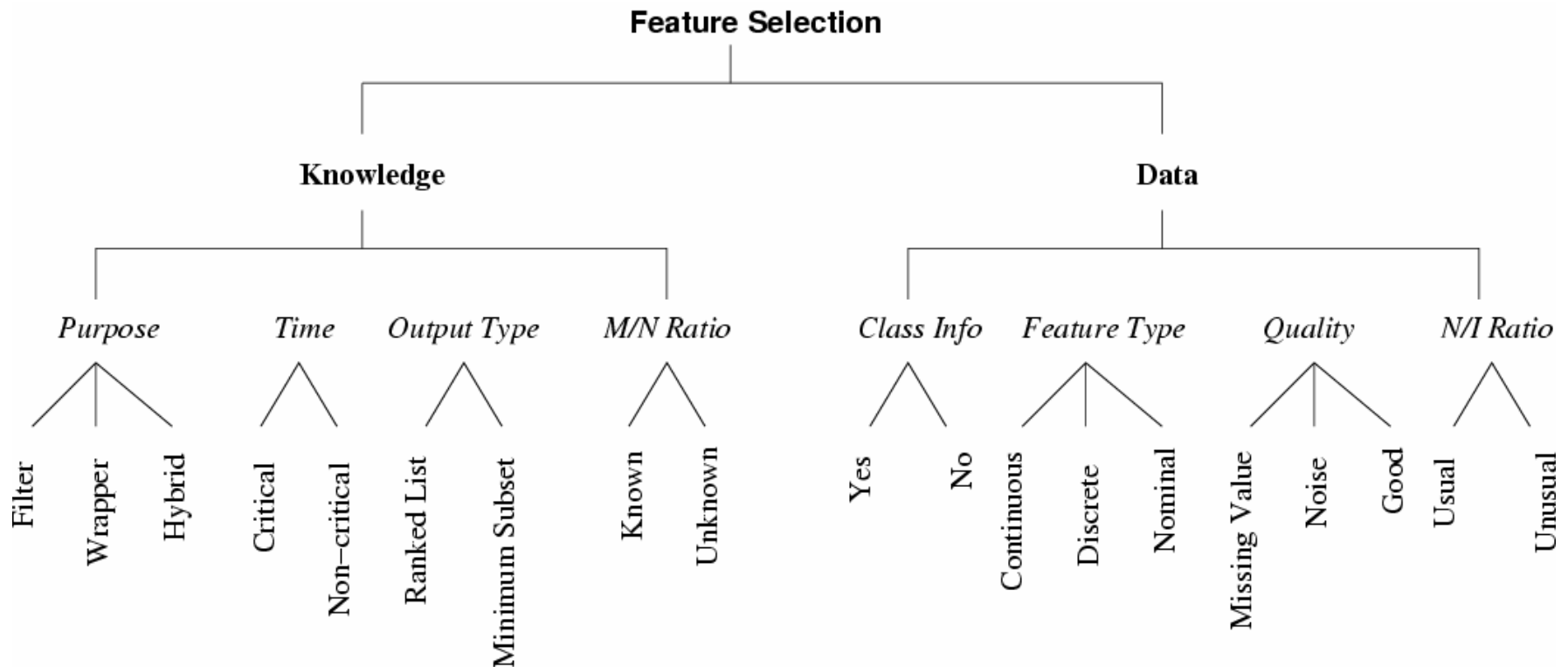
## ■ Feature selection for EM

- ❑ Searching through feature subsets
- ❑ Applying EM on each candidate subset
- ❑ Evaluating goodness of each candidate subset based on the goodness of resulting clusters



# Guideline for Selecting Algorithms

- A unifying platform (Liu and Yu 2005)



# Handling High-dimensional Data

- High-dimensional data

- As in gene expression microarray analysis, text categorization, ...
- With hundreds to tens of thousands of features
- With many irrelevant and redundant features

- Recent research results

- Redundancy based feature selection
  - *Yu and Liu, ICML-2003, JMLR-2004*

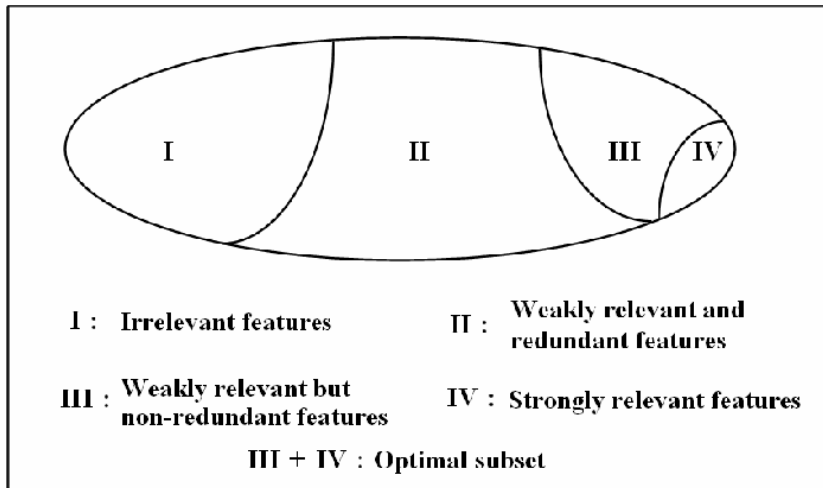
# Limitations of Existing Methods

- Individual feature evaluation
  - Focusing on identifying relevant features without handling feature redundancy
  - Time complexity:  $O(N)$
- Feature subset evaluation
  - Relying on minimum feature subset heuristics to implicitly handling redundancy while pursuing relevant features
  - Time complexity: at least  $O(N^2)$

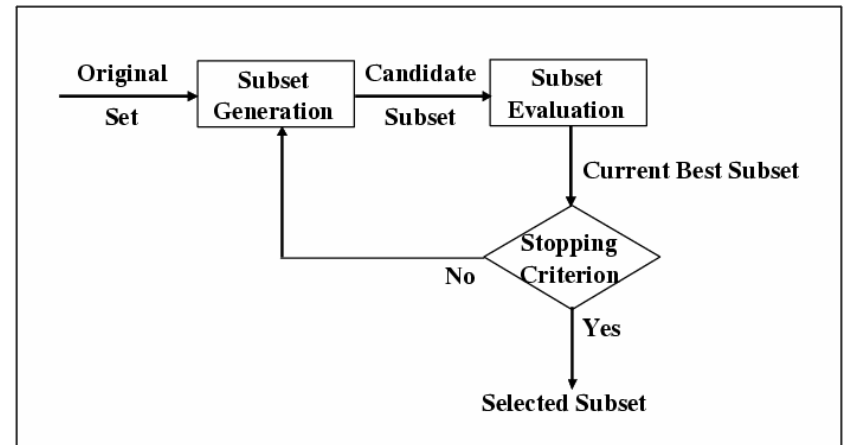
# Goals

- High effectiveness
  - Able to handle both irrelevant and redundant features
  - Not pure individual feature evaluation
- High efficiency
  - Less costly than existing subset evaluation methods
  - Not traditional heuristic search methods

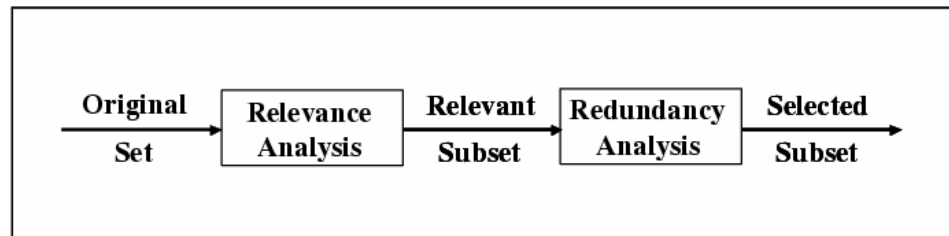
# Our Solution – A New Framework of Feature Selection



A view of feature relevance and redundancy



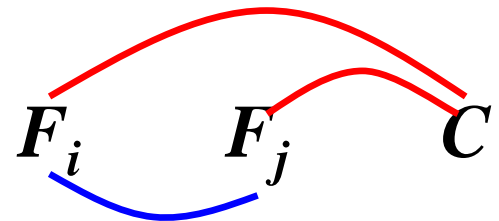
A traditional framework of feature selection



A new framework of feature selection

# Approximation

- Reasons for approximation
  - Searching for an optimal subset is combinatorial
  - Over-searching on training data can cause over-fitting
- Two steps of approximation
  - To approximately find the set of relevant features
  - To approximately determine feature redundancy among relevant features
- Correlation-based measure
  - C-correlation (feature  $F_i$  and class  $C$ )
  - F-correlation (feature  $F_i$  and  $F_j$  )



# Determining Redundancy

- Hard to decide redundancy

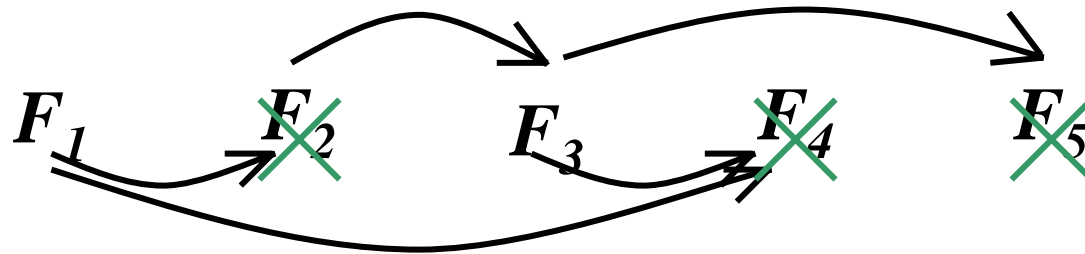
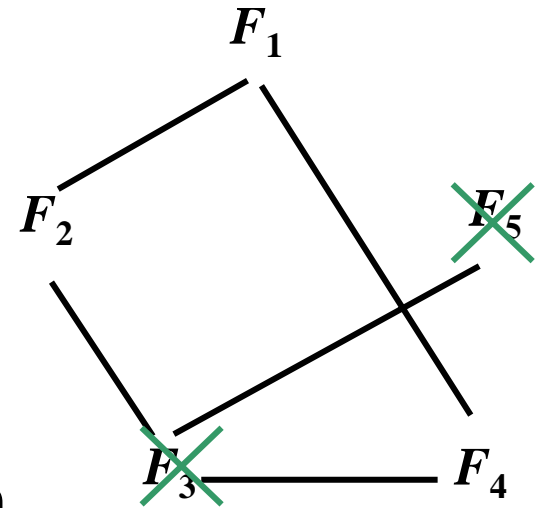
- Redundancy criterion
  - Which one to keep

- Approximate redundancy criterion

$F_j$  is redundant to  $F_i$  iff

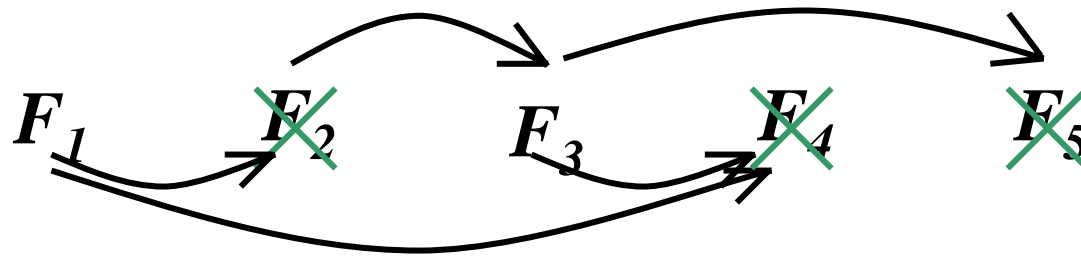
$$SU(F_i, C) \geq SU(F_j, C) \text{ and } SU(F_i, F_j) \geq SU(F_j, C)$$

- Predominant feature: not redundant to any feature in the current set



# FCBF (Fast Correlation-Based Filter)

- Step 1: Calculate  $SU$  value for each feature, order them, select relevant features based on a threshold
- Step 2: Start with the first feature to eliminate all features that are redundant to it
- Repeat Step 2 with the next remaining feature until the end of list



- **Step 1:**  $O(N)$
- **Step 2:** average case  $O(N \log N)$



# Real-World Applications

- Customer relationship management
  - Ng and Liu, 2000 (**NUS**)
- Text categorization
  - Yang and Pederson, 1997 (**CMU**)
  - Forman, 2003 (**HP Labs**)
- Image retrieval
  - Swets and Weng, 1995 (**MSU**)
  - Dy *et al.*, 2003 (**Purdue University**)
- Gene expression microarray data analysis
  - Golub *et al.*, 1999 (**MIT**)
  - Xing *et al.*, 2001 (**UC Berkeley**)
- Intrusion detection
  - Lee *et al.*, 2000 (**Columbia University**)

# Text Categorization

- Text categorization
  - Automatically assigning predefined categories to new text documents
  - Of great importance given massive on-line text from WWW, Emails, digital libraries...
- Difficulty from high-dimensionality
  - Each unique term (word or phrase) representing a feature in the original feature space
  - Hundreds or thousands of unique terms for even a moderate-sized text collection
- Desirable to reduce the feature space without sacrificing categorization accuracy

# Feature Selection in Text Categorization

- A comparative study in (*Yang and Pederson, ICML, 1997*)
  - 5 metrics evaluated and compared
    - Document Frequency (DF), Information Gain (IG), Mutual Information (MI),  $\chi^2$  statistics (CHI), Term Strength (TS)
    - IG and CHI performed the best
  - Improved classification accuracy of  $k$ -NN achieved after removal of up to 98% unique terms by IG
- Another study in (*Forman, JMLR, 2003*)
  - 12 metrics evaluated on 229 categorization problems
  - A new metric, Bi-Normal Separation, outperformed others and improved accuracy of SVMs

# Content-Based Image Retrieval (CBIR)

## ■ Image retrieval

- ❑ An explosion of image collections from scientific, civil, military equipments
- ❑ Necessary to index the images for efficient retrieval

## ■ Content-based image retrieval (CBIR)

- ❑ Instead of indexing images based on textual descriptions (e.g., keywords, captions)
- ❑ Indexing images based on visual contents (e.g., color, texture, shape)

## ■ Traditional methods for CBIR

- ❑ Using all indexes (features) to compare images
- ❑ Hard to scale to large size image collections

# Feature Selection in CBIR

- An application in (*Swets and Weng, ISCV, 1995*)
  - A large database of widely varying real-world objects in natural settings
  - Selecting relevant features to index images for efficient retrieval
- Another application in (*Dy et al., Trans. PRMI, 2003*)
  - A database of high resolution computed tomography lung images
  - FSSEM algorithm applied to select critical characterizing features
  - Retrieval precision improved based on selected features

# Gene Expression Microarray Analysis

## ■ Microarray technology

- Enabling simultaneously measuring the expression levels for thousands of genes in a single experiment
- Providing new opportunities and challenges for data mining

## ■ Microarray data

<div>Gene</div> <div>Sample</div>	M23197_at	U66497_at	M92287_at	...	Class
Sample 1	261	88	4778	...	ALL
Sample 2	101	74	2700	...	ALL
Sample 3	1450	34	498	...	AML
.	.	.	.	...	.
.	.	.	.	...	.
.	.	.	.	...	.

# Motivation for Gene (Feature) Selection

## ■ Data mining tasks

Data points are:

Genes

Samples

Clustering

Grouping similar genes together to find co-regulated genes

Grouping similar samples together to find classes or subclasses

Classification

Building a classifier to predict the classes of new samples

## ■ Data characteristics in sample classification

- ❑ High dimensionality (thousands of genes)
- ❑ Small sample size (often less than 100 samples)

## ■ Problems

- ❑ Curse of dimensionality
- ❑ Overfitting the training data

# Feature Selection in Sample Classification

- An application in (*Golub, Science, 1999*)
  - On leukemia data (7129 genes, 72 samples)
  - Feature ranking method based on linear correlation
  - Classification accuracy improved by 50 top genes
- Another application in (*Xing et al., ICML, 2001*)
  - A hybrid of filter and wrapper method
    - Selecting best subset of each cardinality based on information gain ranking and Markov blanket filtering
    - Comparing between subsets of the same cardinality using cross-validation
  - Accuracy improvements observed on the same leukemia data



# Intrusion Detection via Data Mining

- Network-based computer systems
  - Playing increasingly vital roles in modern society
  - Targets of attacks from enemies and criminals
- Intrusion detection is one way to protect computer systems
- A data mining framework for intrusion detection in (*Lee et al., AI Review, 2000*)
  - Audit data analyzed using data mining algorithms to obtain frequent activity patterns
  - Classifiers based on selected features used to classify an observed system activity as “legitimate” or “intrusive”

---

# Dimensionality Reduction for Data Mining

## - Techniques, Applications and Trends

### (Part II)

---

**Lei Yu**  
**Binghamton University**



**Jieping Ye, Huan Liu**  
**Arizona State University**



# Outline

- Introduction to dimensionality reduction
- Feature selection (part I)
- Feature extraction (part II)
  - Basics
  - Representative algorithms
  - Recent advances
  - Applications
- Recent trends in dimensionality reduction

# Feature Reduction Algorithms

## ■ Unsupervised

- ❑ Latent Semantic Indexing (LSI): truncated SVD
- ❑ Independent Component Analysis (ICA)
- ❑ Principal Component Analysis (PCA)
- ❑ Manifold learning algorithms

## ■ Supervised

- ❑ Linear Discriminant Analysis (LDA)
- ❑ Canonical Correlation Analysis (CCA)
- ❑ Partial Least Squares (PLS)

## ■ Semi-supervised

# Feature Reduction Algorithms

## ■ Linear

- ❑ Latent Semantic Indexing (LSI): truncated SVD
- ❑ Principal Component Analysis (PCA)
- ❑ Linear Discriminant Analysis (LDA)
- ❑ Canonical Correlation Analysis (CCA)
- ❑ Partial Least Squares (PLS)

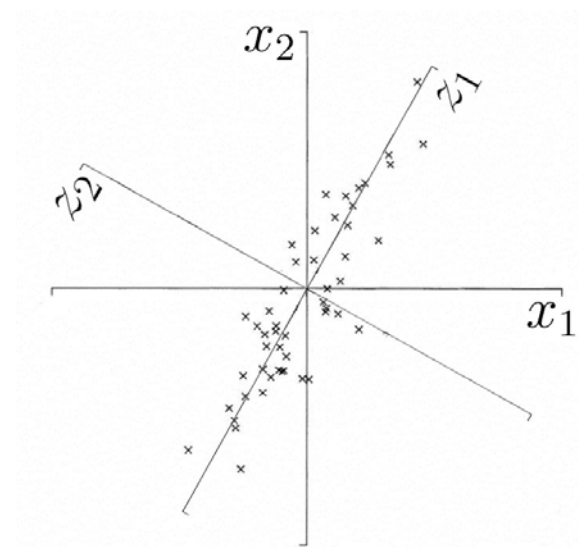
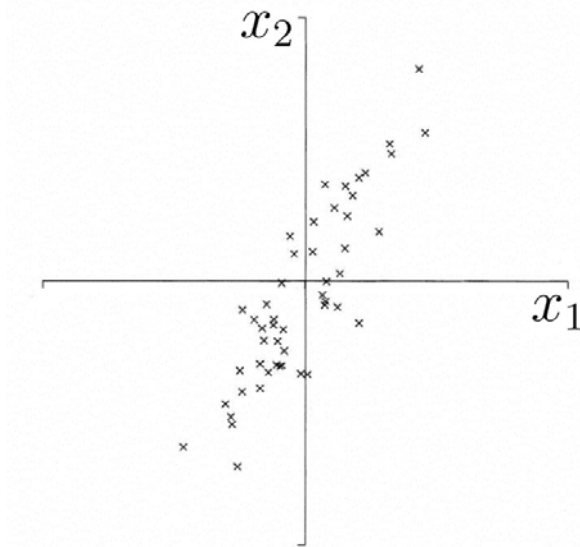
## ■ Nonlinear

- ❑ Nonlinear feature reduction using **kernels**
- ❑ **Manifold learning**

# Principal Component Analysis

- Principal component analysis (PCA)
  - Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables
  - Retains most of the sample's information.
- By information we mean the variation present in the sample, given by the correlations between the original variables.
  - The new variables, called principal components (PCs), are **uncorrelated**, and are ordered by the fraction of the total information each retains.

# Geometric Picture of Principal Components (PCs)



- the 1<sup>st</sup> PC  $z_1$  is a minimum distance fit to a line in  $X$  space
- the 2<sup>nd</sup> PC  $z_2$  is a minimum distance fit to a line in the plane perpendicular to the 1<sup>st</sup> PC

PCs are a series of linear least squares fits to a sample, each orthogonal to all the previous.

# Algebraic Derivation of PCs

- Main steps for computing PCs
  - Form the covariance matrix  $S$ .
  - Compute its eigenvectors:  $\{a_i\}_{i=1}^d$
  - The first  $p$  eigenvectors  $\{a_i\}_{i=1}^p$  form the  $p$  PCs.  $G \leftarrow [a_1, a_2, \dots, a_p]$
  - The transformation  $G$  consists of the  $p$  PCs.

A testpoint  $x \in \mathbb{R}^d \rightarrow G^T x \in \mathbb{R}^p$ .




# Optimality Property of PCA

## Main theoretical result:

The matrix  $G$  consisting of the first  $p$  eigenvectors of the covariance matrix  $S$  solves the following min problem:

$$\min_{G \in \mathbb{R}^{d \times p}} \|X - G(G^T X)\|_F^2 \text{ subject to } G^T G = I_p$$


$$\|X - \bar{X}\|_F^2$$

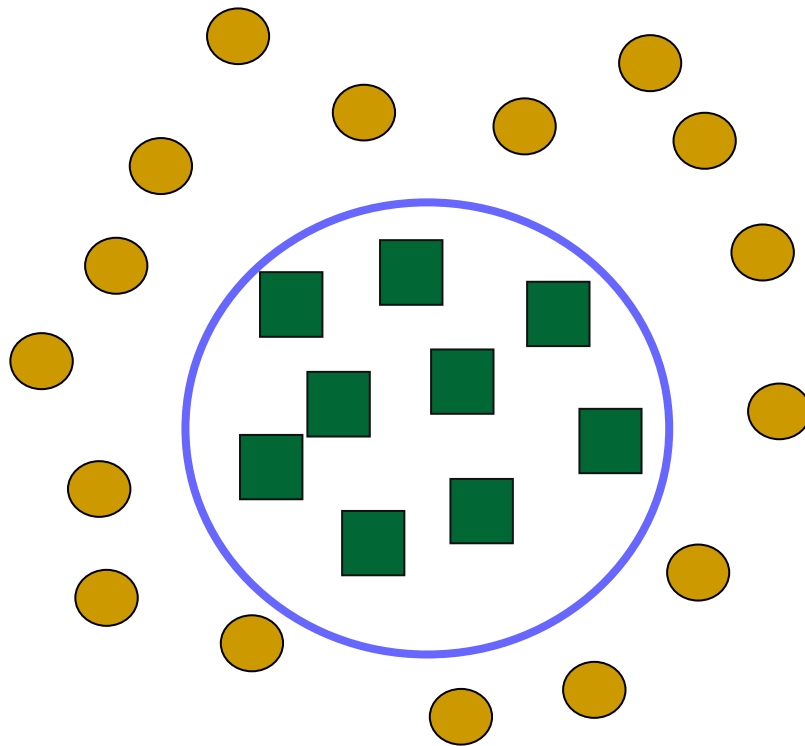
reconstruction error

PCA projection minimizes the reconstruction error among all linear projections of size  $p$ .

# Applications of PCA

- *Eigenfaces for recognition.* Turk and Pentland. 1991.
- *Principal Component Analysis for clustering gene expression data.* Yeung and Ruzzo. 2001.
- *Probabilistic Disease Classification of Expression-Dependent Proteomic Data from Mass Spectrometry of Human Serum.* Lilien. 2003.

# Motivation for Non-linear PCA using Kernels



Linear projections  
will not detect the  
pattern.

# Nonlinear PCA using Kernels

- Traditional PCA applies linear transformation
  - May not be effective for nonlinear data
- Solution: apply nonlinear transformation to potentially very high-dimensional space.

$$\phi : x \rightarrow \phi(x)$$

- Computational efficiency: apply the kernel trick.
  - Require PCA can be rewritten in terms of dot product.
$$K(x_i, x_j) = \phi(x_i) \bullet \phi(x_j)$$

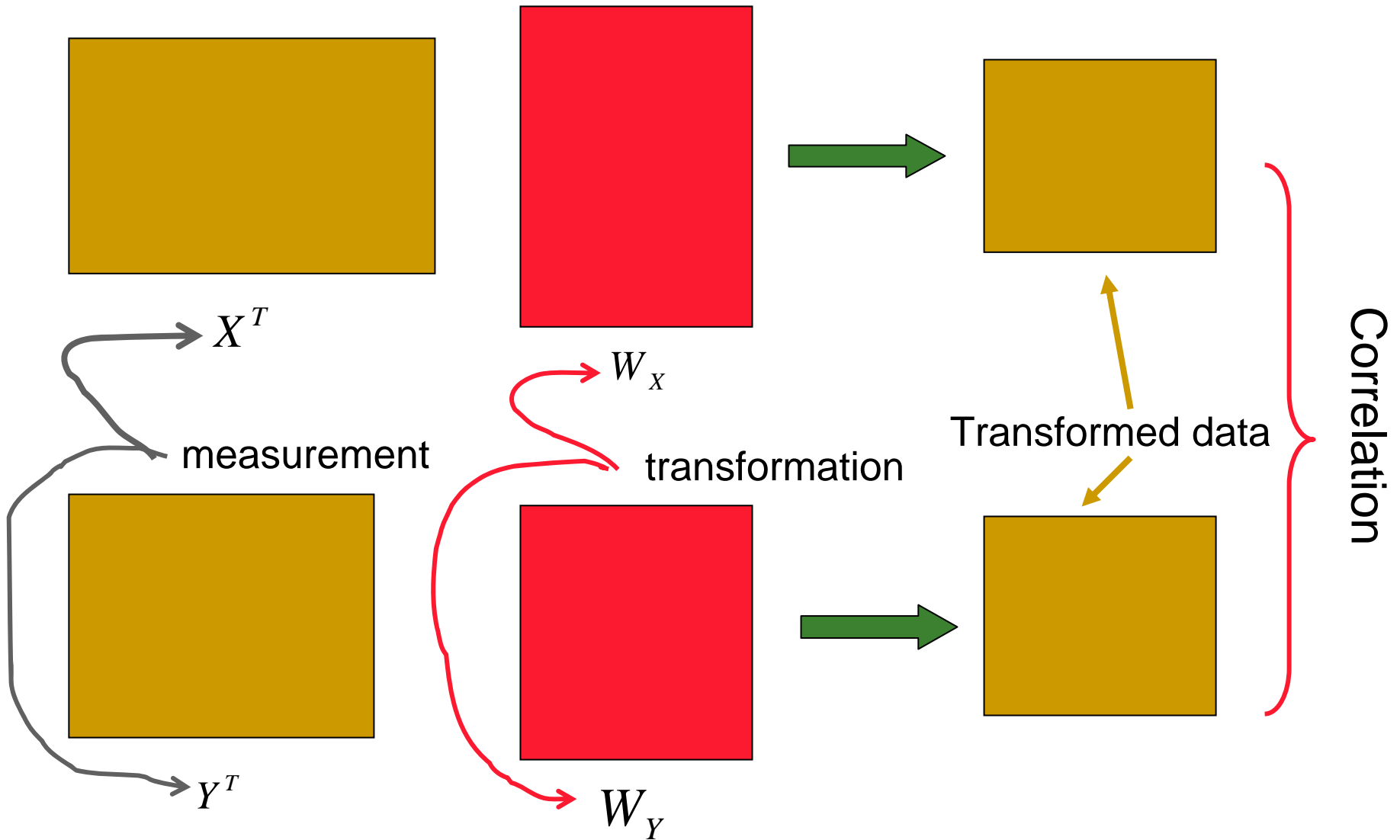
# Canonical Correlation Analysis (CCA)

- CCA was developed first by H. Hotelling.
  - H. Hotelling. Relations between two sets of variates.  
*Biometrika*, 28:321-377, 1936.
- CCA measures the linear relationship between two multidimensional variables.
- CCA finds two bases, one for each variable, that are optimal with respect to correlations.
- Applications in economics, medical studies, bioinformatics and other areas.

# Canonical Correlation Analysis (CCA)

- Two multidimensional variables
  - Two different measurement on the same set of objects
    - Web images and associated text
    - Protein (or gene) sequences and related literature (text)
    - Protein sequence and corresponding gene expression
    - In classification: feature vector and class label
  - Two measurements on the same object are likely to be correlated.
    - May not be obvious on the original measurements.
    - Find the maximum correlation on transformed space.

# Canonical Correlation Analysis (CCA)



# Problem Definition

- Find two sets of basis vectors, one for  $\mathbf{x}$  and the other for  $\mathbf{y}$ , such that the correlations between the *projections* of the variables onto these basis vectors are maximized.

Given  $S = ((\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n))$  of  $(\mathbf{x}, \mathbf{y})$

Compute two basis vectors  $\mathbf{w}_x$  and  $\mathbf{w}_y$  :

$$\mathbf{x} \rightarrow \langle \mathbf{w}_x, \mathbf{x} \rangle \quad S_{x, \mathbf{w}_x} = (\langle \mathbf{w}_x, \mathbf{x}_1 \rangle, \dots, \langle \mathbf{w}_x, \mathbf{x}_n \rangle)$$

$$\mathbf{y} \rightarrow \langle \mathbf{w}_y, \mathbf{y} \rangle \quad S_{y, \mathbf{w}_y} = (\langle \mathbf{w}_y, \mathbf{y}_1 \rangle, \dots, \langle \mathbf{w}_y, \mathbf{y}_n \rangle)$$



# Problem Definition

- Compute the two basis vectors so that the correlations of the projections onto these vectors are maximized.

$$\begin{aligned}\rho &= \max_{\mathbf{w}_x, \mathbf{w}_y} \text{corr}(S_x \mathbf{w}_x, S_y \mathbf{w}_y) \\ &= \max_{\mathbf{w}_x, \mathbf{w}_y} \frac{\langle S_x \mathbf{w}_x, S_y \mathbf{w}_y \rangle}{\|S_x \mathbf{w}_x\| \|S_y \mathbf{w}_y\|}.\end{aligned}$$

# Algebraic Derivation of CCA

Now observe that the covariance matrix of  $(\mathbf{x}, \mathbf{y})$  is

$$C(\mathbf{x}, \mathbf{y}) = \hat{\mathbb{E}} \left[ \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}' \right] = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} = C.$$

The optimization problem is equivalent to

$$\rho = \max_{\mathbf{w}_x, \mathbf{w}_y} \frac{\mathbf{w}_x' C_{xy} \mathbf{w}_y}{\sqrt{\mathbf{w}_x' C_{xx} \mathbf{w}_x \mathbf{w}_y' C_{yy} \mathbf{w}_y}}.$$

where

$$C_{xy} = XY^T, C_{xx} = XX^T$$
$$C_{yx} = YX^T, C_{yy} = YY^T$$

# Algebraic Derivation of CCA

- In general, the k-th basis vectors are given by the k-th eigenvector of

$$C_{xy}C_{yy}^{-1}C_{yx}\mathbf{w}_x = \lambda^2 C_{xx}\mathbf{w}_x.$$

- The two transformations are given by

$$W_X = [w_{x1}, w_{x2}, \dots, w_{xp}]$$
$$W_Y = [w_{y1}, w_{y2}, \dots, w_{yp}]$$

# Nonlinear CCA using Kernels

Key: rewrite the CCA formulation in terms of inner products.

$$\left. \begin{aligned} C_{xx} &= XX^T \\ C_{xy} &= XY^T \\ w_x &= X\alpha \\ w_y &= Y\beta \end{aligned} \right\}$$

$$\rho = \max_{\alpha, \beta} \frac{\alpha^T X^T XY^T Y \beta}{\sqrt{\alpha^T X^T XX^T X \alpha} \sqrt{\beta^T Y^T YY^T Y \beta}}$$



Only inner  
products  
Appear

# Applications in Bioinformatics

- CCA can be extended to multiple views of the data
  - Multiple (larger than 2) data sources
- Two different ways to combine different data sources
  - Multiple CCA
    - Consider all pairwise correlations
  - Integrated CCA
    - Divide into two disjoint sources

# Applications in Bioinformatics

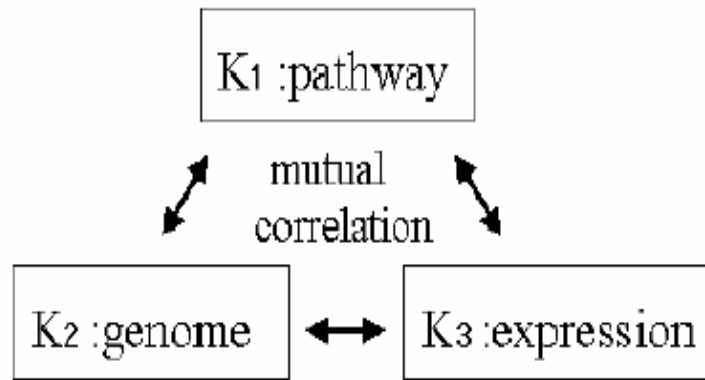


Fig. 1. Mutual correlation model in multiple KCCA (MKCCA).

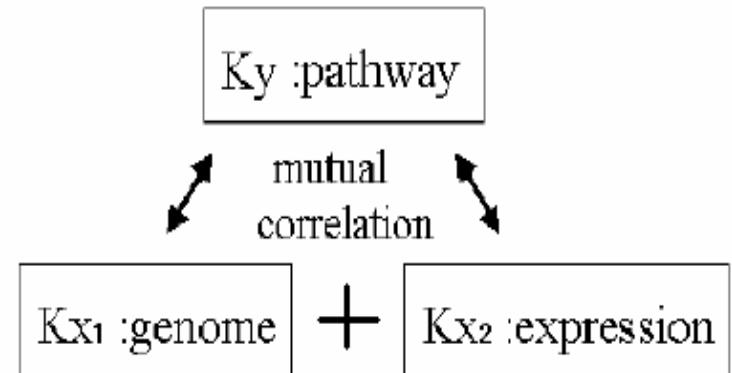


Fig. 2. Mutual correlation model in integrated KCCA (IKCCA).

$K_1$ ,  $K_2$  and  $K_3$  correspond to gene-gene similarities in pathways, genome, and expression.

Source: Extraction of Correlated Gene Clusters from Multiple Genomic Data by Generalized Kernel Canonical Correlation Analysis. *ISMB'03*

<http://cg.ensmp.fr/~vert/publi/ismb03/ismb03.pdf>

# Multidimensional scaling (MDS)

- MDS: Multidimensional scaling
  - Borg and Groenen, 1997
- MDS takes a matrix of **pair-wise distances** and gives a mapping to  $\mathbb{R}^d$ . It finds an embedding that preserves the **interpoint distances**, equivalent to PCA when those distance are Euclidean.
  - Low dimensional data for visualization

# Classical MDS

$D = \left( \|x_i - x_j\|^2 \right)_{ij}$  : distance matrix

$$\Rightarrow P^e D P^e = -2 \left( (x_i - \mu) \bullet (x_j - \mu) \right)_{ij}$$

Centering matrix :

$$P^e = I - \frac{1}{n} e e^T$$

$$P^e A P^e = \{(1 - Q)A(1 - Q)\}_{ij} = A_{ij} - A_{ij}^R - A_{ij}^C + A_{ij}^{RC} \quad (1.26)$$

where  $A^C \equiv A Q$  is the matrix  $A$  with each column replaced by the column mean,  $A^R \equiv Q A$  is  $A$  with each row replaced by the row mean, and  $A^{RC} \equiv Q A Q$  is  $A$  with every element replaced by the mean of all the elements.



# Classical MDS

**Theorem:** Consider the class of symmetric matrices  $A \in S_n$  such that  $A_{ij} \geq 0$  and  $A_{ii} = 0 \ \forall i, j$ . Then  $\bar{A} \equiv -P^e A P^e$  is positive semidefinite if and only if  $A$  is a distance matrix (with embedding space  $\mathcal{R}^d$  for some  $d$ ). Given that  $A$  is a distance matrix, the minimal embedding dimension  $d$  is the rank of  $\bar{A}$ , and the embedding vectors are any set of Gram vectors of  $\bar{A}$ , scaled by a factor of  $\frac{1}{\sqrt{2}}$ . ([Geometric Methods for Feature Extraction and Dimensional Reduction](#) – Burges, 2005)

$$D = \left( \|x_i - x_j\|^2 \right)_{ij} : \text{distance matrix} \Rightarrow P^e D P^e = -2 \left( (x_i - \mu) \bullet (x_j - \mu) \right)_{ij}$$

Problem : Given  $D$ , how to find  $x_i$  ?

$$-P^e D P^e / 2 = \bar{D} = U_d \Sigma_d U_d^T = \left( U_d \Sigma_d^{0.5} \right) \left( U_d \Sigma_d^{0.5} \right)^T$$

$\Rightarrow$  Choose  $x_i$ , for  $i = 1, \dots, n$ , from the rows of  $U_d \Sigma_d^{0.5}$

# Classical MDS

- If Euclidean distance is used in constructing  $D$ , MDS is equivalent to PCA.
- The dimension in the embedded space is  $d$ , if the rank equals to  $d$ .
- If only the first  $p$  eigenvalues are important (in terms of magnitude), we can truncate the eigen-decomposition and keep the first  $p$  eigenvalues only.
  - Approximation error

# Classical MDS

- So far, we focus on classical MDS, assuming  $D$  is the squared distance matrix.
  - Metric scaling
- How to deal with more general dissimilarity measures
  - Non-metric scaling

Metric scaling:  $-P^e D P^e = 2 \left( (x_i - \mu) \bullet (x_j - \mu) \right)_{ij}$

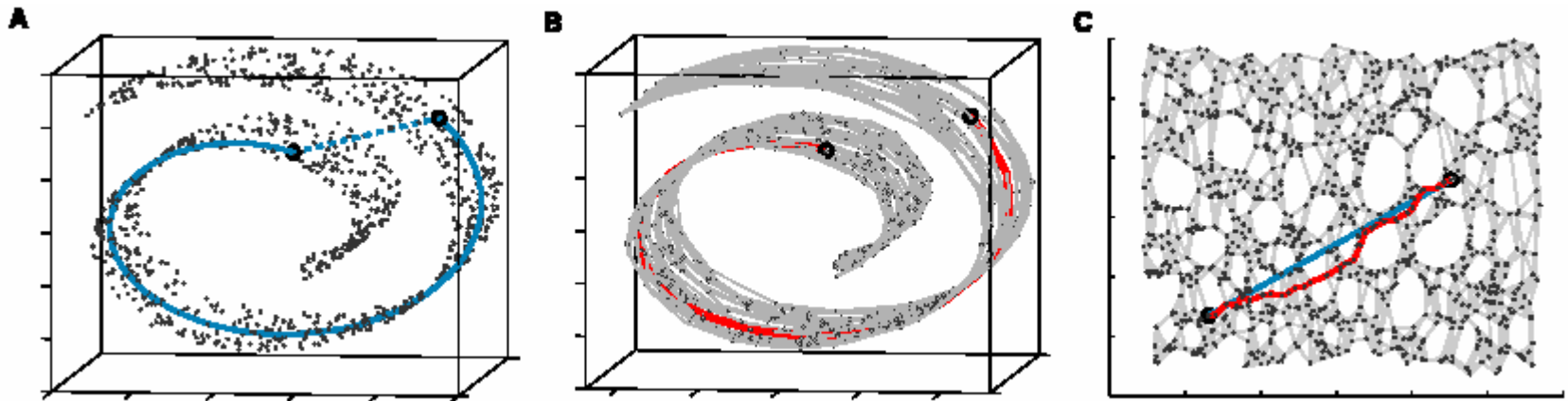
Nonmetric scaling:  $-P^e D P^e$  may not be positive semi-definite

Solutions: (1) Add a large constant to its diagonal.

(2) Find its nearest positive semi-definite matrix by setting all negative eigenvalues to zero.

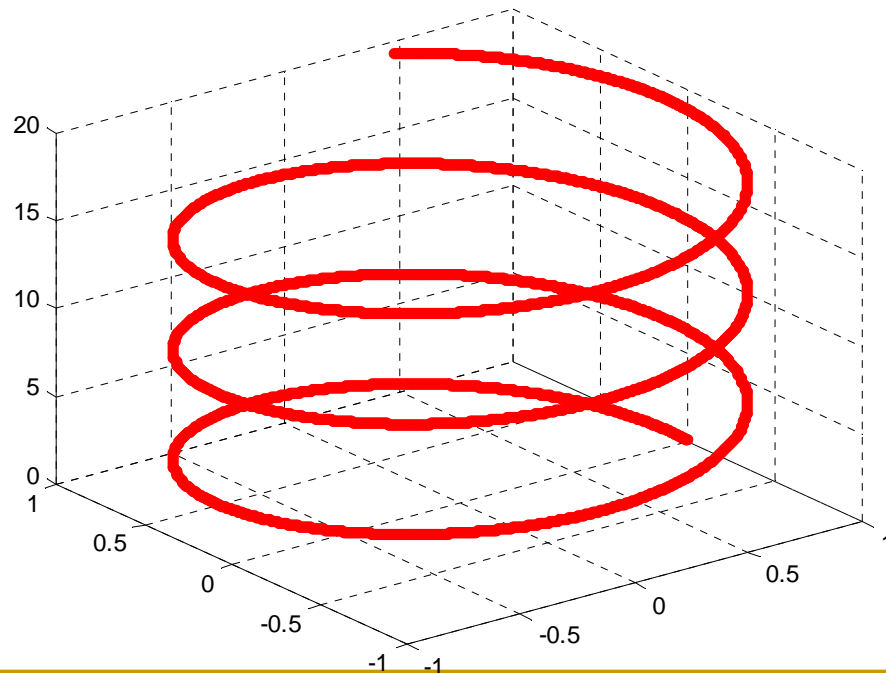
# Manifold Learning

- Discover low dimensional representations (smooth manifold) for data in high dimension.
- A manifold is a topological space which is locally Euclidean
- An example of nonlinear manifold:



# Deficiencies of Linear Methods

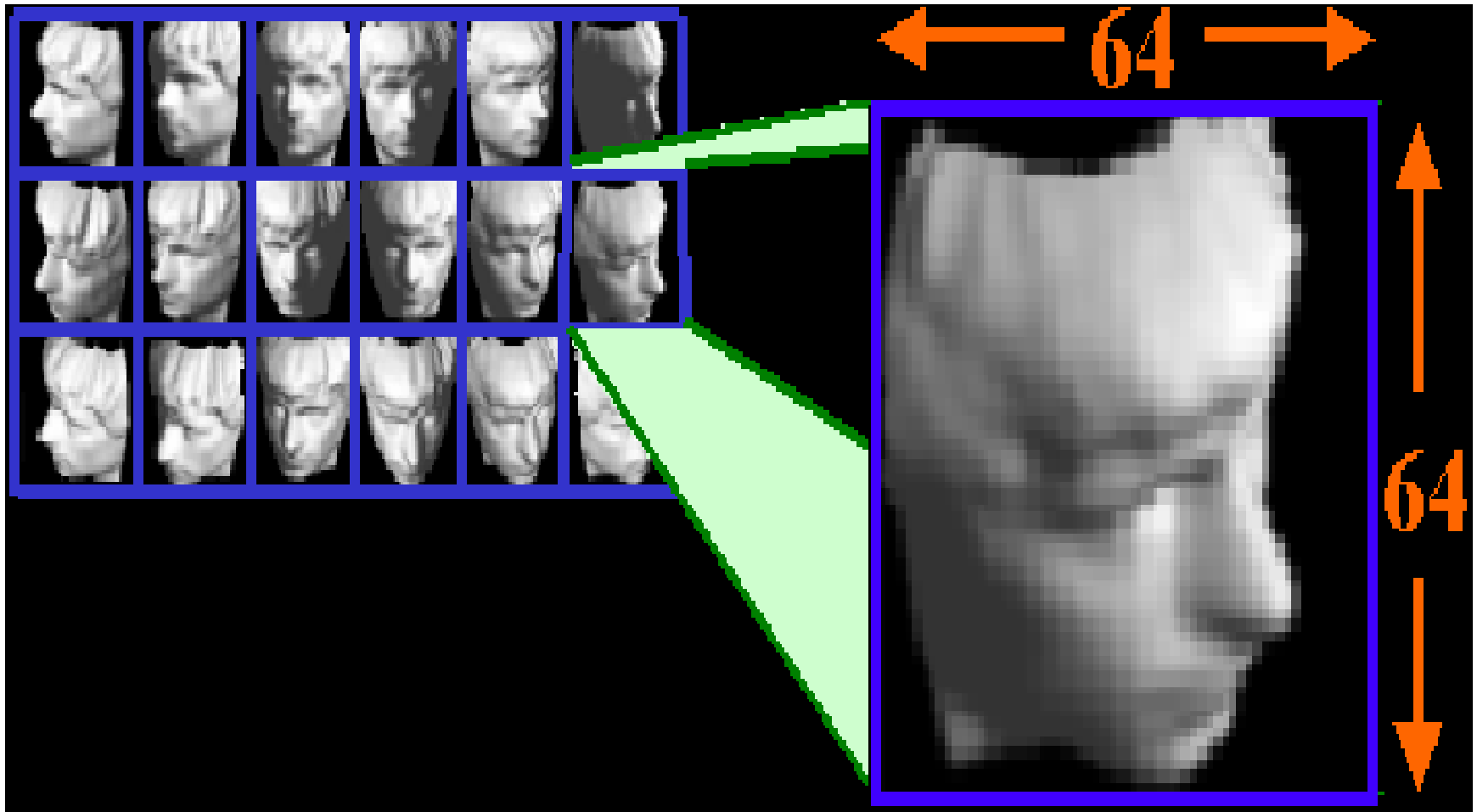
- Data may not be best summarized by linear combination of features
  - Example: PCA cannot discover 1D structure of a helix



# Intuition: how does your brain store these pictures?

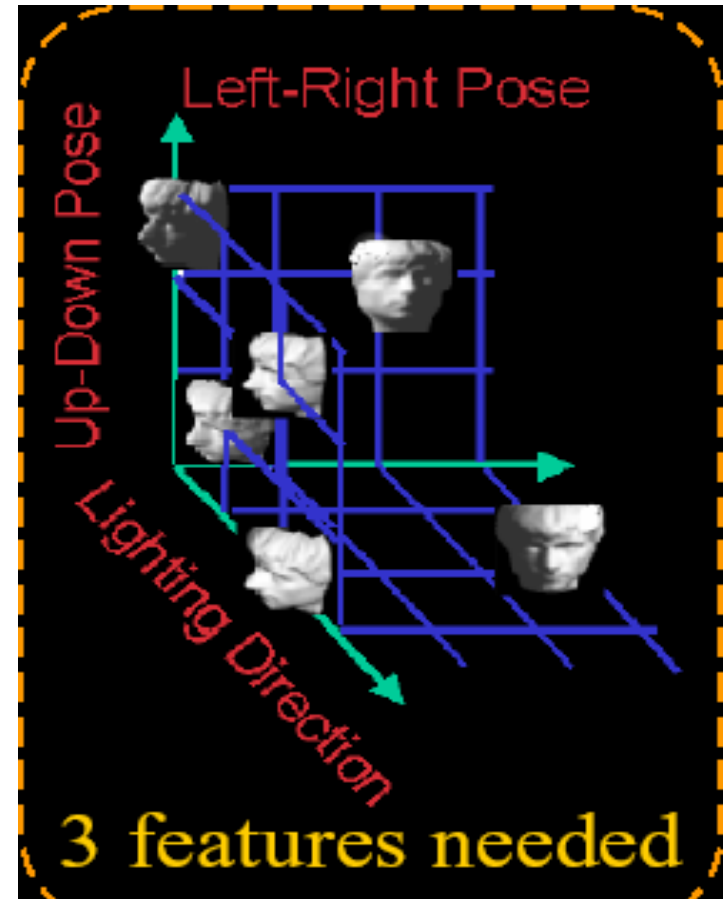


# Brain Representation



# Brain Representation

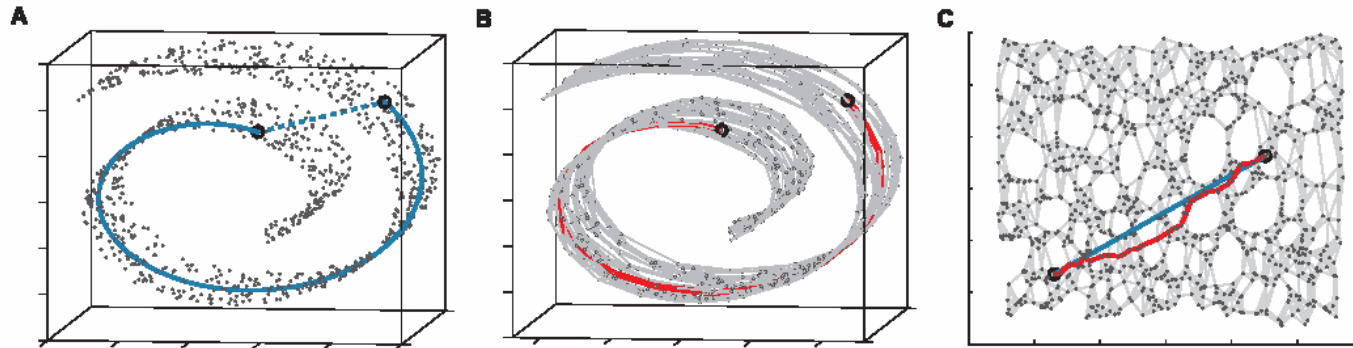
- Every pixel?
  - Or perceptually meaningful structure?
    - Up-down pose
    - Left-right pose
    - Lighting direction
- So, your brain successfully reduced the high-dimensional inputs to an intrinsically 3-dimensional manifold!





# Nonlinear Approaches- Isomap

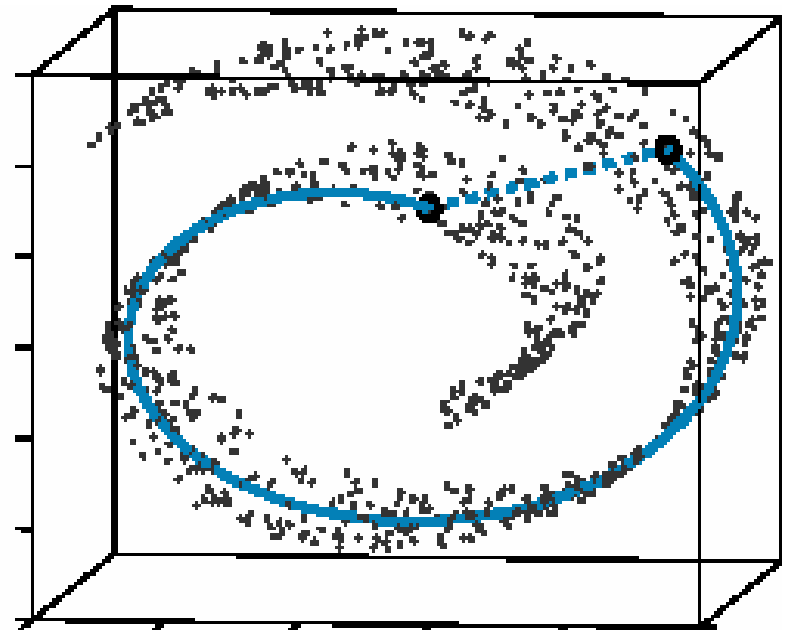
Josh. Tenenbaum, Vin de Silva, John langford 2000



- Constructing neighbourhood graph  $G$
- For each pair of points in  $G$ , Computing shortest path distances ---- **geodesic distances**.
- Use Classical MDS with geodesic distances.  
Euclidean distance  $\rightarrow$  Geodesic distance

# Sample Points with Swiss Roll

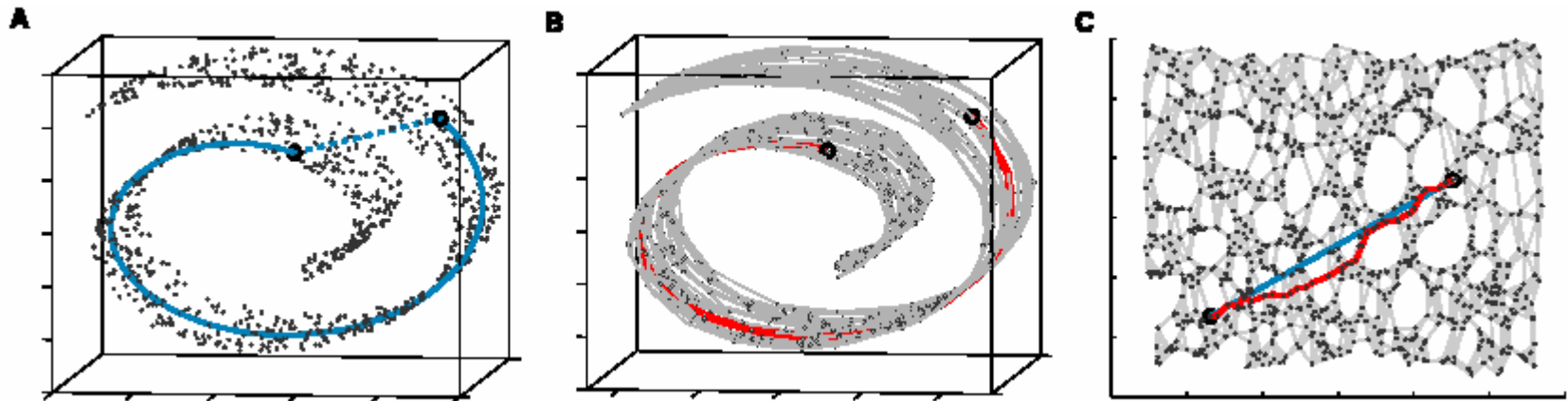
- Altogether there are 20,000 points in the “Swiss roll” data set. We sample 1000 out of 20,000.



# Construct Neighborhood Graph $G$

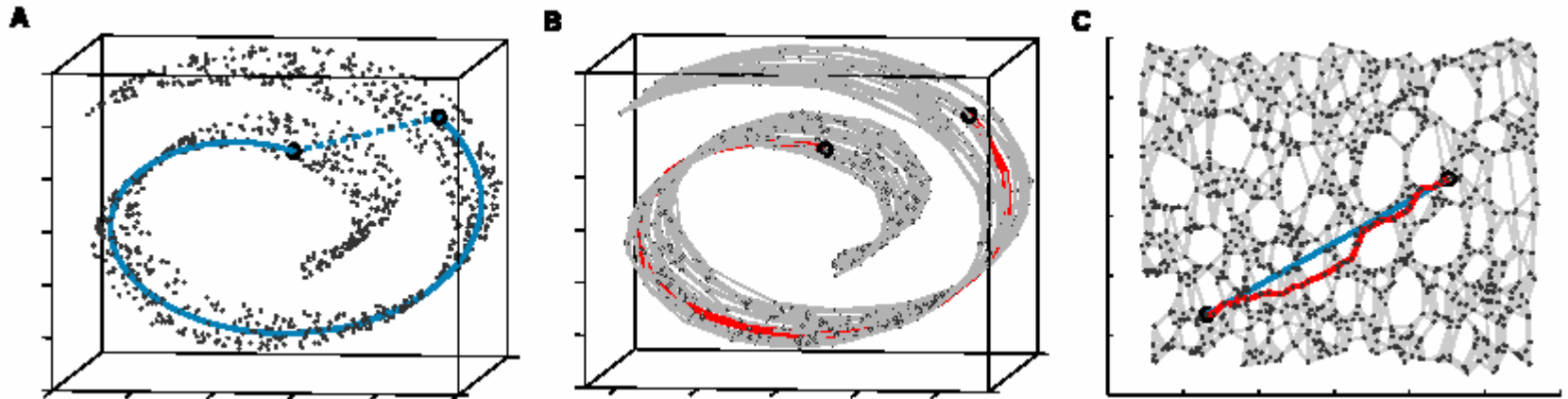
**K- nearest neighborhood** ( $K=7$ )

$D_G$  is 1000 by 1000 (Euclidean) distance matrix of two neighbors (figure A)



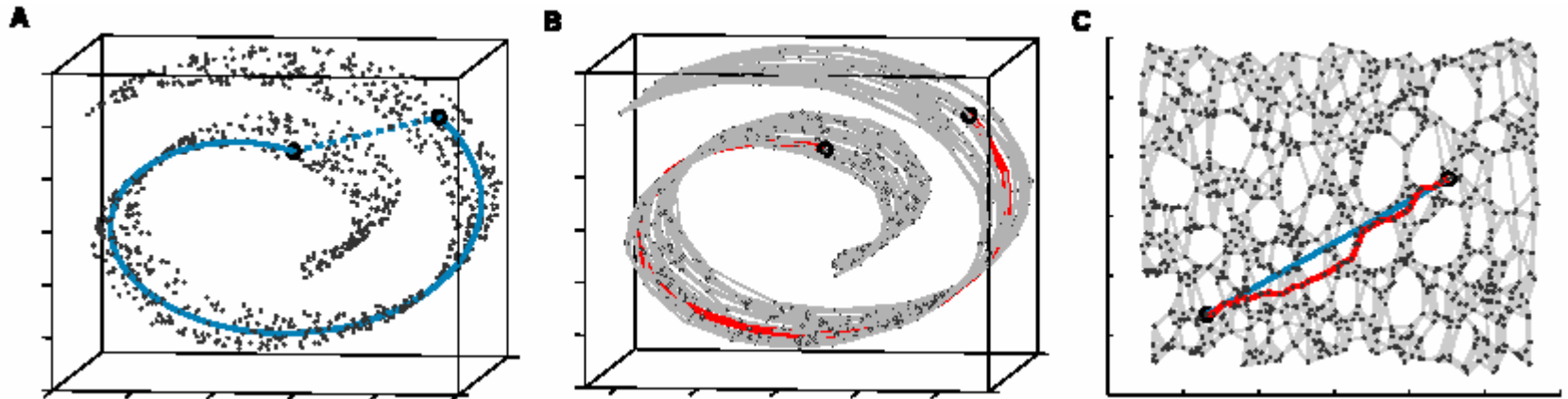
# Compute All-Points Shortest Path in $G$

Now  $D_G$  is 1000 by 1000 **geodesic** distance matrix of two arbitrary points **along the manifold** (figure B)



# Use MDS to Embed Graph in $\mathbb{R}^d$

Find a  $d$ -dimensional Euclidean space  $Y$  (Figure c) to preserve the pairwise distances.



# The Isomap Algorithm

**Table 1.** The Isomap algorithm takes as input the distances  $d_X(i, j)$  between all pairs  $i, j$  from  $N$  data points in the high-dimensional input space  $X$ , measured either in the standard Euclidean metric (as in Fig. 1A) or in some domain-specific metric (as in Fig. 1B). The algorithm outputs coordinate vectors  $y_i$  in a  $d$ -dimensional Euclidean space  $Y$  that (according to Eq. 1) best represent the intrinsic geometry of the data. The only free parameter ( $\epsilon$  or  $K$ ) appears in Step 1.

Step		
1	Construct neighborhood graph	Define the graph $G$ over all data points by connecting points $i$ and $j$ if [as measured by $d_X(i, j)$ ] they are closer than $\epsilon$ ( $\epsilon$ -Isomap), or if $i$ is one of the $K$ nearest neighbors of $j$ ( $K$ -Isomap). Set edge lengths equal to $d_X(i, j)$ .
2	Compute shortest paths	Initialize $d_G(i, j) = d_X(i, j)$ if $i, j$ are linked by an edge; $d_G(i, j) = \infty$ otherwise. Then for each value of $k = 1, 2, \dots, N$ in turn, replace all entries $d_G(i, j)$ by $\min\{d_G(i, j), d_G(i, k) + d_G(k, j)\}$ . The matrix of final values $D_G = \{d_G(i, j)\}$ will contain the shortest path distances between all pairs of points in $G$ (16, 19).
3	Construct $d$ -dimensional embedding	Let $\lambda_\rho$ be the $\rho$ -th eigenvalue (in decreasing order) of the matrix $\tau(D_G)$ (17), and $v_\rho^i$ be the $i$ -th component of the $\rho$ -th eigenvector. Then set the $\rho$ -th component of the $d$ -dimensional coordinate vector $y_i$ equal to $\sqrt{\lambda_\rho} v_\rho^i$ .

# Isomap: Advantages

- Nonlinear
- Globally optimal
  - Still produces globally optimal low-dimensional Euclidean representation even though input space is highly folded, twisted, or curved.
- Guarantee asymptotically to recover the true dimensionality.

# Isomap: Disadvantages

- May not be stable, dependent on topology of data
- Guaranteed asymptotically to recover geometric structure of nonlinear manifolds
  - As  $N$  increases, pairwise distances provide better approximations to geodesics, but cost more computation
  - If  $N$  is small, geodesic distances will be very inaccurate.



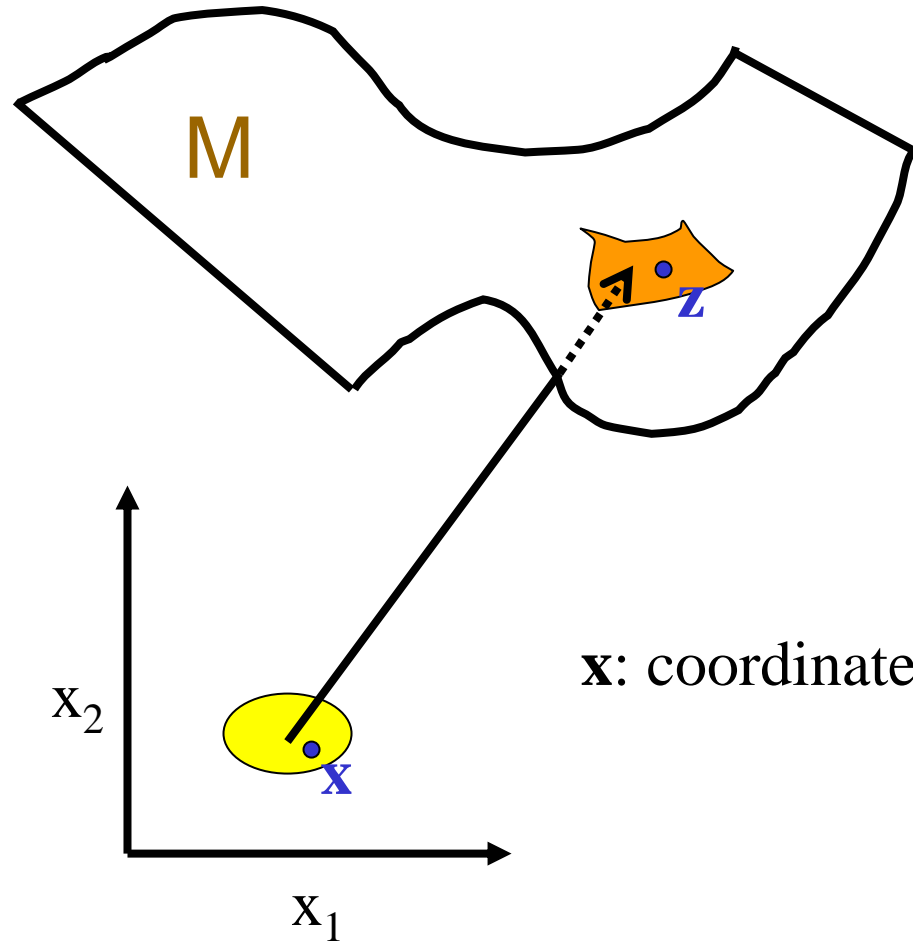
# Characteristics of a Manifold

$\mathbb{R}^n$

Locally it is a linear patch

Key: how to combine all local patches together?

$\mathbb{R}^2$



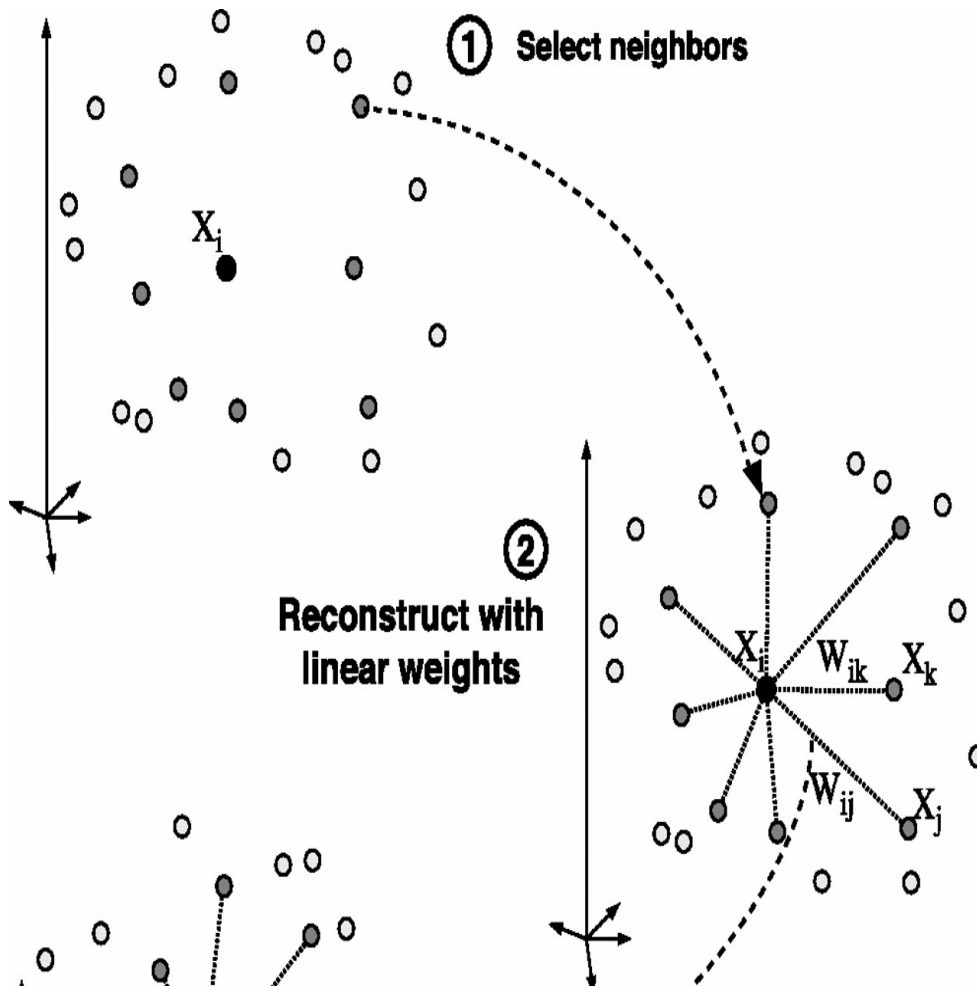
# LLE: Intuition

- Assumption: manifold is approximately “linear” when viewed locally, that is, in a small neighborhood
  - Approximation error,  $e(W)$ , can be made small

$$\min_W \left\| X_i - \sum_{j=1}^K W_{ij} X_j \right\|^2 \quad (1)$$

- Local neighborhood is effected by the constraint  $W_{ij}=0$  if  $z_i$  is not a neighbor of  $z_j$
- A good projection should preserve this local geometric property as much as possible

# LLE: Intuition



We expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold.

Each point can be written as a linear combination of its neighbors.

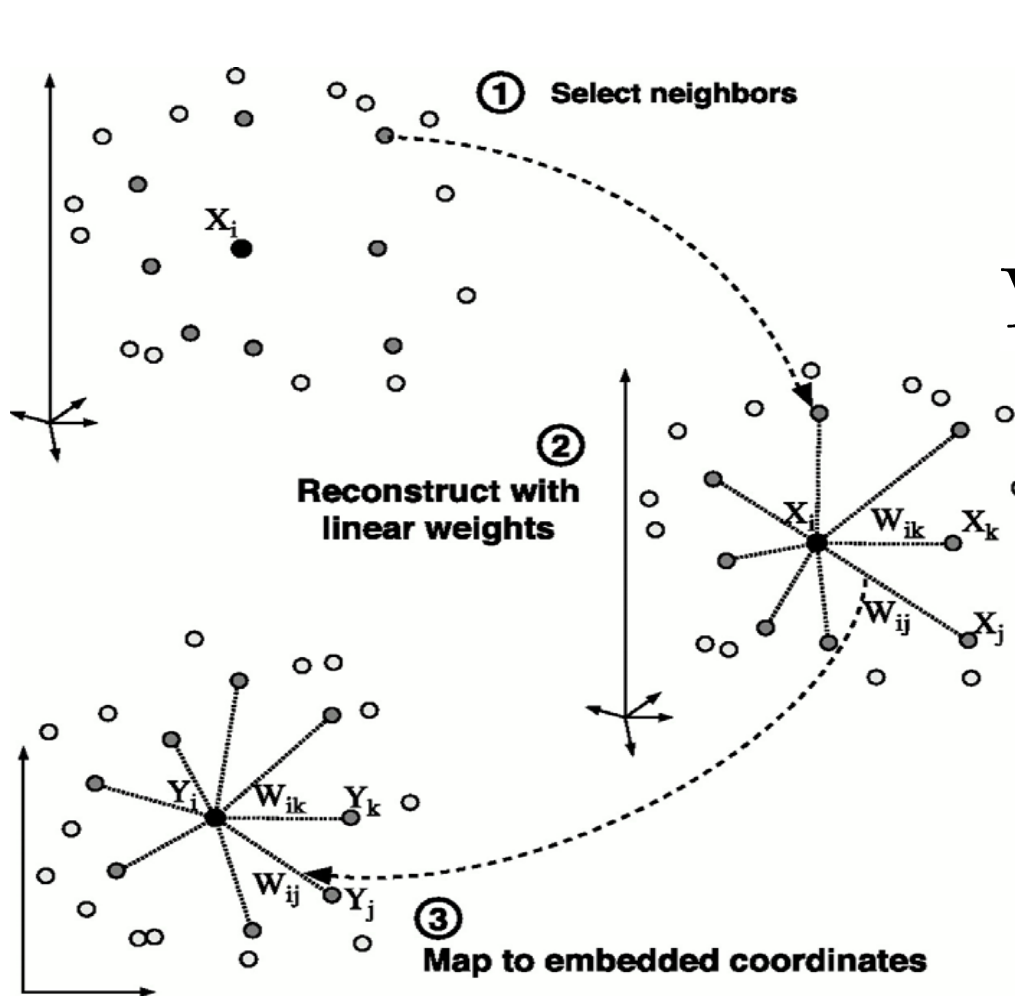
The weights chosen to minimize the reconstruction Error.

$$\min_W \left\| X_i - \sum_{j=1}^K W_{ij} X_j \right\|^2 \quad (1)$$

# LLE: Intuition

- The weights that minimize the reconstruction errors are invariant to rotation, rescaling and translation of the data points.
  - Invariance to translation is enforced by adding the constraint that the weights sum to one.
  - The weights characterize the intrinsic geometric properties of each neighborhood.
- **The same weights that reconstruct the data points in  $D$  dimensions should reconstruct it in the manifold in  $d$  dimensions.**
  - **Local geometry is preserved**

# LLE: Intuition



Low-dimensional embedding

$$Y_{d \times N} = [Y_1 | Y_2 | \dots | Y_N]$$

$$\min_Y \sum_{i=1}^N \| Y_i - Y W_i \|^2$$

the  $i$ -th row of  $W$

Use the same weights  
from the original space

# Local Linear Embedding (LLE)

- Assumption: manifold is approximately “linear” when viewed locally, that is, in a small neighborhood
- Approximation error,  $\varepsilon(\mathbf{W})$ , can be made small

$$\varepsilon(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

- Meaning of  $\mathbf{W}$ : a linear representation of every data point by its neighbors
  - This is an intrinsic geometrical property of the manifold
- A good projection should preserve this geometric property as much as possible

# Constrained Least Square Problem

Compute the optimal weight for each point individually:

$$\varepsilon = \left| \vec{x} - \sum_j w_j \vec{\eta}_j \right|^2 = \left| \sum_j w_j (\vec{x} - \vec{\eta}_j) \right|^2 = \sum_{jk} w_j w_k C_{jk},$$

Neighbors of  $x$   $C_{jk} = (\vec{x} - \vec{\eta}_j) \cdot (\vec{x} - \vec{\eta}_k).$

This error can be minimized in closed form, using a Lagrange multiplier to enforce the constraint that  $\sum_j w_j = 1$ . In terms of the inverse local covariance matrix, the optimal weights are given by:

$$w_j = \frac{\sum_k C_{jk}^{-1}}{\sum_{lm} C_{lm}^{-1}}. \quad (5)$$

Zero for all non-neighbors of  $x$

# Finding a Map to a Lower Dimensional Space

- $\mathbf{Y}_i$  in  $\mathbb{R}^k$ : projected vector for  $\mathbf{X}_i$
- The geometrical property is best preserved if the error below is small

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2$$

Use the same weights  
computed above

- $\mathbf{Y}$  is given by the eigenvectors of the lowest  $d$  non-zero eigenvalues of the matrix

$$\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$$



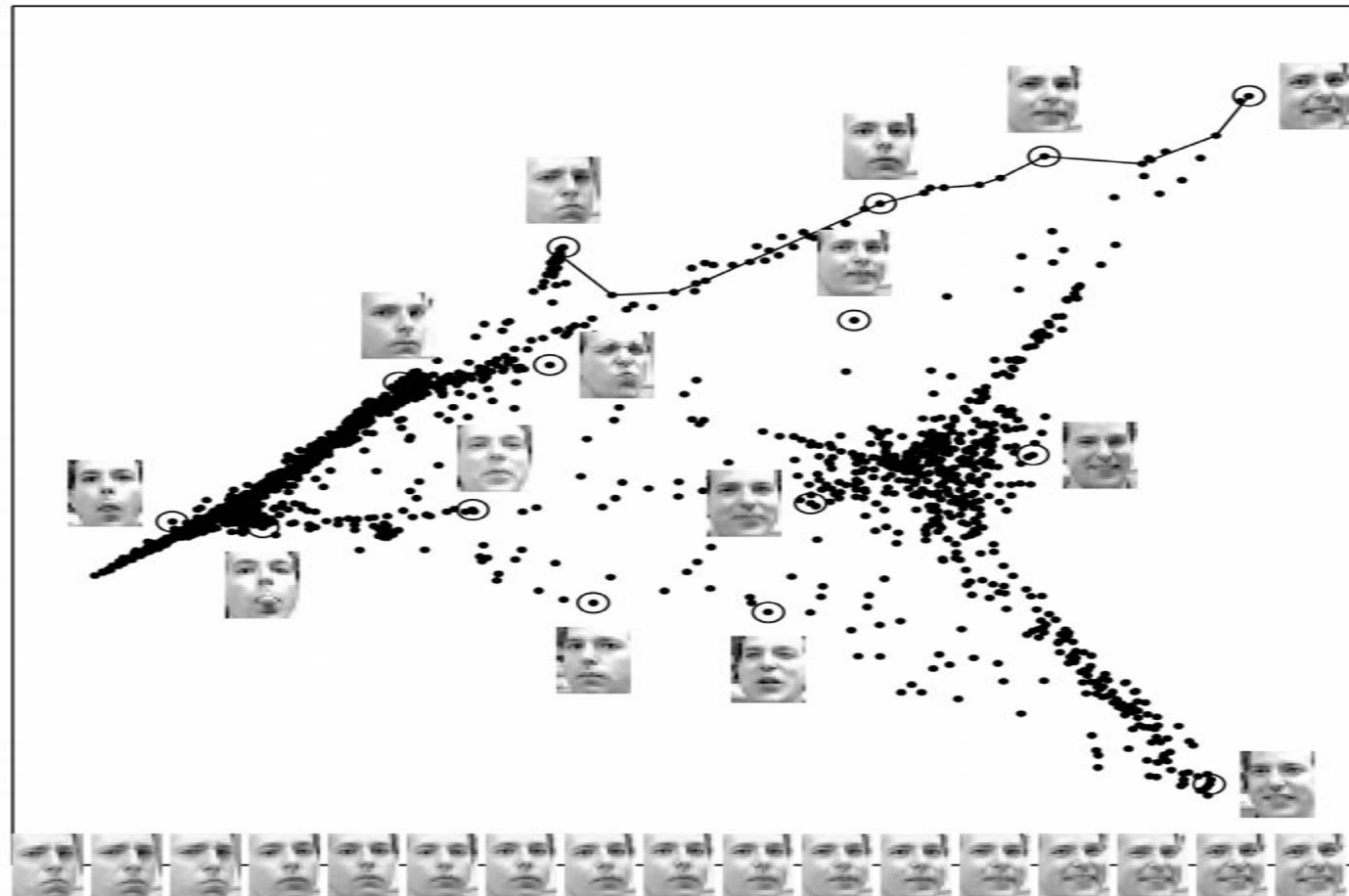
# The LLE Algorithm

## LLE ALGORITHM

1. Compute the neighbors of each data point,  $\vec{X}_i$ .
2. Compute the weights  $W_{ij}$  that best reconstruct each data point  $\vec{X}_i$  from its neighbors, minimizing the cost in eq. (1) by constrained linear fits.
3. Compute the vectors  $\vec{Y}_i$  best reconstructed by the weights  $W_{ij}$ , minimizing the quadratic form in eq. (2) by its bottom nonzero eigenvectors.

Figure 2: Summary of the LLE algorithm, mapping high dimensional data points,  $\vec{X}_i$ , to low dimensional embedding vectors,  $\vec{Y}_i$ .

# Examples



Images of faces mapped into the embedding space described by the first two coordinates of LLE. Representative faces are shown next to circled points. The bottom images correspond to points along the top-right path (linked by solid line) illustrating one particular mode of variability in pose and expression.

# Experiment on LLE

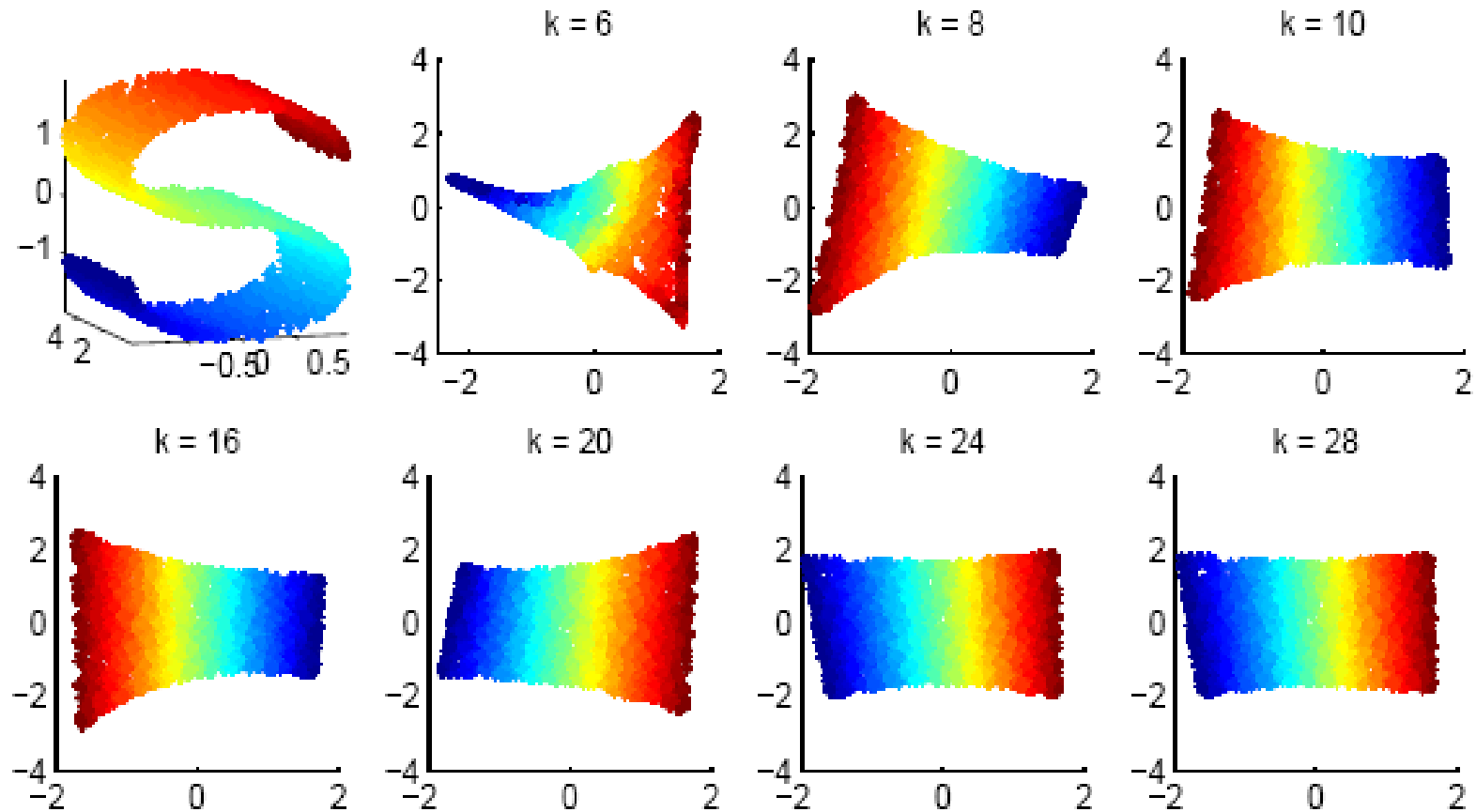


FIG. 5. *S-curve (top left) and computed 2D coordinates by LLE with various neighborhood size  $k$ .*

# Laplacian Eigenmaps

- **Laplacian Eigenmaps for Dimensionality Reduction and Data Representation**
  - M. Belkin, P. Niyogi
- **Key steps**
  - Build the adjacency graph
  - Choose the weights for edges in the graph (**similarity**)
  - Eigen-decomposition of the **graph laplacian**
  - Form the low-dimensional embedding

# Step 1: Adjacency Graph Construction

1. Step 1 [Constructing the Adjacency Graph]. We put an edge between nodes  $i$  and  $j$  if  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are “close”. There are two variations:

- (a)  **$\epsilon$ -neighborhoods.** [parameter  $\epsilon \in \mathbb{R}$ ] Nodes  $i$  and  $j$  are connected by an edge if  $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \epsilon$  where the norm is the usual Euclidean norm in  $\mathbb{R}^l$ .

Advantages: geometrically motivated, the relationship is naturally transitive.

Disadvantages: often leads to graphs with several connected components, difficult to choose  $\epsilon$ .

- (b)  **$n$  nearest neighbors.** [parameter  $n \in \mathbb{N}$ ] Nodes  $i$  and  $j$  are connected by an edge if  $i$  is among  $n$  nearest neighbors of  $j$  or  $j$  is among  $n$  nearest neighbors of  $i$ .

Advantages: easier to choose, does not tend to lead to disconnected graphs.

Disadvantages: less geometrically intuitive.

## Step 2: Choosing the Weight

2. Step 2.<sup>1</sup> [Choosing the weights]. Here, as well, we have two variations for weighting the edges:

(a) Heat kernel. [parameter  $t \in \mathbb{R}$ ]. If nodes  $i$  and  $j$  are connected, put

$$W_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}}$$

The justification for this choice of weights will be provided later.

(b) Simple-minded. [No parameters].  $W_{ij} = 1$  if and only if vertices  $i$  and  $j$  are connected by an edge.

A simplification which avoids the necessity of choosing  $t$ .

# Steps: Eigen-Decomposition

3. Step 3. [Eigenmaps] Assume the graph  $G$ , constructed above, is connected, otherwise proceed with Step 3 for each connected component. Compute eigenvalues and eigenvectors for the generalized eigenvector problem:

$$Lf = \lambda Df \tag{1}$$

where  $D$  is diagonal weight matrix, its entries are column (or row, since  $W$  is symmetric) sums of  $W$ ,  $D_{ii} = \sum_j W_{ji}$ .  $L = D - W$  is the Laplacian matrix. Laplacian is a symmetric, positive semidefinite matrix which can be thought of as an operator on functions defined on vertices of  $G$ .

## Step 4: Embedding

Let  $\mathbf{f}_0, \dots, \mathbf{f}_{k-1}$  be the solutions of equation 1, ordered according to their eigenvalues,

$$L\mathbf{f}_0 = \lambda_0 D\mathbf{f}_0$$

$$L\mathbf{f}_1 = \lambda_1 D\mathbf{f}_1$$

...

$$L\mathbf{f}_{k-1} = \lambda_{k-1} D\mathbf{f}_{k-1}$$

$$0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{k-1}$$

We leave out the eigenvector  $\mathbf{f}_0$  corresponding to eigenvalue 0 and use the next  $m$  eigenvectors for embedding in  $m$ -dimensional Euclidean space.

$$\mathbf{x}_i \rightarrow (\mathbf{f}_1(i), \dots, \mathbf{f}_m(i))$$

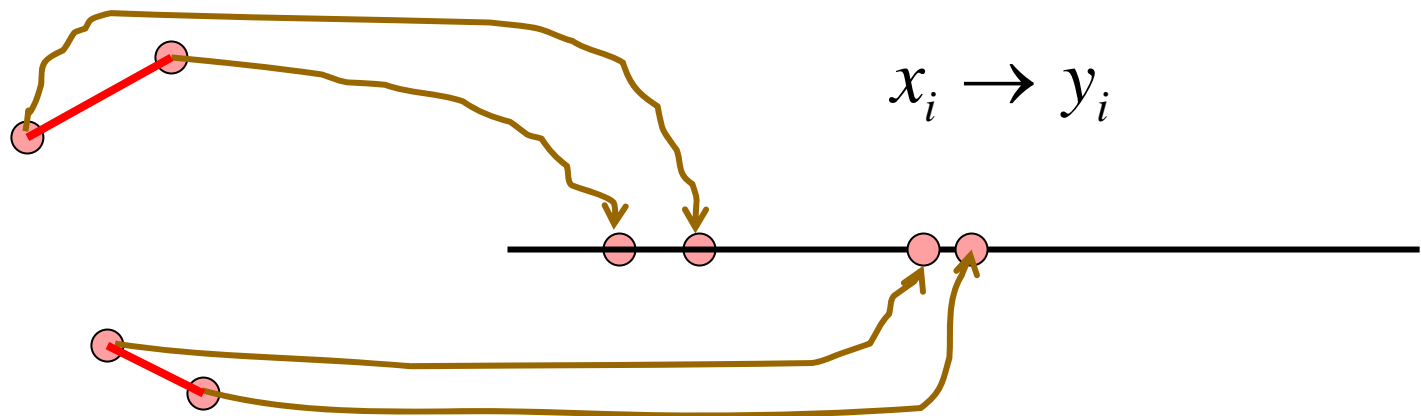


# Justification

given a data set we construct a weighted graph  $G = (V, E)$

Consider the problem of mapping the graph to a line so that pairs of points with large similarity (weight) stay as close as possible.

Let  $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$  be such a map.



A reasonable criterion for choosing the mapping is to minimize

$$\sum_{ij} (y_i - y_j)^2 W_{ij}$$

# Justification

It turns out that for any  $\mathbf{y}$ , we have

$$\frac{1}{2} \sum_{i,j} (y_i - y_j)^2 W_{ij} = \mathbf{y}^T L \mathbf{y} \quad (2)$$

where as before,  $L = D - W$ . To see this, notice that  $W_{ij}$  is symmetric and  $D_{ii} = \sum_j W_{ij}$ . Thus

$$\begin{aligned} \sum_{i,j} (y_i - y_j)^2 W_{ij} &= \sum_{i,j} (y_i^2 + y_j^2 - 2y_i y_j) W_{ij} = \\ \sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2 \sum_{i,j} y_i y_j W_{ij} &= 2\mathbf{y}^T L \mathbf{y} \end{aligned}$$

Note that this calculation also shows that  $L$  is positive semidefinite.

Therefore, the minimization problem reduces to finding

$$\underset{\substack{\mathbf{y} \\ \mathbf{y}^T D \mathbf{y} = 1}}{\operatorname{argmin}} \mathbf{y}^T L \mathbf{y}$$

# An Example

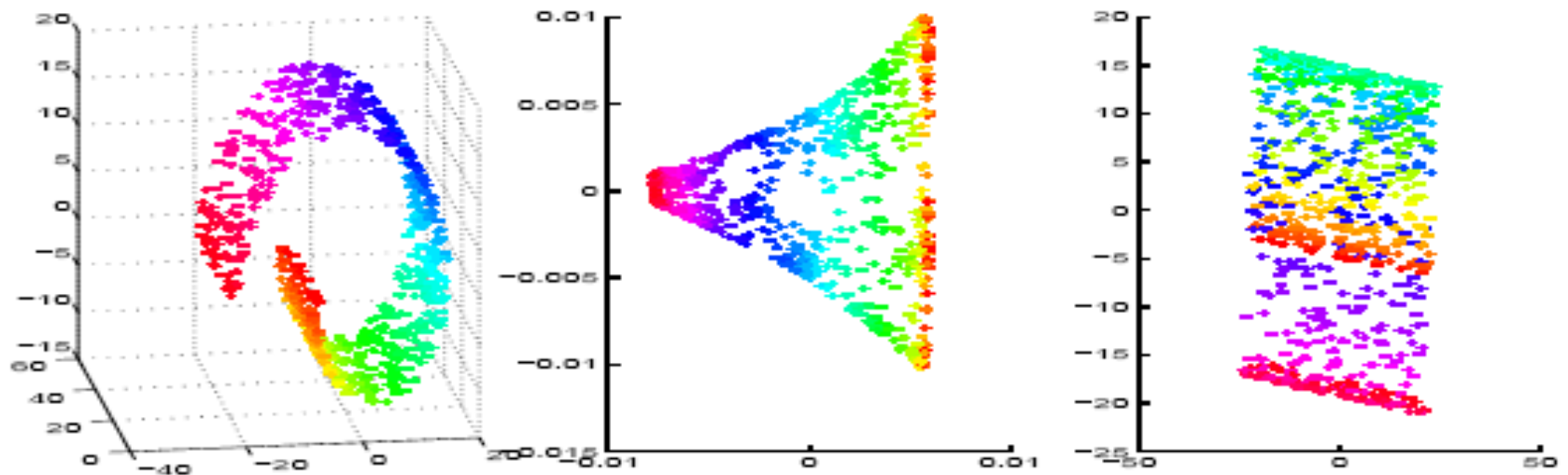


Figure 1: “Swiss roll”, 2-dimensional Laplacian representation, and PCA representation, left to right. For the purposes of illustration we compare a spectral 2-dimensional representation of the “swiss roll” to principal component analysis. PCA is limited to projections and therefore cannot produce a good representation of non-linear data.

# A Unified framework for ML

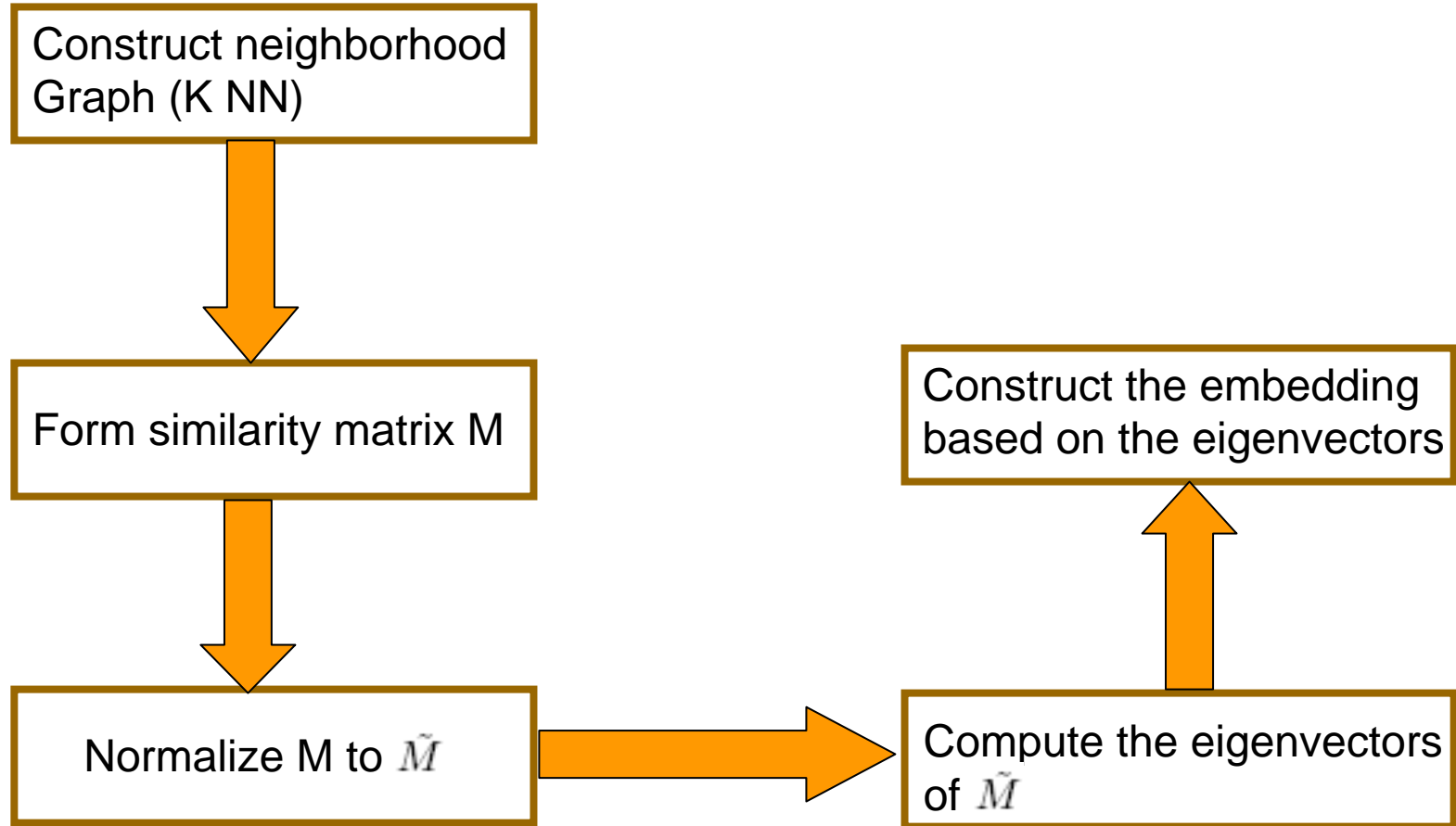
1. Start from a data set  $D = \{x_1, \dots, x_n\}$  with  $n$  points in some space. Construct a  $n \times n$  “neighborhood” or similarity matrix  $M$ . Let us denote  $K_D(\cdot, \cdot)$  (or  $K$  for shorthand) the two-argument function (sometimes dependent on  $D$ ) which produces  $M$  by  $M_{ij} = K_D(x_i, x_j)$ .

2. Optionally transform  $M$ , yielding a “normalized” matrix  $\tilde{M}$ . Equivalently, this corresponds to applying a symmetric two-argument function  $\tilde{K}_D$  to each pair of examples  $(x_i, x_j)$  to obtain  $\tilde{M}_{ij}$ .

3. Compute the  $m$  largest eigenvalues  $\lambda'_j$  and eigenvectors  $v_j$  of  $\tilde{M}$ . Only positive eigenvalues should be considered.

4. The embedding of each example  $x_i$  is the vector  $y_i$  with  $y_{ij}$  the  $i$ -th element of the  $j$ -th principal eigenvector  $v_j$  of  $\tilde{M}$ . Alternatively (MDS and Isomap), the embedding is  $e_i$ , with  $e_{ij} = \sqrt{\lambda'_j} y_{ij}$ . If the first  $m$  eigenvalues are positive, then  $e_i \cdot e_j$  is the best approximation of  $\tilde{M}$  using only  $m$  coordinates, in the squared error sense.

# Flowchart of the Unified Framework



optional

# Outline

- Introduction to dimensionality reduction
- Feature selection (part I)
- Feature extraction (part II)
  - Basics
  - Representative algorithms
  - Recent advances
  - Applications
- Recent trends in dimensionality reduction

---

# Trends in Dimensionality Reduction

- Dimensionality reduction for complex data
  - Biological data
  - Streaming data
- Incorporating prior knowledge
  - Semi-supervised dimensionality reduction
- Combining feature selection with extraction
  - Develop new methods which achieve feature “selection” while efficiently considering feature interaction among all original features

# Feature Interaction

**Definition : ( $k$ th order Feature Interaction)**  $F$  is a feature subset with  $k$  features  $F_1, F_2, \dots, F_k$ . Let  $\mathcal{C}$  denote a metric that measures the relevance of the class label with a feature or a feature subset. Features  $F_1, F_2, \dots, F_k$  are said to interact with each other iff: for an arbitrary partition  $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_l\}$  of  $F$ , where  $l \geq 2$  and  $\mathcal{F}_i \neq \phi$ , we have

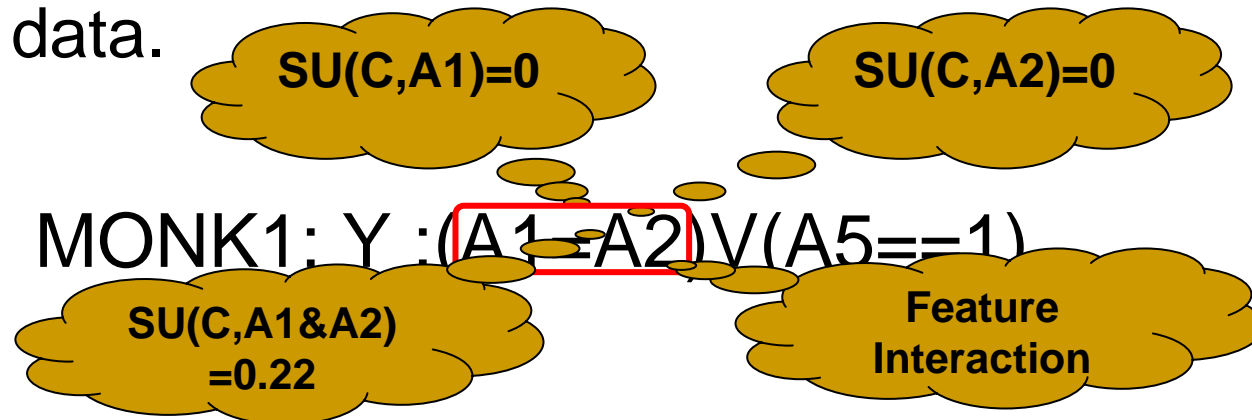
$$\forall i \in [1, l], \quad \mathcal{C}(F) > \mathcal{C}(\mathcal{F}_i)$$

- A set of features are interacting with each, if they become more relevant when considered together than considered individually.
- A feature could lose its relevance due to the absence of any other feature interacting with it, or irreducibility [Jakulin05].



# Feature Interaction

- Two examples of feature interaction: MONK1 & Corral data.



Corral: Y : (A0^A1) V (B0^B1)

- Existing efficient feature selection algorithms can not handle feature interaction very well

	FCBF	CFS	ReliefF	FOCUS
Corral	$A_0, A_1, B_0, B_1, \mathbf{R}$	$A_0, -, -, -, \mathbf{R}$	$A_0, A_1, B_0, B_1, \mathbf{R}$	$A_0, A_1, B_0, B_1$
Monk1	$-, -, A_5$	$-, -, A_5$	$A_1, A_2, A_5$	$A_1, A_2, A_5$

# Illustration using synthetic data

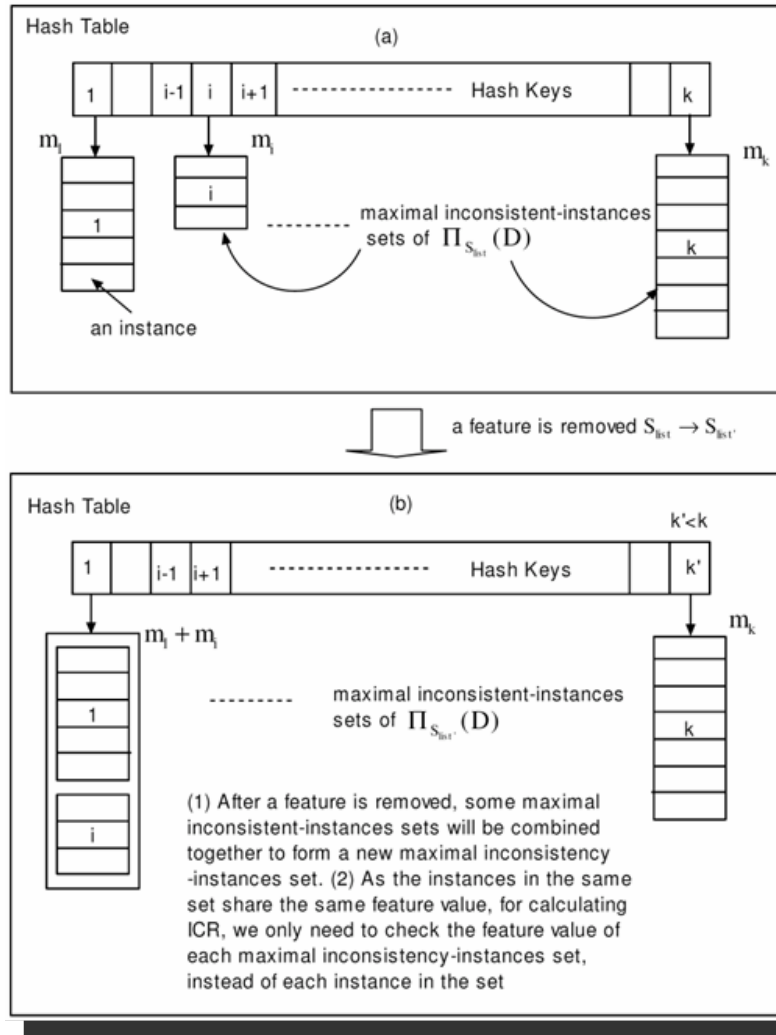
- MONKs data, for class  $C = 1$ 
  - (1) MONK1:  $(A1 = A2)$  or  $(A5 = 1)$ ;
  - (2) MONK2: Exactly two  $A_i = 1$ ; (all features are relevant)
  - (3) MONK3:  $(A5 = 3 \text{ and } A4 = 1)$  or  $(A5 \neq 4 \text{ and } A2 \neq 3)$
- Experiment with FCBF, ReliefF, CFS, FOCUS

	Relevant Features	FCBF	CFS	ReliefF	FOCUS
MONKS Data (Full Data)					
Monk1	$A_1, A_2, A_5$	$- , - , A_5$	$- , - , A_5$	$A_1, A_2, A_5$	$A_1, A_2, A_5$
Monk2	$A_1, A_2, A_3, A_4,$ $A_5, A_6$	$A_1, A_2, A_3, A_4,$ $A_5, A_6$	$- , - , - , - ,$ $A_5, -$	$- , A_2, A_3, A_4,$ $- , A_6$	$A_1, A_2, A_3, A_4,$ $A_5, A_6$
Monk3	$A_2, A_4, A_5$	$A_2, A_4, A_5$	$A_2, - , -$	$A_2, - , A_5$	$A_2, A_4, A_5$

# Existing Solutions for Feature Interaction

- Existing efficient feature selection algorithms usually assume feature independence.
- Others attempt to explicitly address Feature Interactions by finding them.
  - Find out all Feature Interaction is impractical.
- Some existing efficient algorithm can only (partially) address low order Feature Interaction, 2 or 3-way Feature Interaction.

# Handle Feature Interactions (INTERACT)



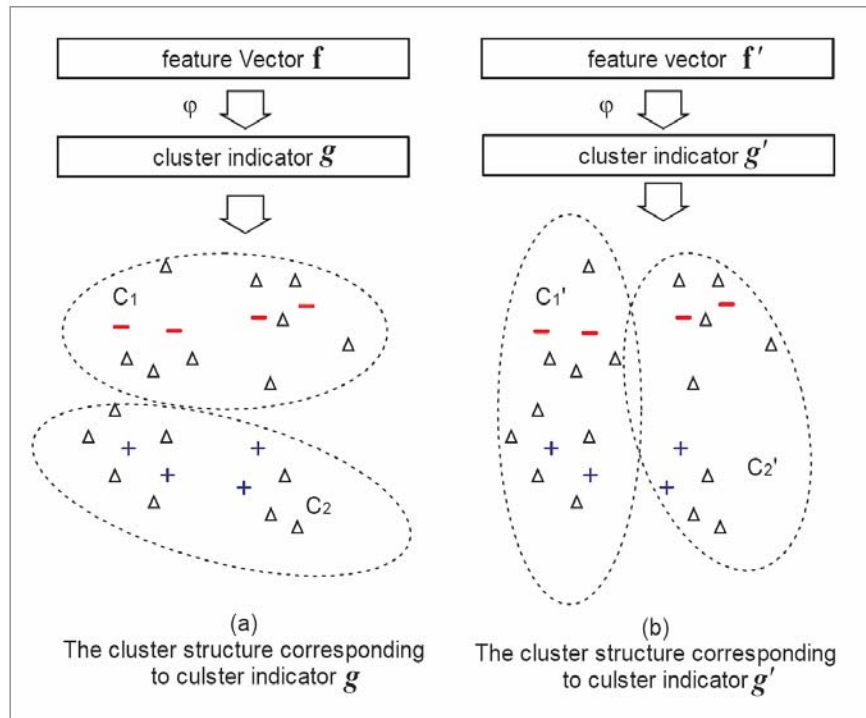
- Designing a feature scoring metric based on the consistency hypothesis: c-contribution.
- Designing a data structure to facilitate the fast update of c-contribution
- Selecting a simple and fast search schema
- INTERACT is a backward elimination algorithm [Zhao-Liu07]

# Semi-supervised Feature Selection

DEFINITION : (SEMI-SUPERVISED FEATURE SELECTION) *Given data  $X_L$  and  $X_U \subseteq R^m$ , semi-supervised feature selection is to use both  $X_L$  and  $X_U$  to identify the set of most relevant features  $\{F_{j_1}, F_{j_2}, \dots, F_{j_k}\}$  of the target concept, where  $k \leq m$  and  $j_r \in \{1, 2, \dots, m\}$  for  $r \in \{1, 2, \dots, k\}$ .*

- For handling small labeled-sample problem
  - *Labeled data is few, but unlabeled data is abundant*
  - *Neither supervised nor unsupervised works well*
- Using both labeled and unlabeled data

# Measure Feature Relevance



Transformation Function:

$$\mathbf{g}_f = \mathbf{f} - \frac{\sum_i^n f_i d_i}{\text{vol}V} \cdot \mathbf{e};$$

Relevance Measurement:

$$\lambda \frac{\sum_{v_i \sim v_j} (g_i - g_j)^2 \times w_{ij}}{2 \sum_{v_i \in V} g_i^2 \times d_i} + (1 - \lambda)(1 - NMI(\hat{\mathbf{g}}, \mathbf{y}))$$

- Construct cluster indicator from features.
- Measure the fitness of the cluster indicator using both labeled and unlabeled data.
- sSelect algorithm uses spectral analysis [Zhao-Liu07S].

---

# References

- Almuallim, H., & Dietterich, T. G. (1994). Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69, 279–305.
- Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Advances in Neural Information Processing Systems*, 15.
- Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97, 245–271.
- Cox, T. (2000). *Multidimensional scaling*. Chapman & Hall/CRC; 2 edition.
- Dash, M., Choi, K., Scheuermann, P., & Liu, H. (2002). Feature selection for clustering – a filter solution. *Proceedings of the Second International Conference on Data Mining* (pp. 115–122).
- Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis: An International Journal*, 1, 131–156.

---

# References

- Dash, M., & Liu, H. (2003). Consistency-based search in feature selection. *Artificial Intelligence*, 151, 155–176.
- Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3, 185–205.
- Ding, C., & Ye, J. (2005). 2-Dimensional singular value decomposition for 2D maps and images. *Proceedings of the Fifth SIAM International Conference on Data Mining (SDM)*.
- Dy, J. G., & Brodley, C. E. (2000). Feature subset selection and order identification for unsupervised learning. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 247–254).



---

# References

- Dy, J. G., Brodley, C. E., Kak, A. C., Broderick, L. S., & Aisen, A. M. (2003). Un-supervised feature selection applied to content-based retrieval of lung images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 373–378.
- Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, 179–188.
- Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3, 1289–1305.
- Gilad-Bachrach, R., Navot, A., & Tishby, N. (2004). Margin based feature selection - theory and algorithms. *Proceedings of the 21st International Conference on Machine Learning*.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46, 389–422.
- Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 359–366).

---

# References

- Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 312–377.
- J. B. Tenenbaum, V. d. S., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290, 2319–2323.
- Jakulin, A., & Bratko, I. (2004). Testing the significance of attribute interactions. *Proceedings of the 21st International Conference on Machine Learning*.
- Jolliffe, I. T. (1986). *Principal component analysis*. New York: Springer-Verlag.
- Ketterling, J. (1971). Canonical analysis of several sets of variables. *Biometrika*, 433–451.
- Kim, Y., Street, W., & Menczer, F. (2000). Feature selection for unsupervised learning via evolutionary search. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 365–369).
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97, 273–324.

---

# References

- Lee, W., Stolfo, S. J., & Mok, K. W. (2000). Adaptive intrusion detection: A data mining approach. *AI Review*, 14, 533 – 567.
- Liu, H., & Motoda, H. (Eds.). (1998a). *Feature extraction, construction and selection: A data mining perspective*. Boston: Kluwer Academic Publishers. 2nd Printing, 2001.
- Liu, H., & Motoda, H. (1998b). *Feature selection for knowledge discovery and data mining*. Boston: Kluwer Academic Publishers.
- Liu, H., & Setiono, R. (1996). A probabilistic approach to feature selection - a filter solution. *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 319–327).
- Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 1226–1238.
- Robnik-Sikonja, M., & Kononenko, I. (2001). Comprehensible interpretation of relief's estimates. *Proceedings of Eighteenth International Conference on Machine Learning* (pp. 433–440).

---

# References

- Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290, 23232326.
- Sun, Y., & Li, J. (2006). Iterative relief for feature weighting. *Proceedings of the 23rd International Conference on Machine Learning* (pp. 913–920).
- Swets, D. L., & Weng, J. J. (1995). Efficient content-based image retrieval using automatic feature selection. *IEEE International Symposium on Computer Vision* (pp. 85–90).
- Talavera, L. (1999). Feature selection as a preprocessing step for hierarchical clustering. *Proceedings of International Conference on Machine Learning (ICML'99)* (pp. 389–397).
- Williams, C., & Seeger, M. (2001). Using the nystrom method to speedup kernel machines. *Advances in Neural Information Processing Systems*.
- Wold, H. (1985). Partial least squares. *Encyclopedia of the Statistical Sciences*, 6, 581–5911.

---

# References

- Yang, Y., & Pederson, J. O. (1997). A comparative study on feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 412–420).
- Ye, J. (2005). Generalized low rank approximations of matrices. *Machine Learning*, 61, 167–191.
- Ye, J. (2006). *Least squares linear discriminant analysis* (Technical Report TR-06-003, Department of Computer Science and Engineering, Arizona State University).
- Ye, J., & Xiong, T. (2006a). Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. *Journal of Machine Learning Research* (p. 11831204).
- Ye, J., & Xiong, T. (2006b). *SVM versus least squares SVM* (Technical Report TR-06-012, Department of Computer Science and Engineering, Arizona State University).
-

# References

- Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: a fast correlation-based filter solution. *Proceedings of the twentieth International Conference on Machine Learning* (pp. 856–863).
- Yu, L., & Liu, H. (2004). Redundancy based feature selection for microarray data. *Proceedings of the Tenth ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 737–742).
- Yu, S., Yu, K., Tresp, V., Kriegel, H.-P., & Wu, M. (2006). Supervised probabilistic principal component analysis. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge Discovery and Data Mining* (pp. 464–473).

# Reference

- Z. Zhao, H. Liu, Searching for Interacting Features, IJCAI 2007
- A. Jakulin, Machine learning based on attribute interactions, Ph.D. thesis, University of Ljubljana 2005.
- Z. Zhao, H. Liu, Semi-supervised Feature Selection via Spectral Analysis, SDM 2007