



Machine Learning

Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University

Course Outline

- Course details:
 - MCA 5th Semester
 - Course Code & Name: CS304 - Machine Learning No of credits: 6
 - MCA/M.Sc. 3rd Semester
 - Course Code & Name: CS209 - Machine Learning No of credits: 5
- Instructor details:
 - Name: Dr. S. Suresh, Assistant Professor
 - Contact number: 9941506562
 - Email: suresh.selvam@bhu.ac.in
- Text book:
 - David Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press
 - Tom M. Mitchell, Machine Learning, Mc Graw Hill

Old Syllabus

CS304	Machine Learning	L 4	T 0	P 2	C 6
-------	------------------	--------	--------	--------	--------



Machine Learning Concepts: Designing a Learning System; Styles of Learning; Supervised learning; Unsupervised Learning; Semi-Supervised Learning; Basics of Decision Theory, Information Theory and Probability Distributions; Linear and Logistic Regression.

Bayesian Learning: Notion of Prior, Likelihood and Posterior; Naïve Bayes and Conditional Independence; Estimation using Maximum Likelihood; Hidden variables and Missing Data; Bayesian Models.

Applications: Naive Bayes, Nearest Neighbour and Linear Classification Models; K-means and Expectation Maximization for Clustering; Mixture Models.

Machine Learning Applications and Laboratory Exercises.

Suggested Readings:

1. David Barber, Bayesian Reasoning and Machine Learning, CUP.
2. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer.
3. Tom M. Mitchell, Machine Learning, Mc Graw Hill.
4. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press..
5. Daphne Koller and Nir Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press..
6. Peter Harrington, Machine Learning in Action, Manning Publications..

New Syllabus

CS209	Machine Learning	L	T	P
		3	0	2

Machine Learning Concepts: Designing a Learning System, Styles of Learning; Supervised learning; Unsupervised Learning; Semi-Supervised Learning; Basics of Decision Theory, Information Theory and Probability Distributions; Linear and Logistic Regression.

Bayesian Learning: Notion of Prior, Likelihood and Posterior; Naïve Bayes and Conditional Independence; Estimation using Maximum Likelihood; Hidden variables and Missing Data; Bayesian Models.

Classification & Clustering: Naive Bayes, Nearest Neighbour and Linear Classification Models; K-means and Expectation Maximization for Clustering; Mixture Models, Flat and Hierarchical Clustering, Applications of Classification and Clustering.

Suggested Readings:

1. T. M. Mitchell, Machine Learning, McGraw Hill.
2. C. M. Bishop, Pattern Recognition and Machine Learning, Springer.
3. K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press.
4. D. Barber, B. Reasoning, Machine Learning, CUP.
5. P. Harrington, Machine Learning in Action, Manning Publications.

Introduction

- What is Machine Learning?
- Motivation: Why Machine Learning?
- Applications of Machine Learning
- Designing a Learning System
- Issues in Machine Learning

What is Learning?

- Dictionary defines “to learn” as
 - To get knowledge of something by study, experience, or being taught.
 - To become aware by information or from observation
 - To commit to memory
 - To be informed of or to ascertain
 - To receive instruction
- Things learn when they change their behavior in a way that makes them perform better in future.

What is Learning?

- “Learning denotes changes in a system that enable a system to do the same task more efficiently the next time.” -- *Herbert Simon*
- “Learning is constructing or modifying representations of what is being experienced.”
-- *Ryszard Michalski*
- “Learning is making useful changes in our minds.”
-- *Marvin Minsky*

Machine Learning: A Definition

- **Definition:**

A computer program (machine) is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Examples of Successful Applications of Machine Learning

- Learning to recognize spoken words (Lee, 1989; Waibel, 1989).
- Learning to drive an autonomous vehicle (Pomerleau, 1989).
- Learning to classify new astronomical structures (Fayyad et al., 1995).
- Learning to play world-class backgammon (Tesauro 1992, 1995).

Why is Machine Learning Important?

- Some tasks cannot be defined well, except by examples (e.g., recognizing people).
- Relationships and correlations can be hidden within large amounts of data. Machine Learning/Data Mining may be able to find these relationships.
- Human designers often produce machines that do not work as well as desired in the environments in which they are used.

Why is Machine Learning Important (Cont'd)?

- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans (e.g., medical diagnostic).
- Environments change over time.
- New knowledge about tasks is constantly being discovered by humans. It may be difficult to continuously re-design systems “by hand”.

Areas of Influence for Machine Learning

- **Statistics:** How best to use samples drawn from unknown probability distributions to help decide from which distribution some new sample is drawn?
- **Brain Models:** Non-linear elements with weighted inputs (Artificial Neural Networks) have been suggested as simple models of biological neurons.
- **Adaptive Control Theory:** How to deal with controlling a process having unknown parameters that must be estimated during operation?

Areas of Influence for Machine Learning (Cont'd)

- **Psychology:** How to model human performance on various learning tasks?
- **Artificial Intelligence:** How to write algorithms to acquire the knowledge humans are able to acquire, at least, as well as humans?
- **Evolutionary Models:** How to model certain aspects of biological evolution to improve the performance of computer programs?

Effects of Programs that Learn

- Application areas
 - Learning from medical records which treatments are most effective for new diseases
 - Houses learning from experience to optimize energy costs based on usage patterns of their occupants
 - Personal software assistants learning the evolving interests of users in order to highlight especially relevant stories from the online morning newspaper

Effective Applications of Learning

- Speech recognition
 - outperform all other approaches that have been attempted to date
- Data mining
 - Learning algorithms being used to discover valuable knowledge from large commercial databases
 - detect fraudulent use of credit cards
- Play Games
 - Play backgammon at levels approaching the performance of human world champions

Learning Programs

- A computer program is said to **learn** from experiences E with respect to some class of tasks T and performance P , if its performance at tasks in T , as measured by P , improves with experience E
- Examples
 - **A checkers learning problem:**
 - Task T : playing checkers
 - Performance measure P : percent of games won against opponents
 - Training experience E : playing practice games against itself
 - **Handwriting recognition learning problem:**
 - Task T : recognizing and classifying handwritten words within images
 - Performance measure P : percent of words correctly classified
 - Training experience E : a database of handwritten words with given classifications

Designing a Learning System: An Example

1. Problem Description
2. Choosing the Training Experience
3. Choosing the Target Function
4. Choosing a Representation for the Target Function
5. Choosing a Function Approximation Algorithm
6. Final Design

Three Aspects of Learning Systems

- **1. Models:**
 - decision trees,
 - linear threshold units, neural networks,
 - Bayesian networks (polytrees, belief networks, influence diagrams, HMMs),
 - genetic algorithms, instance-based (nearest-neighbor)
- **2. Algorithms (e.g., for decision trees):**
 - ID3, C4.5,
 - CART, OC1
- **3. Methodologies:**
 - supervised,
 - unsupervised,
 - reinforcement;
 - knowledge-guided

Designing a Learning System

- Consider designing a program to learn to play checkers, with the goal of entering it in the world checkers tournament
- Requires the following sets
 - Choosing Training Experience
 - Choosing the Target Function
 - Choosing the Representation of the Target Function
 - Choosing the Function Approximation Algorithm

Choosing the Training Experience (1)

- Will the training experience provide *direct* or *indirect* feedback?
 - Direct Feedback: system learns from examples of individual checkers board states and the correct move for each
 - Indirect Feedback: Move sequences and final outcomes of various games played
 - *Credit assignment problem*: Value of early states must be inferred from the outcome
- Degree to which the learner controls the sequence of training examples
 - Teacher selects informative boards and gives correct move
 - Learner proposes board states that it finds particularly confusing. Teacher provides correct moves
 - Learner controls board states and (indirect) training classifications

Choosing the Training Experience (2)

- How well the training experience represents the distribution of examples over which the final system performance P will be measured
 - If training the checkers program consists only of experiences played against itself, it may never encounter crucial board states that are likely to be played by the human checkers champion
 - Most theory of machine learning rests on the assumption that the distribution of training examples is identical to the distribution of test examples

Partial Design of Checkers Learning Program

- A checkers learning problem:
 - Task T : playing checkers
 - Performance measure P : percent of games won in the world tournament
 - Training experience E : games played against itself
- Remaining choices
 - The exact type of knowledge to be learned
 - A representation for this target knowledge
 - A learning mechanism

Choosing the Target Function (1)

- Assume that you can determine legal moves
- Program needs to learn the best move from among legal moves
 - Defines large search space known a priori
 - *target function*: $\text{ChooseMove} : \mathcal{B} \rightarrow \mathcal{M}$
- ChooseMove is difficult to learn given indirect training
- Alternative target function
 - An evaluation function that assigns a numerical score to any given board state
 - $V: \mathcal{B} \rightarrow \mathbb{R}$ (where \mathbb{R} is the set of real numbers)
 - $V(b)$ for an arbitrary board state b in \mathcal{B}
 - if b is a final board state that is won, then $V(b) = 100$
 - if b is a final board state that is lost, then $V(b) = -100$
 - if b is a final board state that is drawn, then $V(b) = 0$
 - if b is not a final state, then $V(b) = V(b')$, where b' is the best final board state that can be achieved starting from b and playing optimally until the end of the game

Choosing the Target Function (2)

- $V(b)$ gives a recursive definition for board state b
 - Not usable because not efficient to compute except is first three trivial cases
 - *nonoperational* definition
- Goal of learning is to discover an operational description of V
- Learning the target function is often called function approximation
 - Referred to as \hat{V}

Choosing a Representation for the Target Function

- Choice of representations involve trade offs
 - Pick a very expressive representation to allow close approximation to the ideal target function V
 - More expressive, more training data required to choose among alternative hypotheses
- Use linear combination of the following board features:
 - x_1 : the number of black pieces on the board
 - x_2 : the number of red pieces on the board
 - x_3 : the number of black kings on the board
 - x_4 : the number of red kings on the board
 - x_5 : the number of black pieces threatened by red (i.e. which can be captured on red's next turn)
 - x_6 : the number of red pieces threatened by black

$$\hat{V}(b) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6$$

Partial Design of Checkers Learning Program

- A checkers learning problem:
 - Task T : playing checkers
 - Performance measure P : percent of games won in the world tournament
 - Training experience E : games played against itself
 - *Target Function*: V : Board $\rightarrow \mathbb{R}$
 - *Target function representation*

$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

Choosing a Function Approximation Algorithm

- To learn \hat{V} we require a set of training examples describing the board b and the training value

$$V_{train}(b)$$

- Ordered pair $\langle b, V_{train}(b) \rangle$

$$\langle\langle x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0 \rangle, +100 \rangle$$

Estimating Training Values

- Need to assign specific scores to *intermediate* board states
- Approximate intermediate board state b using the learner's current approximation of the next board state following b

$$V_{train}(b) \leftarrow \hat{V}(\text{Successor}(b))$$

- Simple and successful approach
- More accurate for states closer to end states

Some Issues in Machine Learning

- What Algorithms Can Approximate Functions Well? When?
- How Do Learning System Design Factors Influence Accuracy?
 - Number of training examples
 - Complexity of hypothesis representation
- How Do Learning Problem Characteristics Influence Accuracy?
 - Noisy data
 - Multiple data sources
- What Are The Theoretical Limits of Learnability?
- How Can Prior Knowledge of Learner Help?
- What Clues Can We Get From Biological Learning Systems?
- How Can Systems Alter Their Own Representation?



Styles of Learning

Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University

Introduction

- What is Machine Learning?
- Motivation: Why Machine Learning?
- Applications of Machine Learning
- Designing a Learning System
- Issues in Machine Learning

Algorithms



- The success of machine learning system also depends on the algorithms.
- The algorithms control the search to find and build the knowledge structures.
- The learning algorithms should extract useful information from training examples.

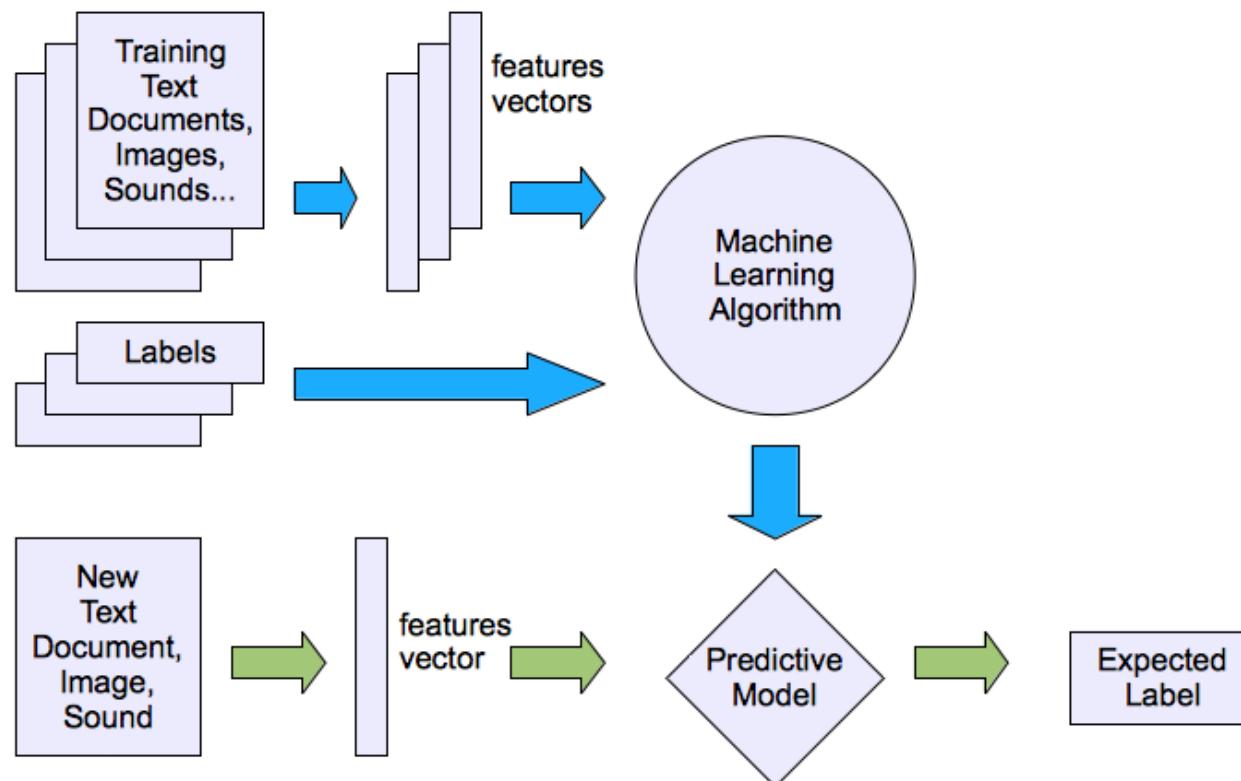
Algorithms



- **Supervised learning**
 - Prediction
 - Classification (discrete labels), Regression (real values)
- **Unsupervised learning**
 - Clustering
 - Probability distribution estimation
 - Finding association (in features)
 - Dimension reduction
- **Semi-supervised learning**
- **Reinforcement learning**
 - Decision making (robot, chess machine)

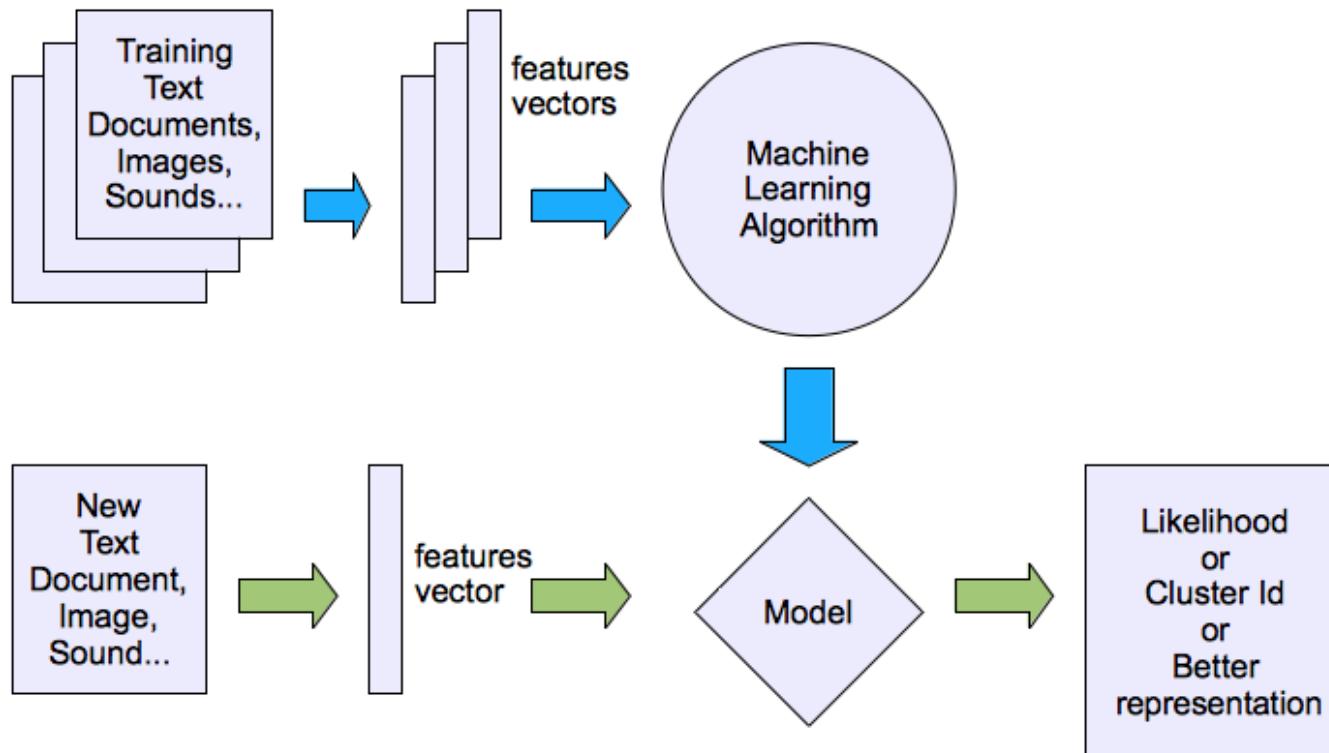
Machine learning structure

■ Supervised learning



Machine learning structure

■ Unsupervised learning



Supervised Learning

- Training data includes both the input and the desired results.
- For some examples the correct results (targets) are known and are given in input to the model during the learning process.
- The construction of a proper training, validation and test set (Bok) is crucial.
- These methods are usually fast and accurate.
- Have to be able to **generalize**: give the correct results when new data are given in input without knowing a priori the target.

Supervised Learning: Uses

Example: decision trees tools that create rules

- Prediction of future cases: Use the rule to predict the output for future inputs
- Knowledge extraction: The rule is easy to understand
- Compression: The rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Examples of Supervised Learning

- Examples of Supervised Learning
 - Classification
 - Spam filtering: Is an email spam or not
 - Image classification: Given an image, output which objects are present in the image (dog, cat, computer, building, so on)
 - Regression
 - Given information about a house, predict its price
 - Netflix: Given a user and a movie, predict the rating the user is going to give to the movie (which can then be used for providing recommendations)

Unsupervised Learning

- The model is not provided with the correct results during the training.
- Can be used to cluster the input data in classes on the basis of their statistical properties only.
- Cluster significance and labeling.
- The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

Unsupervised Learning

- Learning “what normally happens”
- No output
- Clustering: Grouping similar instances
- Other applications: Summarization, Association Analysis
- Example applications
 - Customer segmentation in CRM
 - Image compression: Color quantization
 - Bioinformatics: Learning motifs

Examples of Unsupervised Learning

- Clustering
 - Given a list of customers and information about them, discover groups of similar users. This knowledge can then be used for targeted marketing.
 - Anomaly detection: Given measurements from sensors in a manufacturing facility, identify anomalies, i.e. that something is wrong
- Association
 - Discover patterns in data such as whenever it rains, people tend to stay indoors. When it is hot, people buy more ice-cream.

Reinforcement Learning

- In the above two problem categories, the input is given to us.
- In reinforcement learning, the key difference is that the input itself depends on actions we take.
- For example, in robotics, we might start in a situation where the robot does not know anything about the surrounding it is in. As it does certain actions, it finds out more about the world. But the world it sees depends on whether it chose to move forward, or turn right.
- The robot is known as an agent, and is in some environment (surrounding). At each time step, it can take some action and it might receive some reward (say the robot fell in a ditch, or found a lake on Mars).

Reinforcement Learning

- Topics:
 - Policies: what actions should an agent take in a particular situation
 - Utility estimation: how good is a state (\rightarrow used by policy)
- No supervised output but delayed reward
- Credit assignment problem (what was responsible for the outcome)
- Applications:
 - Game playing
 - Robot in a maze
 - Multiple agents, partial observability, ...

Example reinforcement learning

- Robotics: A robot is in a maze, and it needs to find a way out.
- Training an AI for a complex game such as Civilization or Dota

Semi-Supervised Learning

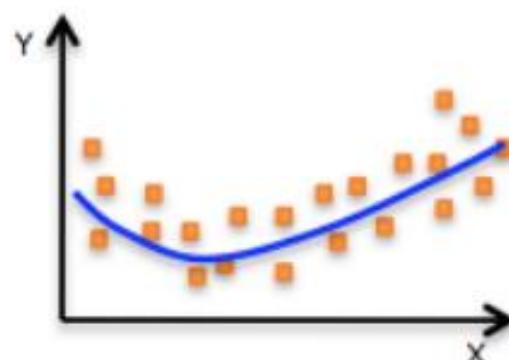
- For example, in the image classification problem, we are given a number of images and the objects present in those images as training data.
- However, we may have a strategy for using the large amount of images available on the web (for which the objects have not been annotated).
- This is an example of semi-supervised learning, i.e. we have some data that is labelled, and some that is not labelled

A common problem: OVERFITTING

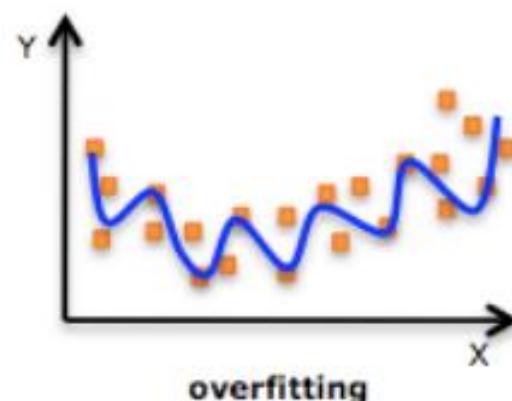
- Learn the “data” and not the underlying function
- Performs well on the data used during the training and poorly with new data.



Underfitting



Just right!



overfitting

How to avoid it: use proper subsets, early stopping.

Fundamentals of Decision Theory

Decision Theory

- “**an analytic and systematic approach to the study of decision making**”

Good decisions:

- based on reasoning
- consider all available data and possible alternatives
- employ a quantitative approach

Bad decisions:

- not based on reasoning
- do not consider all available data and possible alternatives
- do not employ a quantitative approach

- **A good decision may occasionally result in an unexpected outcome; it is still a good decision if made properly**
- **A bad decision may occasionally result in a good outcome if you are lucky; it is still a bad decision**

Steps in Decision Theory

1. List the possible alternatives (actions/decisions)
2. Identify the possible outcomes
3. List the payoff or profit or reward
4. Select one of the decision theory models
5. Apply the model and make your decision

Example

The Thompson Lumber Company

- Problem.
 - The Thompson Lumber Co. must decide whether or not to expand its product line by manufacturing and marketing a new product, backyard storage sheds
- Step 1: List the possible alternatives
 - alternative:* “a course of action or strategy that may be chosen by the decision maker”
 - (1) Construct a large plant to manufacture the sheds
 - (2) Construct a small plant
 - (3) Do nothing

The Thompson Lumber Company

- Step 2: Identify the states of nature
 - (1) The market for storage sheds could be favorable
 - high demand
 - (2) The market for storage sheds could be unfavorable
 - low demand

state of nature: “an outcome over which the decision maker has little or no control”
e.g., lottery, coin-toss, whether it will rain today

The Thompson Lumber Company

- Step 3: List the possible rewards
 - A reward for all possible combinations of alternatives and states of nature
 - *Conditional values*: “reward depends upon the alternative and the state of nature”
 - with a favorable market:
 - a large plant produces a net profit of \$200,000
 - a small plant produces a net profit of \$100,000
 - no plant produces a net profit of \$0
 - with an unfavorable market:
 - a large plant produces a net loss of \$180,000
 - a small plant produces a net loss of \$20,000
 - no plant produces a net profit of \$0

Reward tables

- A means of organizing a decision situation, including the rewards from different situations given the possible states of nature

Actions	States of Nature	
	a	b
1	Reward 1a	Reward 1b
2	Reward 2a	Reward 2b

- Each decision, 1 or 2, results in an outcome, or reward, for the particular state of nature that occurs in the future
- May be possible to assign probabilities to the states of nature to aid in selecting the best outcome

The Thompson Lumber Company

	States of Nature	
Actions		

The Thompson Lumber Company

	States of Nature	
Actions	Favorable Market	Unfavorable Market
Large plant	\$200,000	-\$180,000
Small plant	\$100,000	-\$20,000
No plant	\$0	\$0

The Thompson Lumber Company

- Steps 4/5: Select an appropriate model and apply it
 - Model selection depends on the operating environment and degree of uncertainty

Decision Making Environments

- Decision making under certainty
- Decision making under uncertainty
 - Non-deterministic uncertainty
 - Probabilistic uncertainty (risk)

Decision Making Under Certainty

- Decision makers know with certainty the consequences of every decision alternative
 - Always choose the alternative that results in the best possible outcome

Non-deterministic Uncertainty

	States of Nature	
Actions	Favorable Market	Unfavorable Market
Large plant	\$200,000	-\$180,000
Small plant	\$100,000	-\$20,000
No plant	\$0	\$0

- What should we do?

Maximax Criterion

“Go for the Gold”

- Select the decision that results in the maximum of the maximum rewards
- A very optimistic decision criterion
 - Decision maker assumes that the most favorable state of nature for each action will occur
- Most risk prone agent

Maximax

Decision	States of Nature		Maximum in Row
	Favorable	Unfavorable	
Large plant	\$200,000	-\$180,000	\$200,000
Small plant	\$100,000	-\$20,000	\$100,000
No plant	\$0	\$0	\$0

- Thompson Lumber Co. assumes that the most favorable state of nature occurs for each decision alternative
- Select the maximum reward for each decision
 - All three maximums occur if a favorable economy prevails (a tie in case of no plant)
- Select the maximum of the maximums
 - Maximum is \$200,000; corresponding decision is to build the large plant
 - Potential loss of \$180,000 is completely ignored

Maximin Criterion

“Best of the Worst”

- Select the decision that results in the maximum of the minimum rewards
- A very pessimistic decision criterion
 - Decision maker assumes that the minimum reward occurs for each decision alternative
 - Select the maximum of these minimum rewards
- Most risk averse agent

Maximin

Decision	States of Nature		Minimum in Row
	Favorable	Unfavorable	
Large plant	\$200,000	-\$180,000	-\$180,000
Small plant	\$100,000	-\$20,000	-\$20,000
No plant	\$0	\$0	\$0

- Thompson Lumber Co. assumes that the least favorable state of nature occurs for each decision alternative
- Select the minimum reward for each decision
 - All three minimums occur if an unfavorable economy prevails (a tie in case of no plant)
- Select the maximum of the minimums
 - Maximum is \$0; corresponding decision is to do nothing
 - A conservative decision; largest possible gain, \$0, is much less than maximax

Equal Likelihood Criterion

- Assumes that all states of nature are equally likely to occur
 - Maximax criterion assumed the most favorable state of nature occurs for each decision
 - Maximin criterion assumed the least favorable state of nature occurs for each decision
- Calculate the *average reward* for each alternative and select the alternative with the maximum number
 - Average reward: the sum of all rewards divided by the number of states of nature
- Select the decision that gives the highest average reward

Equal Likelihood

Decision	States of Nature		Row Average
	Favorable	Unfavorable	
Large plant	\$200,000	-\$180,000	\$10,000
Small plant	\$100,000	-\$20,000	\$40,000
No plant	\$0	\$0	\$0

Row Averages

$$\text{Large Plant} = \frac{\$200,000 - \$180,000}{2} = \$10,000$$

$$\text{Small Plant} = \frac{\$100,000 - \$20,000}{2} = \$40,000$$

$$\text{Do Nothing} = \frac{\$0 + \$0}{2} = \$0$$

- Select the decision with the highest weighted value
 - Maximum is \$40,000; corresponding decision is to build the small plant

Criterion of Realism

- Also known as the weighted average or Hurwicz criterion
 - A compromise between an optimistic and pessimistic decision
- A coefficient of realism, α , is selected by the decision maker to indicate optimism or pessimism about the future

$$0 \leq \alpha \leq 1$$

When α is close to 1, the decision maker is optimistic.
When α is close to 0, the decision maker is pessimistic.

- *Criterion of realism* = $\alpha(\text{row maximum}) + (1-\alpha)(\text{row minimum})$
 - A weighted average where maximum and minimum rewards are weighted by α and $(1 - \alpha)$ respectively

Criterion of Realism

- Assume a coefficient of realism equal to 0.8

Decision	States of Nature		Criterion of Realism
	Favorable	Unfavorable	
Large plant	\$200,000	-\$180,000	\$124,000
Small plant	\$100,000	-\$20,000	\$76,000
No plant	\$0	\$0	\$0

Weighted Averages

$$\text{Large Plant} = (0.8)(\$200,000) + (0.2)(-\$180,000) = \$124,000$$

$$\text{Small Plant} = (0.8)(\$100,000) + (0.2)(-\$20,000) = \$76,000$$

$$\text{Do Nothing} = (0.8)(\$0) + (0.2)(\$0) = \$0$$

Select the decision with the highest weighted value

**Maximum is \$124,000; corresponding decision
is to build the large plant**

Minimax Regret

- Regret/Opportunity Loss: “the difference between the optimal reward and the actual reward received”
- Choose the alternative that minimizes the maximum regret associated with each alternative
 - Start by determining the maximum regret for each alternative
 - Pick the alternative with the minimum number

Regret Table

- If I knew the future, how much I'd regret my decision...
- Regret for any state of nature is calculated by subtracting each outcome in the column from the best outcome in the same column

Minimax Regret

Decision	States of Nature					Row Maximum	
	Favorable		Unfavorable				
	Payoff	Regret	Payoff	Regret			
Large plant	\$200,000	\$0	-\$180,000	\$180,000	\$180,000		
Small plant	\$100,000	\$100,000	-\$20,000	\$20,000	\$100,000		
No plant	\$0	\$200,000	\$0	\$0	\$200,000		
Best payoff	\$200,000		\$0				

- Select the alternative with the lowest maximum regret

Minimum is \$100,000; corresponding decision is to build a small plant

Summary of Results

Criterion	Decision
Maximax	Build a large plant
Maximin	Do nothing
Equal likelihood	Build a small plant
Realism	Build a large plant
Minimax regret	Build a small plant

Decision Making Environments

- Decision making under certainty
- Decision making under uncertainty
 - Non-deterministic uncertainty
 - Probabilistic uncertainty (risk)

Probabilistic Uncertainty

- Decision makers know the probability of occurrence for each possible outcome
 - Attempt to maximize the expected reward
- Criteria for decision models in this environment:
 - Maximization of expected reward
 - Minimization of expected regret
 - Minimize expected regret = maximizing expected reward!

Expected Reward (Q)

- called Expected Monetary Value (EMV) in DT literature
- “the probability weighted sum of possible rewards for each alternative”
 - Requires a reward table with conditional rewards and probability assessments for all states of nature

$$\begin{aligned} Q(\text{action } a) = & \text{ (reward of 1st state of nature)} \\ & \times (\text{probability of 1st state of nature}) \\ & + \text{ (reward of 2nd state of nature)} \\ & \times (\text{probability of 2nd state of nature}) \\ & + \dots + \text{ (reward of last state of nature)} \\ & \times (\text{probability of last state of nature}) \end{aligned}$$

The Thompson Lumber Company

- Suppose that the probability of a favorable market is exactly the same as the probability of an unfavorable market. Which alternative would give the greatest Q?

	States of Nature		EMV
	Favorable Mkt $p = 0.5$	Unfavorable Mkt $p = 0.5$	
Decision			
Large plant	\$200,000	-\$180,000	\$10,000
Small plant	\$100,000	-\$20,000	\$40,000
No plant	\$0	\$0	\$0

$$Q(\text{large plant}) = (0.5)(\$200,000) + (0.5)(-\$180,000) = \$10,000$$

$$Q(\text{small plant}) = (0.5)(\$100,000) + (0.5)(-\$20,000) = \$40,000$$

$$Q(\text{no plant}) = (0.5)(\$0) + (0.5)(\$0) = \$0$$

Build the small plant

Expected Value of Perfect Information (EVPI)

- It may be possible to purchase additional information about future events and thus make a better decision
 - Thompson Lumber Co. could hire an economist to analyze the economy in order to more accurately determine which economic condition will occur in the future
 - How valuable would this information be?

EVPI Computation

- Look first at the decisions under each state of nature
 - If information was available that perfectly predicted which state of nature was going to occur, the best decision for that state of nature could be made
 - *expected value with perfect information* (EV w/ PI): “the expected or average return if we have perfect information before a decision has to be made”

EVPI Computation

- Perfect information changes environment from decision making under risk to decision making with certainty
 - Build the large plant if you know for sure that a favorable market will prevail
 - Do nothing if you know for sure that an unfavorable market will prevail

Decision	States of Nature	
	Favorable $p = 0.5$	Unfavorable $p = 0.5$
Large plant	\$200,000	-\$180,000
Small plant	\$100,000	-\$20,000
No plant	\$0	\$0

EVPI Computation

- Even though perfect information enables Thompson Lumber Co. to make the correct investment decision, each state of nature occurs only a certain portion of the time
 - A favorable market occurs 50% of the time and an unfavorable market occurs 50% of the time
 - EV w/ PI calculated by choosing the best alternative for each state of nature and multiplying its reward times the probability of occurrence of the state of nature

EVPI Computation

EV w/ PI = (best reward for 1st state of nature)
 X (probability of 1st state of nature)
 + (best reward for 2nd state of nature)
 X (probability of 2nd state of nature)

EV w/ PI = (\$200,000)(0.5) + (\$0)(0.5) = \$100,000

Decision	States of Nature	
	Favorable $p = 0.5$	Unfavorable $p = 0.5$
Large plant	\$200,000	-\$180,000
Small plant	\$100,000	-\$20,000
No plant	\$0	\$0

EVPI Computation

- Thompson Lumber Co. would be foolish to pay more for this information than the extra profit that would be gained from having it
 - *EVPI*: “the maximum amount a decision maker would pay for additional information resulting in a decision better than one made *without perfect information*”
 - EVPI is the expected outcome with perfect information minus the expected outcome without perfect information

$$\text{EVPI} = \text{EV w/ PI} - \text{Q}$$

$$\text{EVPI} = \$100,000 - \$40,000 = \$60,000$$

Using EVPI

- EVPI of \$60,000 is the maximum amount that Thompson Lumber Co. should pay to purchase perfect information from a source such as an economist
 - “Perfect” information is extremely rare
 - An investor typically would be willing to pay some amount less than \$60,000, depending on how reliable the information is perceived to be

Is Expected Value sufficient?

- Lottery 1
 - returns \$0 always
- Lottery 2
 - return \$100 and -\$100 with prob 0.5
- Which is better?

Is Expected Value sufficient?

- Lottery 1
 - returns \$100 always
- Lottery 2
 - return \$10000 (prob 0.01) and \$0 with prob 0.99
- Which is better?
 - depends

Is Expected Value sufficient?

- Lottery 1
 - returns \$3125 always
- Lottery 2
 - return \$4000 (prob 0.75) and -\$500 with prob 0.25
- Which is better?

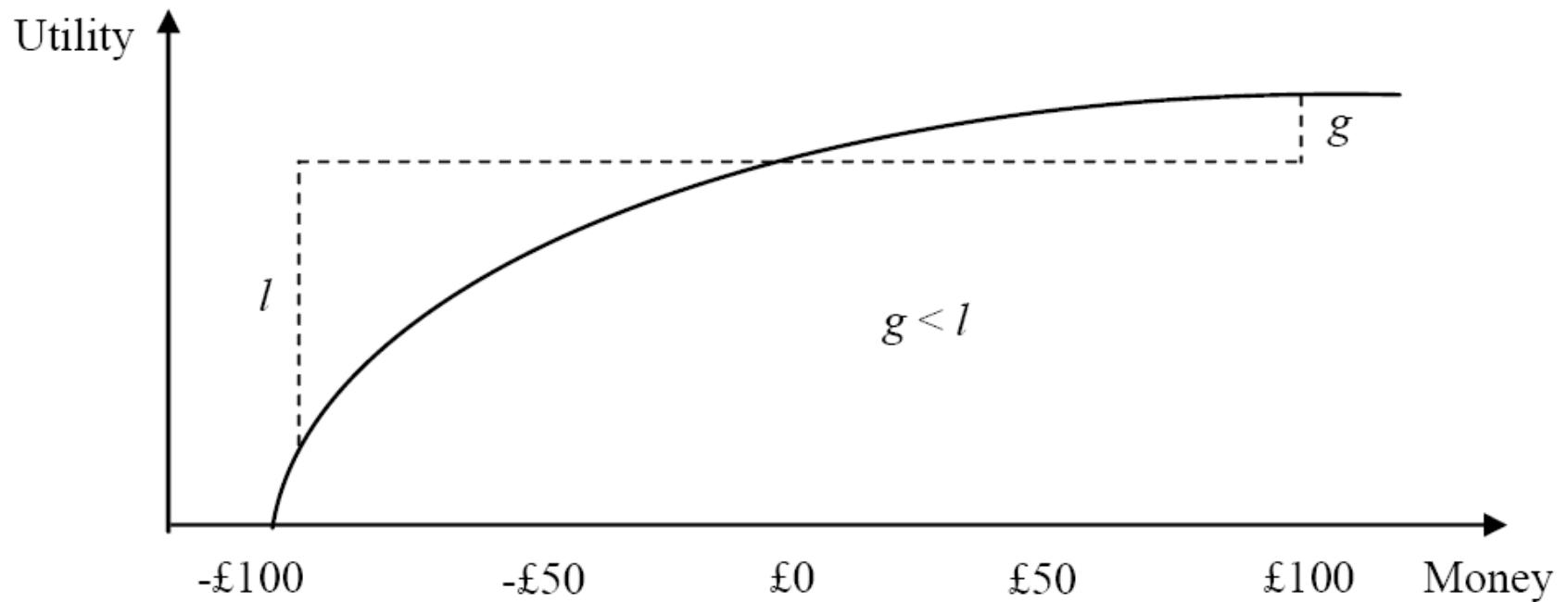
Is Expected Value sufficient?

- Lottery 1
 - returns \$0 always
- Lottery 2
 - return \$1,000,000 (prob 0.5) and -\$1,000,000 with prob 0.5
- Which is better?

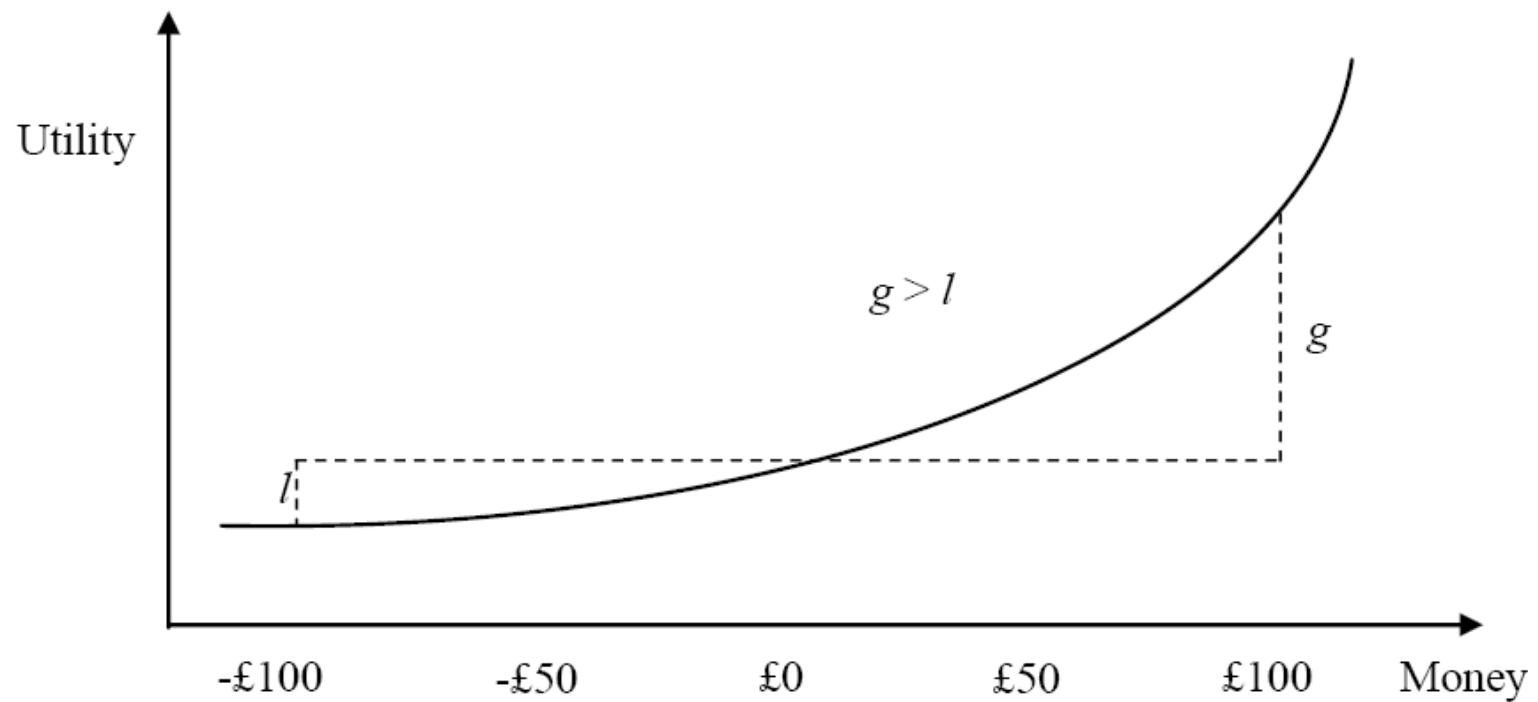
Utility Theory

- Adds a layer of utility over rewards
- Risk averse
 - $|\text{Utility}|$ of high negative money is much MORE than utility of high positive money
- Risk prone
 - Reverse
- Use expected utility criteria...

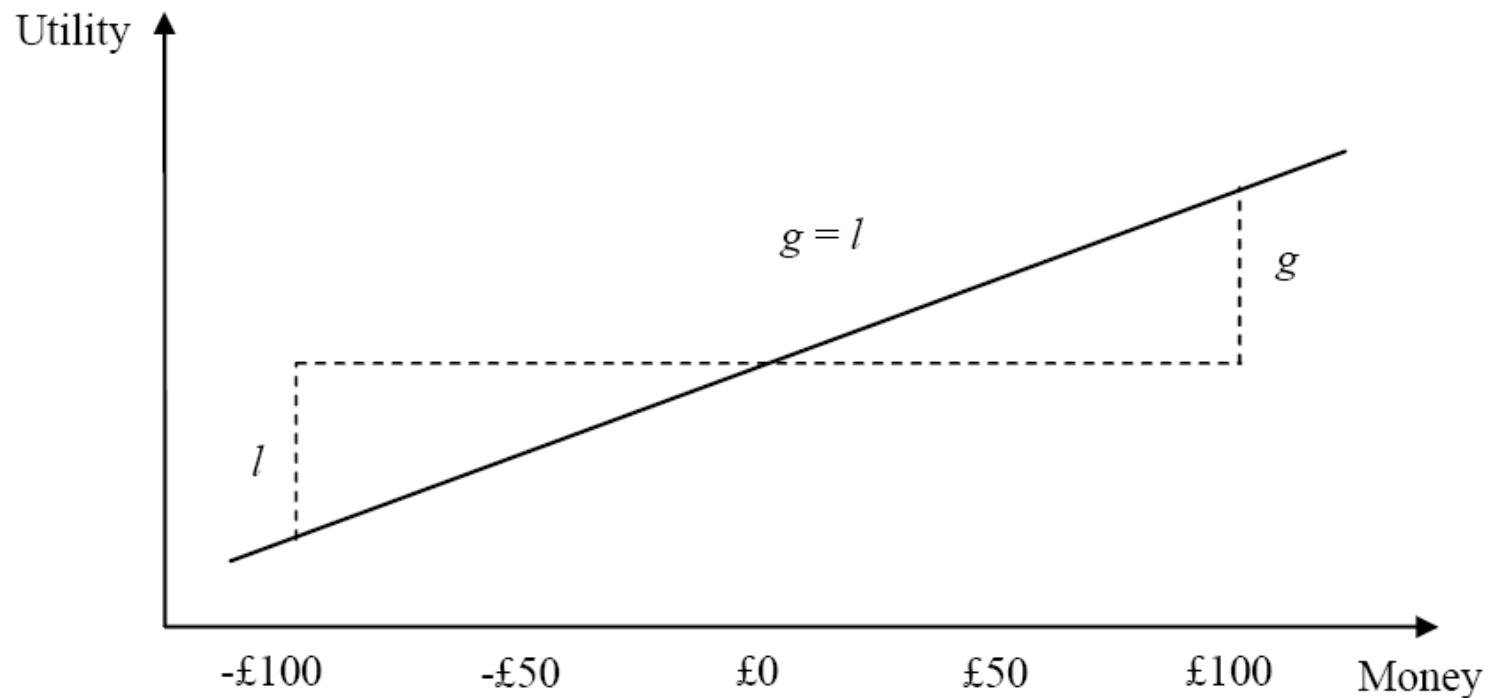
Utility function of risk-averse agent



Utility function of a risk-prone agent



Utility function of a risk-neutral agent



PEAS/Environment

- Performance: utility
- Environment
 - Static – Stochastic – Partially Obs – Discrete –
Episodic – Single
- Actuators
 - alternatives
 - ask for perfect information
- Sensor
 - State of nature



Decision Trees

Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University

Decision Trees

- Can be used instead of a table to show alternatives, outcomes, and payoffs
- Consists of nodes and arcs
- Shows the order of decisions and outcomes

Thompson Lumber Co. Example

Payoffs:

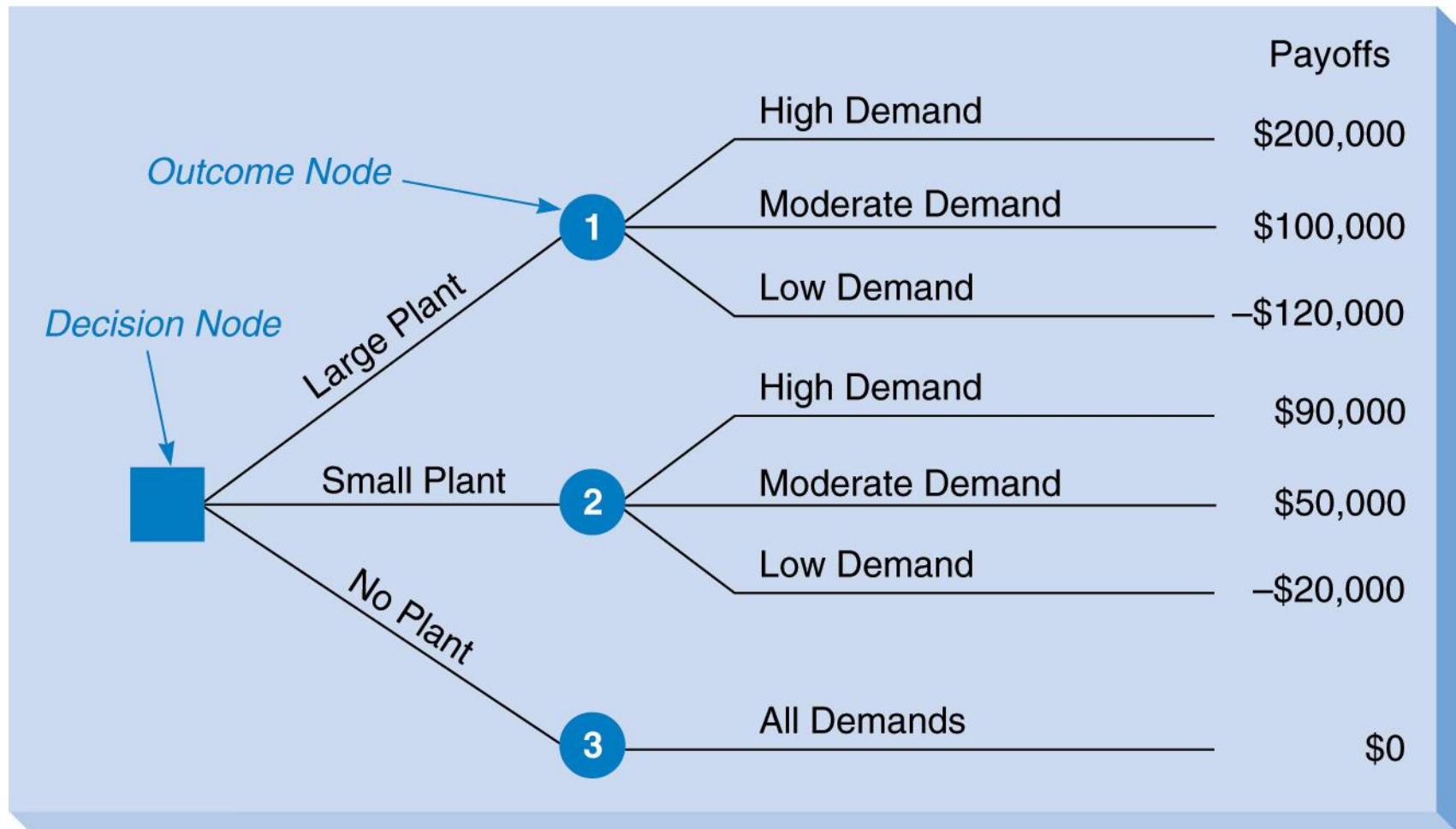
Alternatives	Outcomes (Demand)		
	High	Moderate	Low
Large plant	200,000	100,000	-120,000
Small plant	90,000	50,000	-20,000
No plant	0	0	0

Expected Value of Perfect Information

$$\begin{aligned}\text{EVPI} &= \text{EVwPI} - \text{EMV} \\ &= \$110,000 - \$86,000 = \$24,000\end{aligned}$$

- The “perfect information” increases the expected value by \$24,000
- Would it be worth \$30,000 to obtain this perfect information for demand?

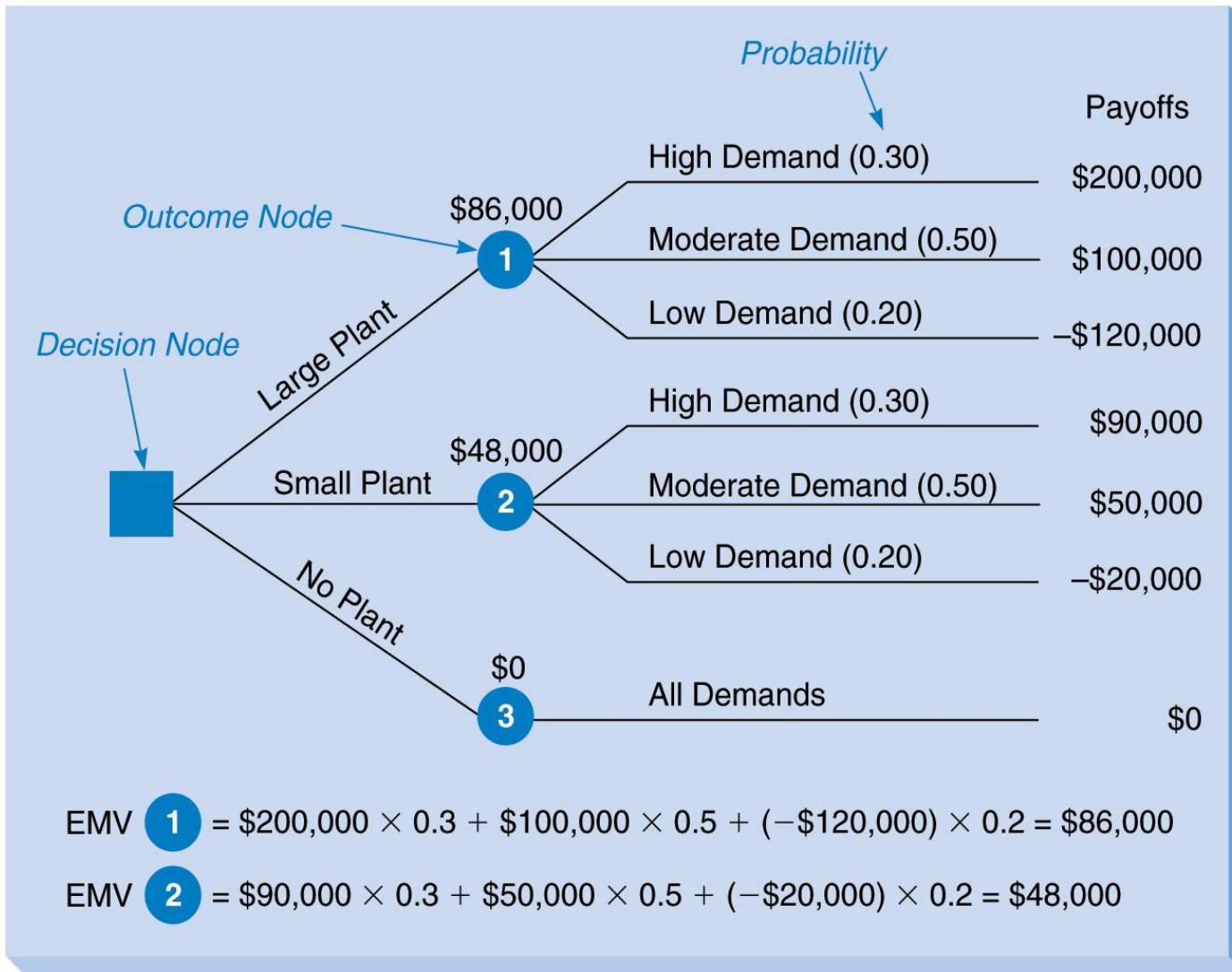
Decision Tree for Thompson Lumber



Folding Back a Decision Tree

- For identifying the best decision in the tree
- Work from right to left
- Calculate the expected payoff at each outcome node
- Choose the best alternative at each decision node (based on expected payoff)

Thompson Lumber Tree with EMV's



Using TreePlan With Excel

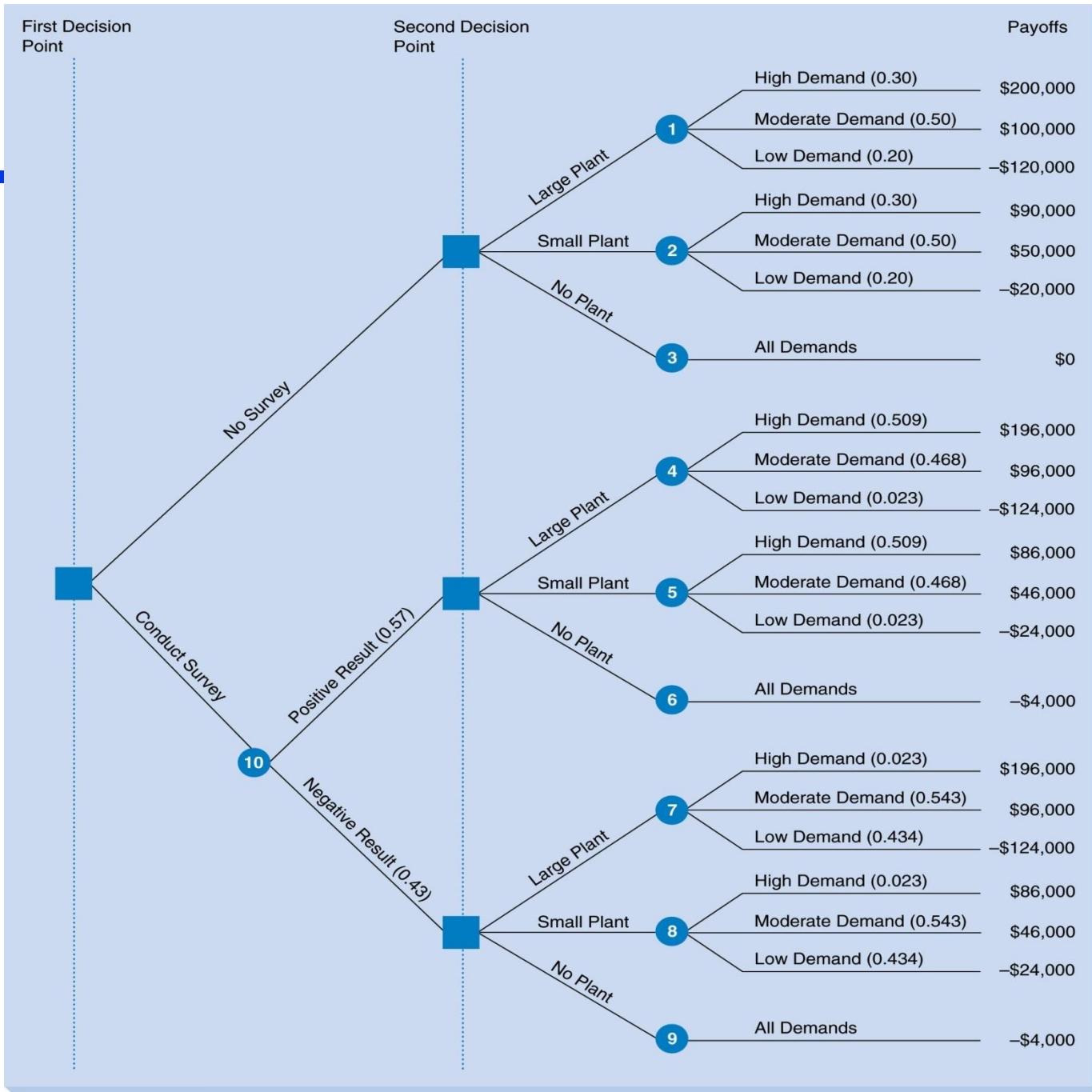
- An add-in for Excel to create and solve decision trees
- Load the file [Treeplan.xla](#) into Excel
(from the CD-ROM)

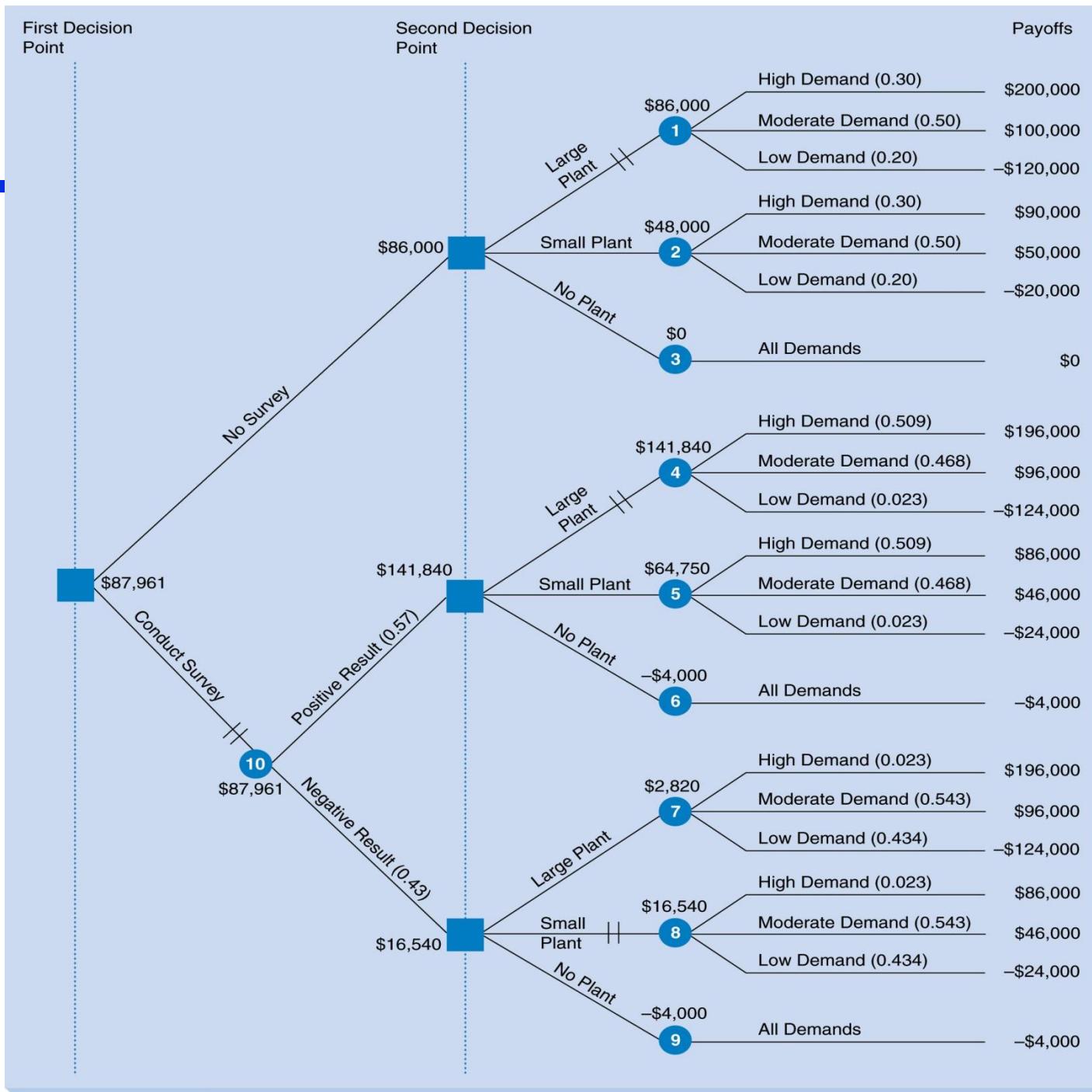
Decision Trees for Multistage Decision-Making Problems

- Multistage problems involve a sequence of several decisions and outcomes
- It is possible for a decision to be immediately followed by another decision
- Decision trees are best for showing the sequential arrangement

Expanded Thompson Lumber Example

- Suppose they will first decide whether to pay \$4000 to conduct a market survey
- Survey results will be imperfect
- Then they will decide whether to build a large plant, small plant, or no plant
- Then they will find out what the outcome and payoff are





Thompson Lumber Optimal Strategy

1. Conduct the survey
2. If the survey results are positive, then build the large plant ($EMV = \$141,840$)
If the survey results are negative, then build the small plant ($EMV = \$16,540$)

Expected Value of Sample Information (EVSI)

- The Thompson Lumber survey provides sample information (not perfect information)
- What is the value of this sample information?
 $\text{EVSI} = (\text{EMV with } \textit{free} \text{ sample information}) - (\text{EMV w/o any information})$

EVSI for Thompson Lumber

If sample information had been free

$$\text{EMV (with free SI)} = 87,961 + 4000 = \\ \$91,961$$

$$\text{EVSI} = 91,961 - 86,000 = \$5,961$$

EVSI vs. EVPI

How close does the sample information come to perfect information?

Efficiency of sample information = $\frac{\text{EVSI}}{\text{EVPI}}$

Thompson Lumber: $5961 / 24,000 = 0.248$

Estimating Probability Using Bayesian Analysis

- Allows probability values to be revised based on new information (from a survey or test market)
- **Prior probabilities** are the probability values before new information
- **Revised probabilities** are obtained by combining the prior probabilities with the new information

Estimating Probability Using Bayesian Analysis

Known **Prior** Probabilities

$$P(HD) = 0.30$$

$$P(MD) = 0.50$$

$$P(LD) = 0.30$$

How do we find the **revised** probabilities where the survey result is **given**?

For example: $P(HD|PS) = ?$

Estimating Probability Using Bayesian Analysis

- It is necessary to understand the **Conditional probability** formula:
$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)}$$
- $P(A|B)$ is the probability of event A occurring, given that event B has occurred
- When $P(A|B) \neq P(A)$, this means the probability of event A has been **revised** based on the fact that event B has occurred

Estimating Probability Using Bayesian Analysis

The marketing research firm provided the following probabilities based on its track record of survey accuracy:

$$P(PS|HD) = 0.967$$

$$P(NS|HD) = 0.033$$

$$P(PS|MD) = 0.533$$

$$P(NS|MD) = 0.467$$

$$P(PS|LD) = 0.067$$

$$P(NS|LD) = 0.933$$

Here the demand is “given,” but we need to reverse the events so the survey result is “given”

Estimating Probability Using Bayesian Analysis

- Finding probability of the demand outcome given the survey result:

$$P(HD|PS) = \frac{P(HD \text{ and } PS)}{P(PS)} = \frac{P(PS|HD) \times P(HD)}{P(PS)}$$

- Known probability values are in blue, so need to find $P(PS)$

$$\begin{aligned} & P(PS|HD) \times P(HD) && 0.967 \times 0.30 \\ & + P(PS|MD) \times P(MD) && + 0.533 \times 0.50 \\ & + \underline{P(PS|LD) \times P(LD)} && \underline{+ 0.067 \times 0.20} \\ & = P(PS) && = 0.57 \end{aligned}$$

Estimating Probability Using Bayesian Analysis

- Now we can calculate $P(HD|PS)$:

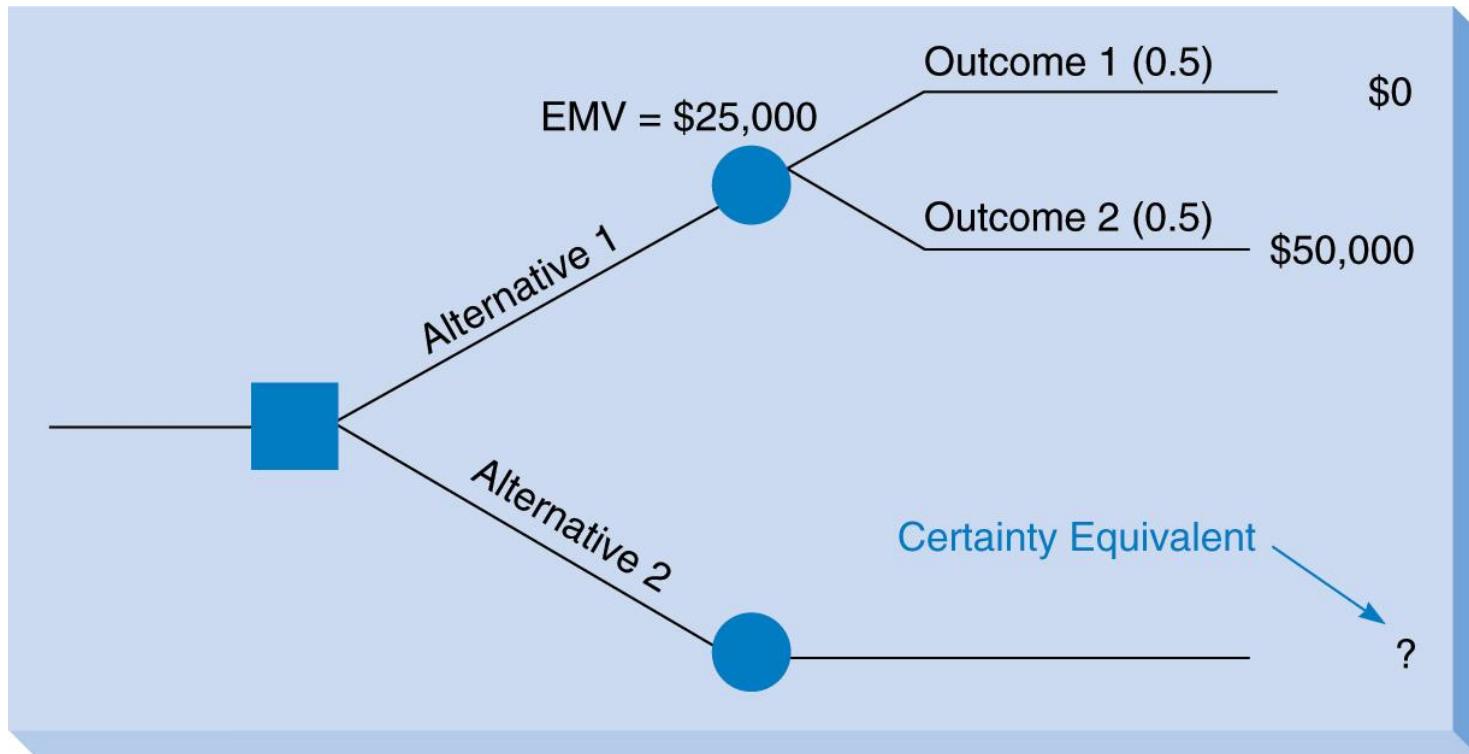
$$P(HD|PS) = \frac{P(PS|HD) \times P(HD)}{P(PS)} = \frac{0.967 \times 0.30}{0.57} = 0.509$$

- The other five conditional probabilities are found in the same manner
- Notice that the probability of HD increased from 0.30 to 0.509 given the positive survey result

Utility Theory

- An alternative to EMV
- People view risk and money differently, so EMV is not always the best criterion
- Utility theory incorporates a person's attitude toward risk
- A **utility function** converts a person's attitude toward money and risk into a number between 0 and 1

Jane's Utility Assessment



Jane is asked: What is the minimum amount that would cause you to choose alternative 2?

Jane's Utility Assessment

- Suppose Jane says \$15,000
- Jane would rather have the certainty of getting \$15,000 rather the possibility of getting \$50,000

- Utility calculation:

$$U(\$15,000) = U(\$0) \times 0.5 + U(\$50,000) \times 0.5$$

Where, $U(\$0) = U(\text{worst payoff}) = 0$

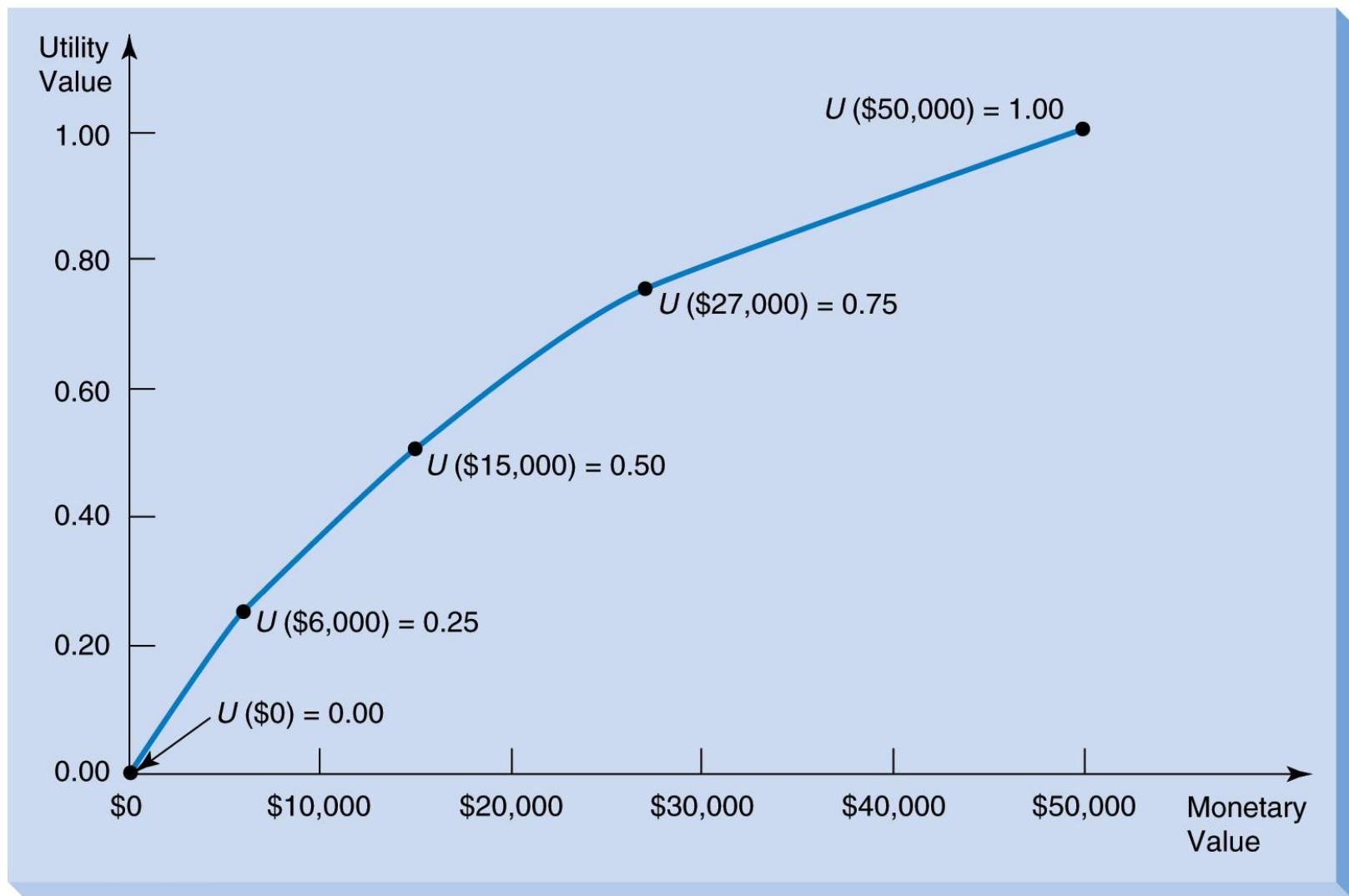
$U(\$50,000) = U(\text{best payoff}) = 1$

$$U(\$15,000) = 0 \times 0.5 + 1 \times 0.5 = 0.5 \quad (\text{for Jane})$$

Jane's Utility Assessment

- The same gamble is presented to Jane multiple times with various values for the two payoffs
- Each time Jane chooses her minimum certainty equivalent and her utility value is calculated
- A **utility curve** plots these values

Jane's Utility Curve



Risk premium

- Different people will have different curves
- Jane's curve is typical of a **risk avoider**
- **Risk premium** is the EMV a person is willing to give up to avoid the risk

Risk premium = (EMV of gamble)
– (Certainty equivalent)

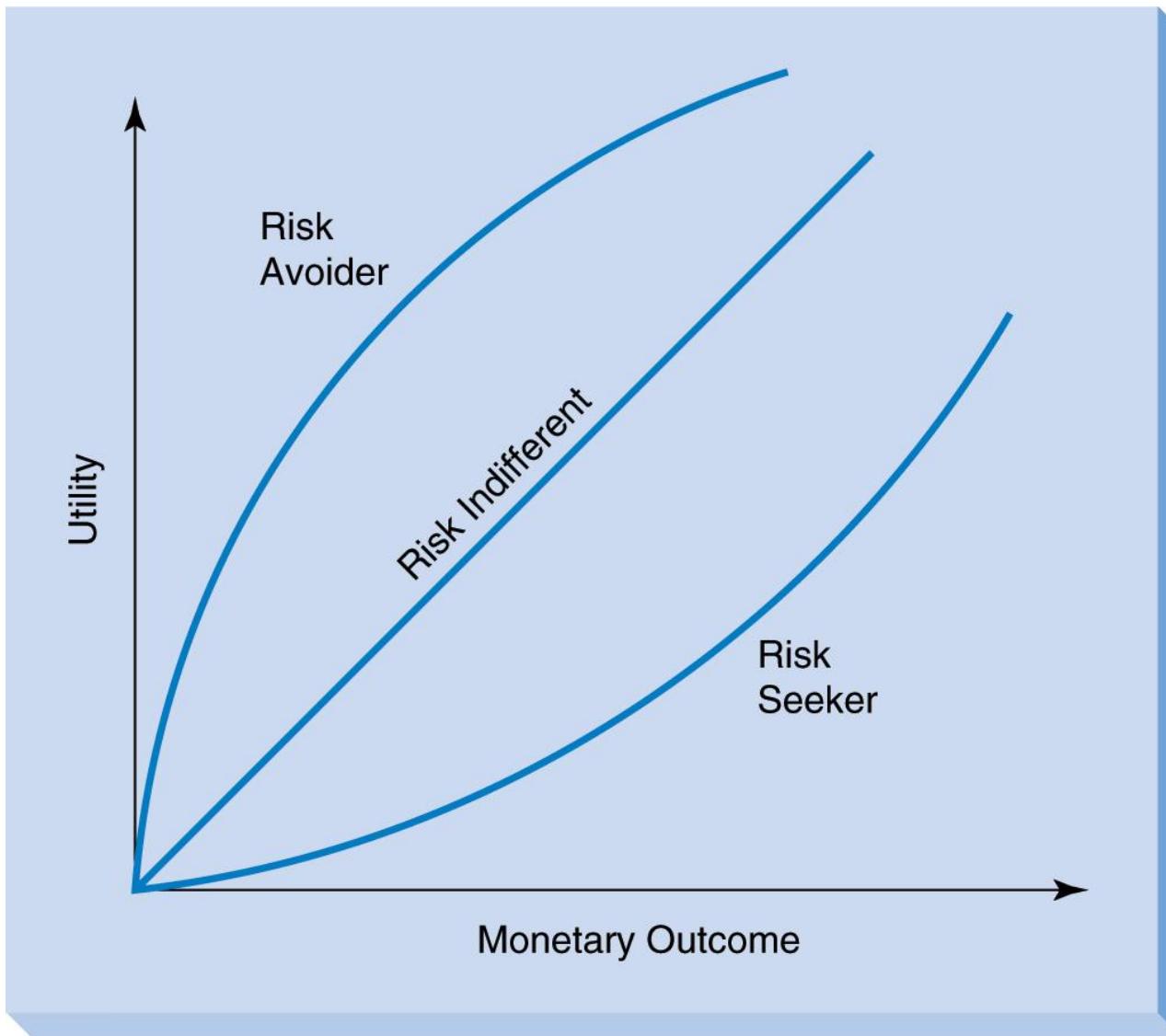
$$\begin{aligned}\text{Jane's risk premium} &= \$25,000 - \$15,000 \\ &= \$10,000\end{aligned}$$

Types of Decision Makers

Risk Premium

- Risk avoiders: > 0
- Risk neutral people: $= 0$
- Risk seekers: < 0

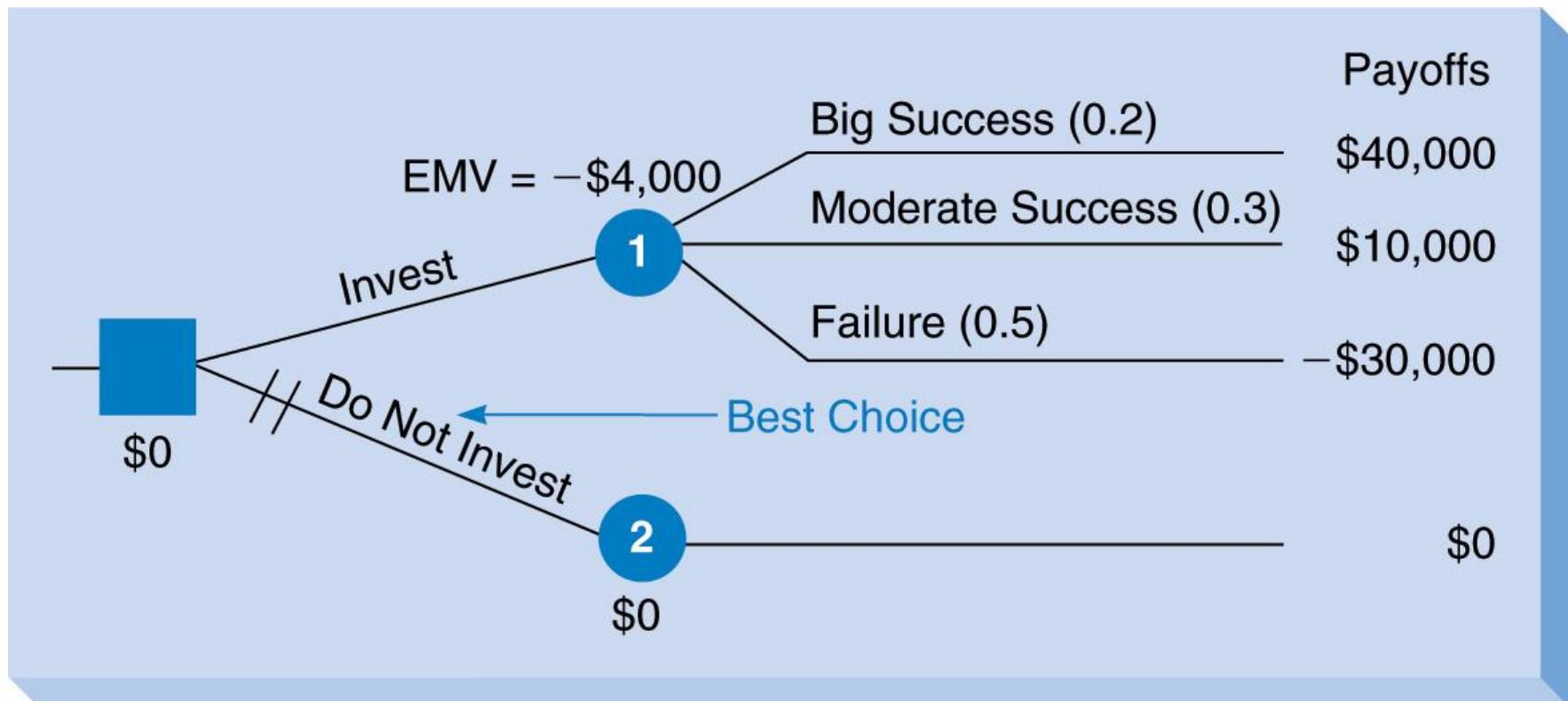
Utility Curves for Different Risk Preferences



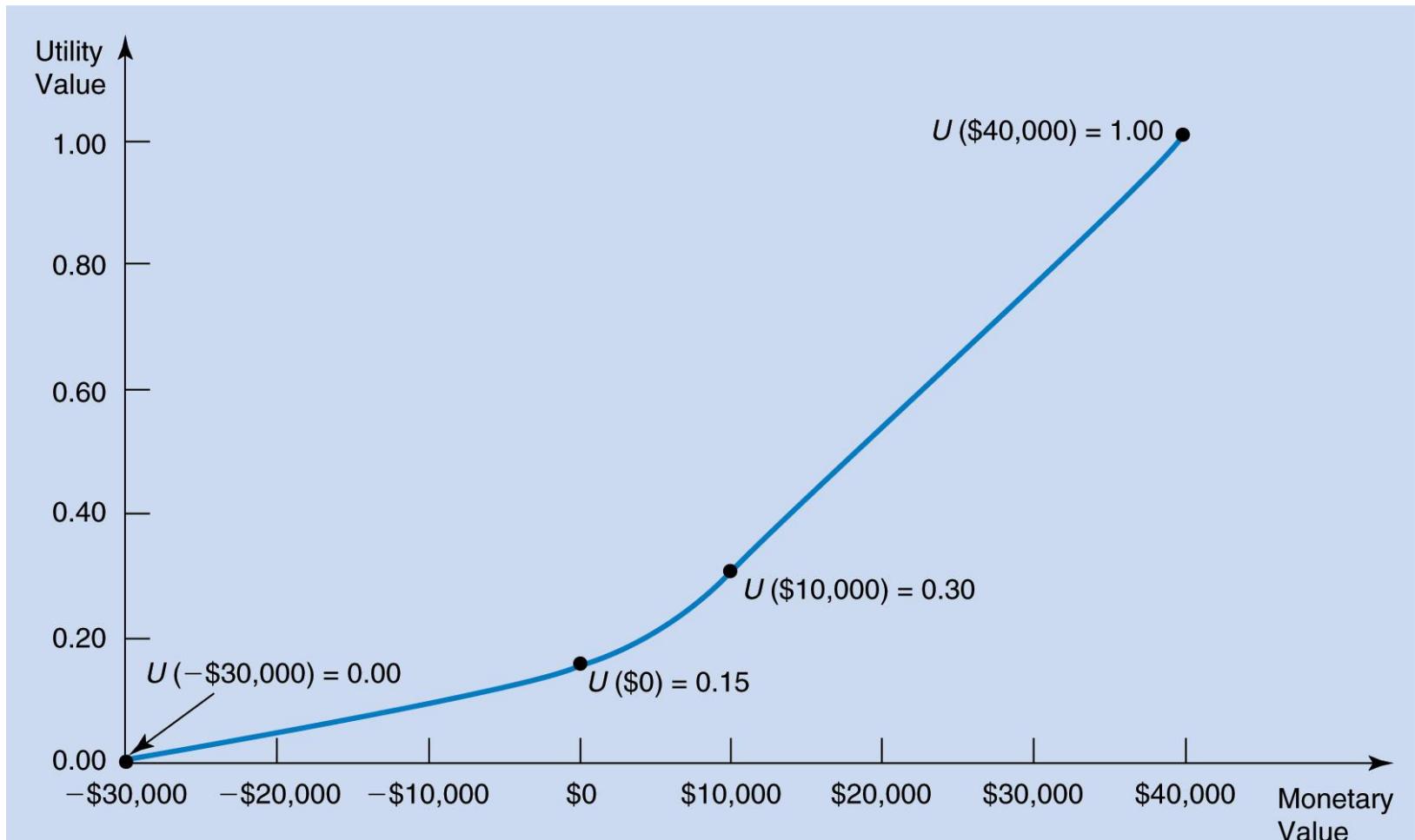
Utility as a Decision Making Criterion

- Construct the decision tree as usual with the same alternative, outcomes, and probabilities
- Utility values replace monetary values
- Fold back as usual calculating expected utility values

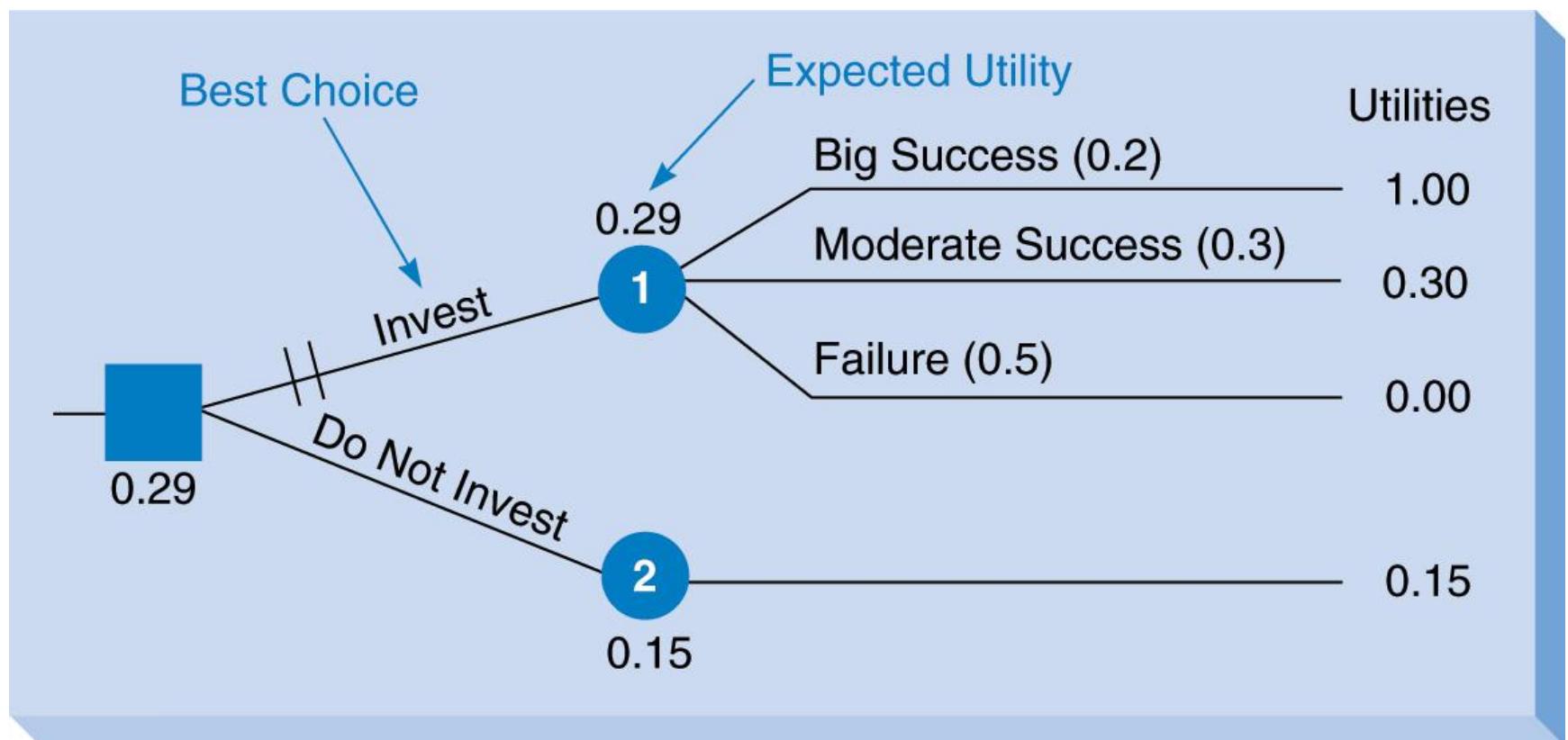
Decision Tree Example for Mark

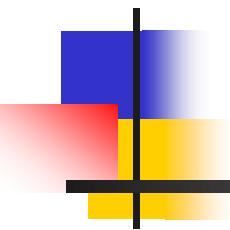


Utility Curve for Mark the Risk Seeker



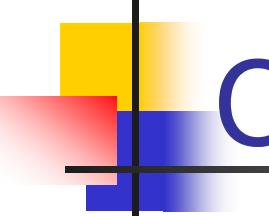
Mark's Decision Tree With Utility Values





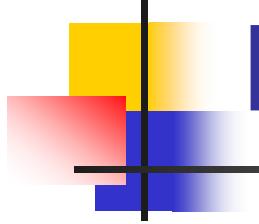
Linear Regression

Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University



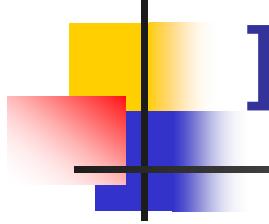
Continuous outcome (means)

Outcome Variable	Are the observations independent or correlated?		Alternatives if the normality assumption is violated (and small sample size):
	Independent	correlated	
Continuous (e.g. pain scale, cognitive function)	Ttest: compares means between two independent groups ANOVA: compares means between more than two independent groups Pearson's correlation coefficient (linear correlation): shows linear correlation between two continuous variables	Paired ttest: compares means between two related groups (e.g., the same subjects before and after) Repeated-measures ANOVA: compares changes over time in the means of two or more groups (repeated measurements) Mixed models/GEE modeling: multivariate regression techniques to compare changes over time between two or more groups; gives rate of change over time	<u>Non-parametric statistics</u> Wilcoxon sign-rank test: non-parametric alternative to the paired ttest Wilcoxon sum-rank test (=Mann-Whitney U test): non-parametric alternative to the ttest Kruskal-Wallis test: non-parametric alternative to ANOVA Spearman rank correlation coefficient: non-parametric alternative to Pearson's correlation coefficient



Recall: Covariance

$$\text{cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{n - 1}$$

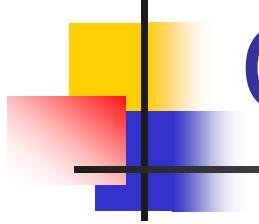


Interpreting Covariance

$\text{cov}(X,Y) > 0 \rightarrow X \text{ and } Y \text{ are positively correlated}$

$\text{cov}(X,Y) < 0 \rightarrow X \text{ and } Y \text{ are inversely correlated}$

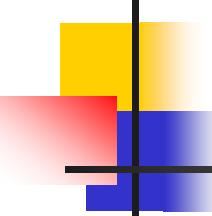
$\text{cov}(X,Y) = 0 \rightarrow X \text{ and } Y \text{ are independent}$



Correlation coefficient

- Pearson's Correlation Coefficient is standardized covariance (unitless):

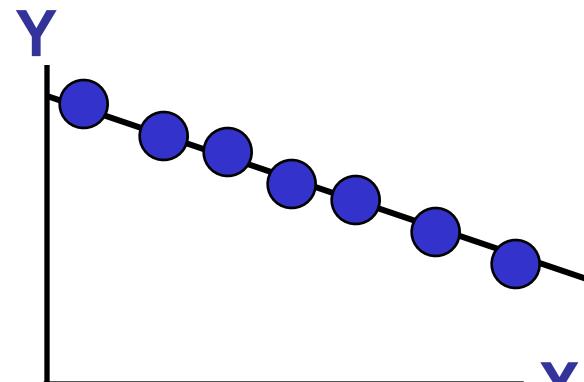
$$r = \frac{\text{covariance}(x, y)}{\sqrt{\text{var } x} \sqrt{\text{var } y}}$$



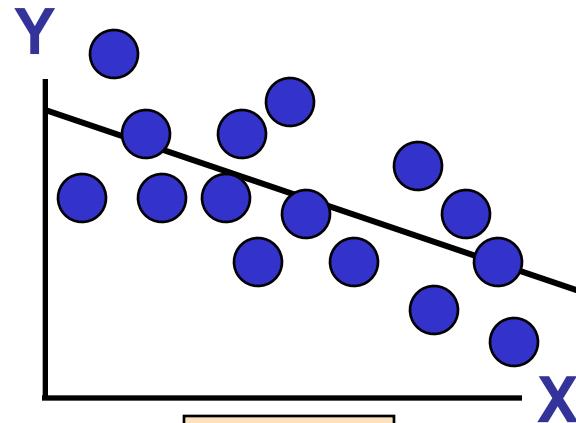
Correlation

- Measures the relative strength of the *linear* relationship between two variables
- Unit-less
- Ranges between -1 and 1
- The closer to -1 , the stronger the negative linear relationship
- The closer to 1 , the stronger the positive linear relationship
- The closer to 0 , the weaker any positive linear relationship

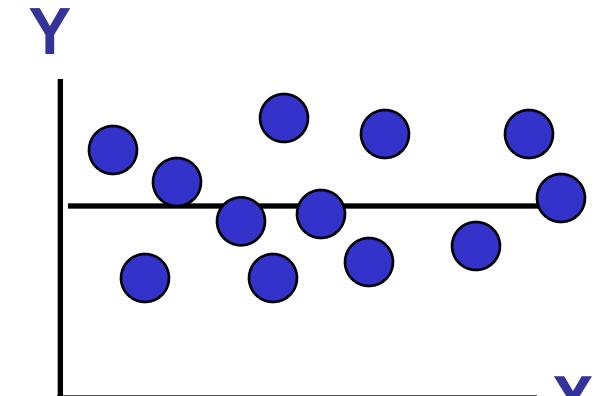
Scatter Plots of Data with Various Correlation Coefficients



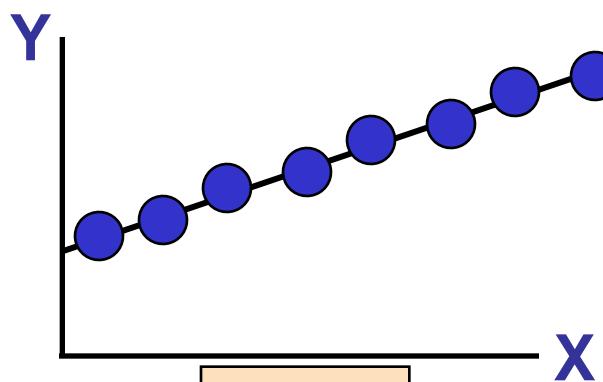
$$r = -1$$



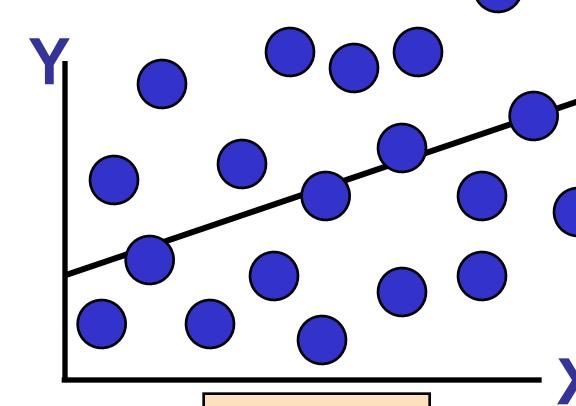
$$r = -.6$$



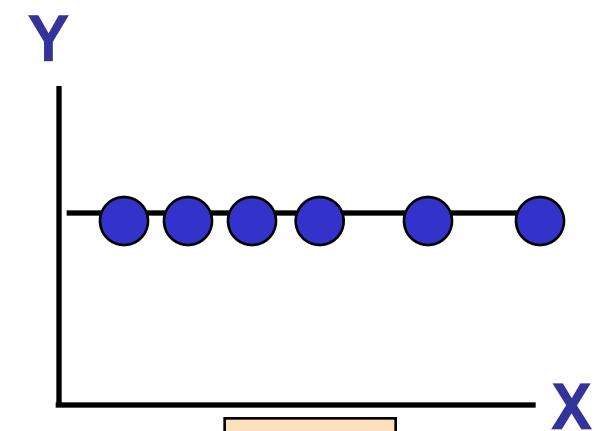
$$r = 0$$



$$r = +1$$



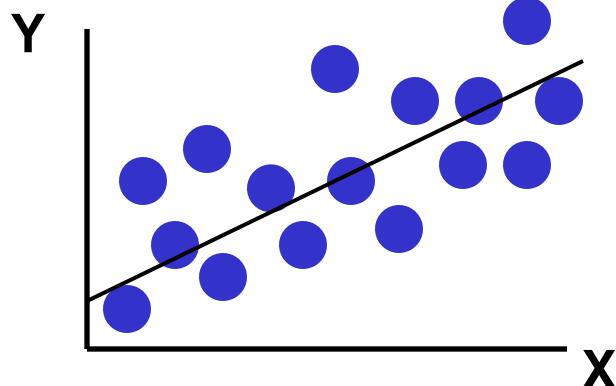
$$r = +.3$$



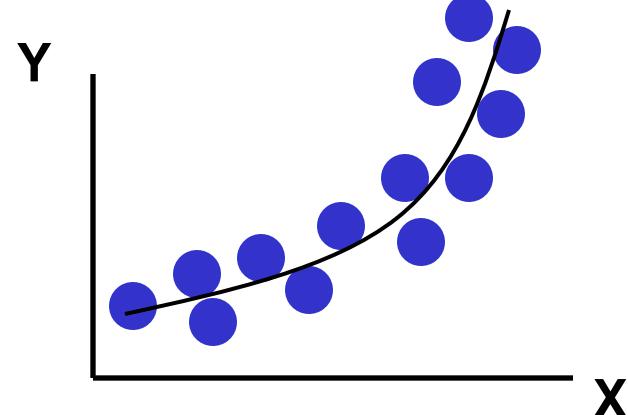
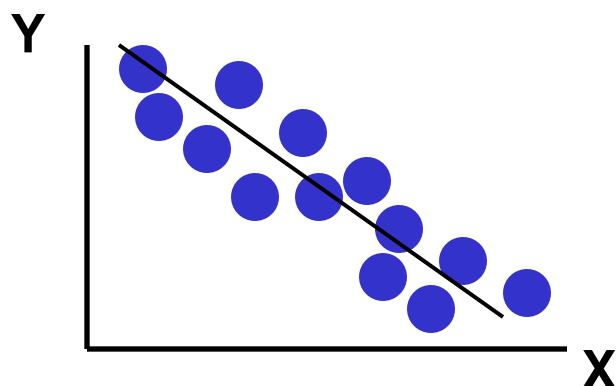
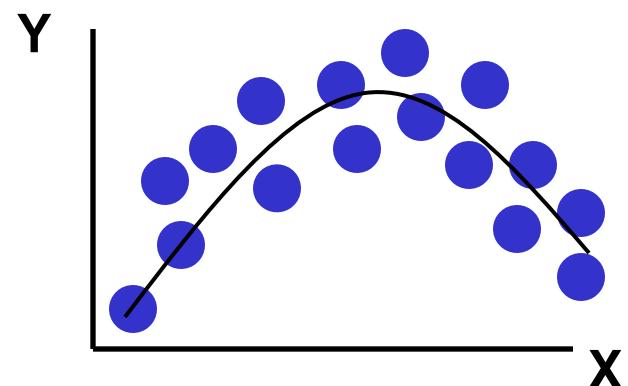
$$r = 0$$

Linear Correlation

Linear relationships

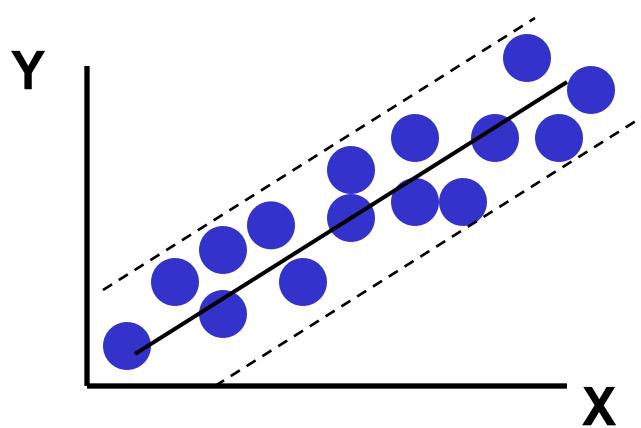


Curvilinear relationships

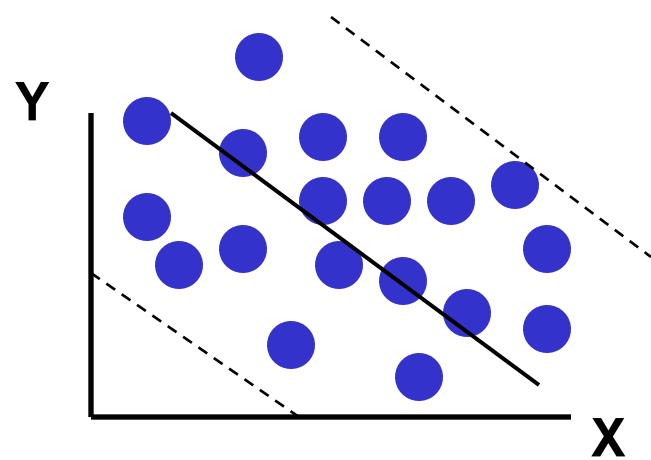
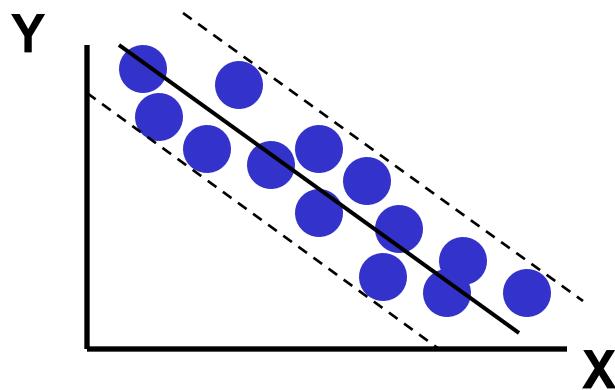
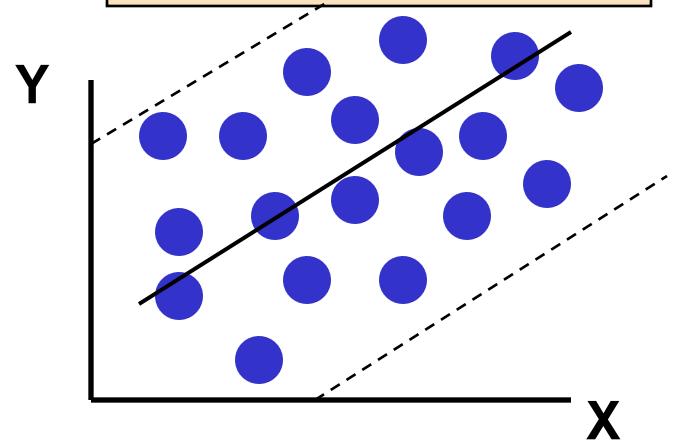


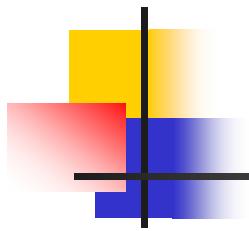
Linear Correlation

Strong relationships



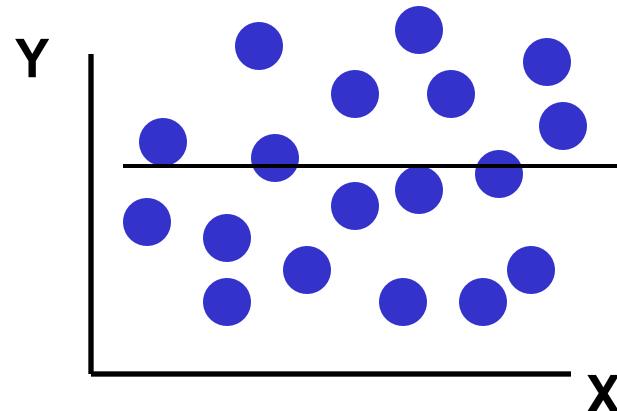
Weak relationships

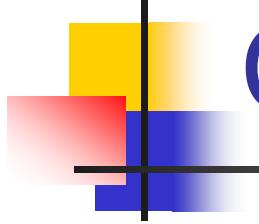




Linear Correlation

No relationship





Calculating by hand...

$$\hat{r} = \frac{\text{covariance}(x, y)}{\sqrt{\text{var } x} \sqrt{\text{var } y}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}}}$$

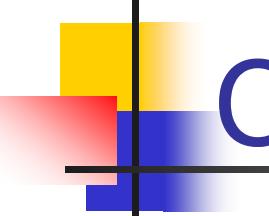
Simpler calculation formula...

$$\hat{r} = \frac{\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}}}}{\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} = \frac{SS_{xy}}{\sqrt{SS_x SS_y}}$$

Numerator of covariance

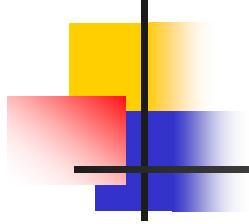
$$\hat{r} = \frac{SS_{xy}}{\sqrt{SS_x SS_y}}$$

Numerators of variance



Continuous outcome (means)

Outcome Variable	Are the observations independent or correlated?		Alternatives if the normality assumption is violated (and small sample size):
	Independent	correlated	
Continuous (e.g. pain scale, cognitive function)	Ttest: compares means between two independent groups ANOVA: compares means between more than two independent groups Pearson's correlation coefficient (linear correlation): shows linear correlation between two continuous variables	Paired ttest: compares means between two related groups (e.g., the same subjects before and after) Repeated-measures ANOVA: compares changes over time in the means of two or more groups (repeated measurements) Mixed models/GEE modeling: multivariate regression techniques to compare changes over time between two or more groups; gives rate of change over time	<u>Non-parametric statistics</u> Wilcoxon sign-rank test: non-parametric alternative to the paired ttest Wilcoxon sum-rank test (=Mann-Whitney U test): non-parametric alternative to the ttest Kruskal-Wallis test: non-parametric alternative to ANOVA Spearman rank correlation coefficient: non-parametric alternative to Pearson's correlation coefficient

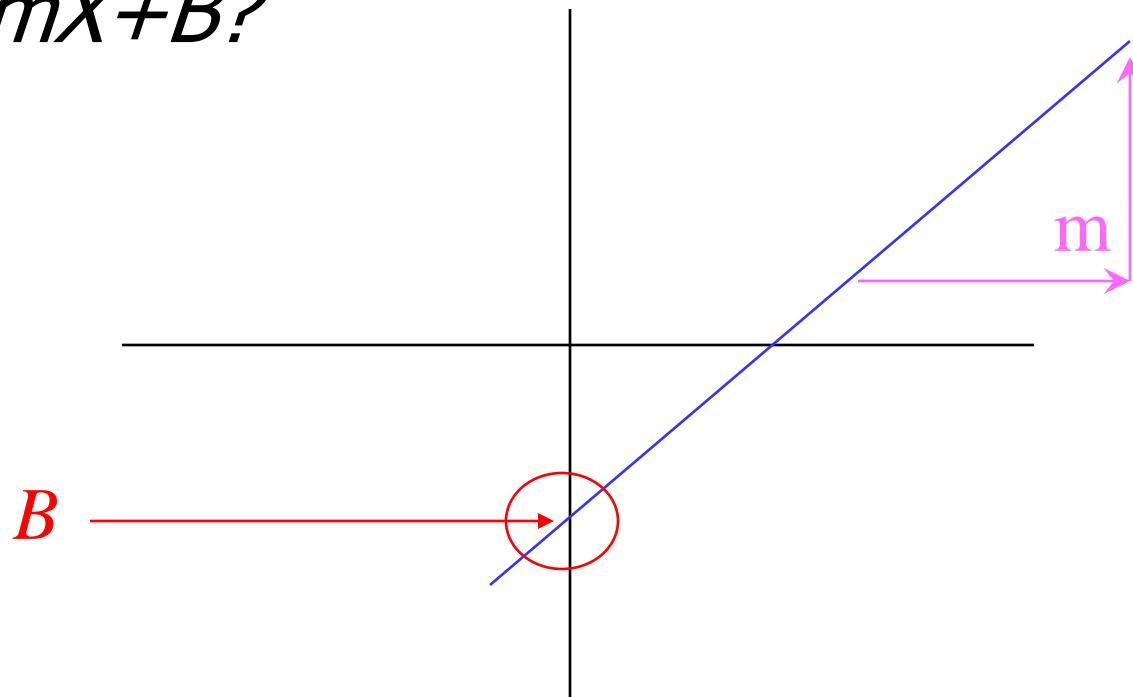


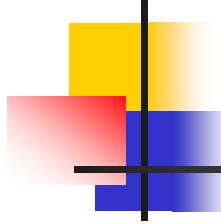
Linear regression

In correlation, the two variables are treated as equals. In regression, one variable is considered independent (=predictor) variable (X) and the other the dependent (=outcome) variable Y .

What is “Linear”?

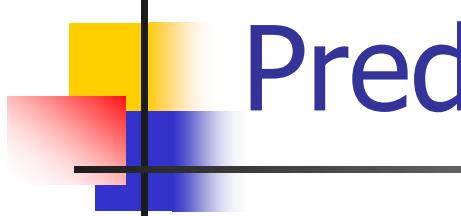
- Remember this:
- $Y=mX+B?$





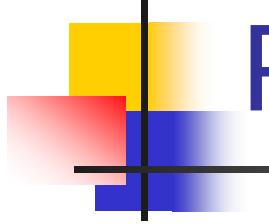
What's Slope?

A slope of 2 means that every 1-unit change in X yields a 2-unit change in Y.



Prediction

If you know something about X, this knowledge helps you predict something about Y. (Sound familiar?...sound like conditional probabilities?)



Regression equation...

Expected value of y at a given level of x =

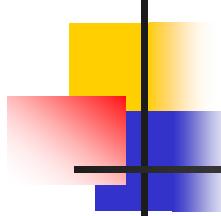
$$E(y_i / x_i) = \alpha + \beta x_i$$

Predicted value for an individual...

$$\hat{y}_i = \underbrace{\alpha + \beta * x_i}_{\text{Fixed – exactly on the line}} + \text{random error}_i$$

Fixed –
exactly
on the
line

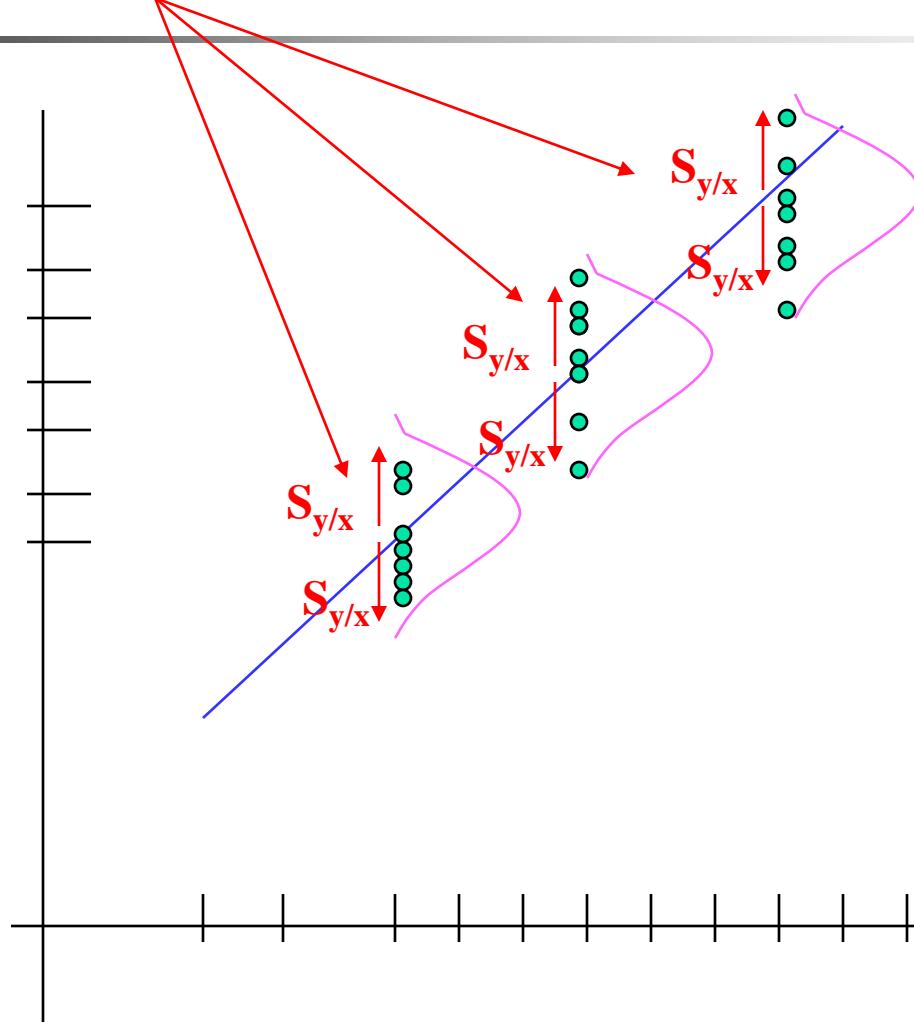
Follows a normal distribution



Assumptions

- Linear regression assumes that...
 - 1. The relationship between X and Y is linear
 - 2. Y is distributed normally at each value of X
 - 3. The variance of Y at every value of X is the same (homogeneity of variances)
 - 4. The observations are independent

The standard error of Y given X is the average variability around the regression line at any given value of X. It is assumed to be equal at all values of X.



Estimating the intercept and slope: least squares estimation

** Least Squares Estimation

A little calculus....

What are we trying to estimate? **β , the slope**, from

What's the constraint? We are trying to minimize the squared distance (hence the “least squares”) between the observations themselves and the predicted values , or (also called the “residuals”, or left-over unexplained variability)

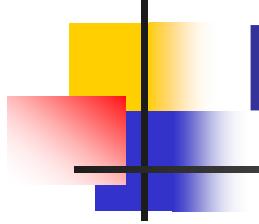
$$\text{Difference}_i = y_i - (\beta x + \alpha) \quad \text{Difference}_i^2 = (y_i - (\beta x + \alpha))^2$$

Find the β that gives the minimum sum of the squared differences. How do you maximize a function? Take the derivative; set it equal to zero; and solve. Typical max/min problem from calculus....

$$\frac{d}{d\beta} \sum_{i=1}^n (y_i - (\beta x_i + \alpha))^2 = 2 \left(\sum_{i=1}^n (y_i - \beta x_i - \alpha)(-x_i) \right)$$

$$2 \left(\sum_{i=1}^n (-y_i x_i + \beta x_i^2 + \alpha x_i) \right) = 0 \dots$$

From here takes a little math trickery to solve for β ...



Resulting formulas...

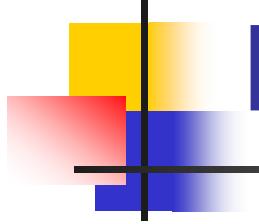
Slope (beta coefficient) =

$$\hat{\beta} = \frac{Cov(x, y)}{Var(x)}$$

Intercept =

$$\text{Calculate: } \hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

Regression line always goes through the point: (\bar{x}, \bar{y})



Relationship with correlation

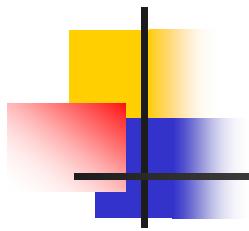
$$\hat{r} = \hat{\beta} \frac{SD_x}{SD_y}$$

In correlation, the two variables are treated as equals. In regression, one variable is considered independent (=predictor) variable (X) and the other the dependent (=outcome) variable Y .

Formula for the standard error of beta (you will not have to calculate by hand!):

$$s_{\hat{\beta}} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}} = \sqrt{\frac{s_{y/x}^2}{SS_x}}$$

where $SS_x = \sum_{i=1}^n (x_i - \bar{x})^2$
and $\hat{y}_i = \hat{\alpha} + \hat{\beta}x_i$



Multiple Linear Regression

- More than one predictor...

$$E(y) = \alpha + \beta_1 * X + \beta_2 * W + \beta_3 * Z \dots$$

Each regression coefficient is the amount of change in the outcome variable that would be expected per one-unit change of the predictor, if all other variables in the model were held constant.

Logistic Regression



Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University

Regression equation...



Expected value of y at a given level of x =

$$E(y_i / x_i) = \alpha + \beta x_i$$

Multiple Linear Regression

- More than one predictor...

$$E(y) = \alpha + \beta_1 * X + \beta_2 * W + \beta_3 * Z \dots$$

Each regression coefficient is the amount of change in the outcome variable that would be expected per one-unit change of the predictor, if all other variables in the model were held constant.

What is Logistic Regression?

- Form of regression that allows the prediction of discrete variables by a mix of continuous and discrete predictors.
- Addresses the same questions that discriminant function analysis and multiple regression do but with no distributional assumptions on the predictors (the predictors do not have to be normally distributed, linearly related or have equal variance in each group)

What is Logistic Regression?

- Logistic regression is often used because the relationship between the DV (a discrete variable) and a predictor is non-linear
 - Example from the text: the probability of heart disease changes very little with a ten-point difference among people with low-blood pressure, but a ten point change can mean a drastic change in the probability of heart disease in people with high blood-pressure.

What is Logistic Regression?

- To predict an outcome variable that is categorical from one or more categorical or continuous predictor variables.
- Used because having a categorical outcome variable violates the assumption of linearity in normal regression.
- Does not assume a linear relationship between DV and IV

What is Logistic Regression?



- Logistic regression combines the independent variables to estimate the probability that a particular event will occur, i.e. a subject will be a member of one of the groups defined by the two categories dependent variable.

Types of logistic regression

- **BINARY LOGISTIC REGRESSION**
 - It is used when the dependent variable is dichotomous.
- **MULTINOMIAL LOGISTIC REGRESSION**
 - It is used when the dependent or outcomes variable has more than two categories.

Binary logistic regression expression

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_k X_k + E$$

BINAR
Y

Y = Dependent Variables

B₀ = Constant

B₁ = Coefficient of variable X₁

X₁ = Independent Variables

E = Error Term

Logistic Regression

In logistic regression the outcome variable is binary, and the purpose of the analysis is to assess the effects of multiple explanatory variables, which can be numeric and/or categorical, on the outcome variable.

When and Why Binary Logistic Regression?

- When the dependent variable is non parametric and we don't have homoscedasticity (variance of DV and IV not equal).
- Used when the dependent variable has only two levels. (Yes/no, male/female, taken/not taken)
- If multivariate normality is suspected.
- If we don't have linearity.

Who uses it?

- Binary Logistic Regression can be used in the following situations.
 - A catalog company wants to increase the proportion of mailings that result in sales.
 - A doctor wants to accurately diagnose a possibly cancerous tumor.
 - A loan officer wants to know whether the next customer is likely to default.
- Using the Binary Logistic Regression procedure, the catalog company can send mailings to the people who are most likely to respond, the doctor can determine whether the tumor is more likely to be benign or malignant, and the loan officer can assess the risk of extending credit to a particular customer.

Sample Size

- 
- Very small samples have so much sampling errors.
 - Very large sample size decreases the chances of errors.
 - Logistic requires larger sample size than multiple regression.

Questions

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
-
- What is the relative importance of each predictor?
 - How does each variable affect the outcome?
 - Does a predictor make the solution better or worse or have no effect?

Questions

- 
- 
- Are there interactions among predictors?
 - Can parameters be accurately predicted?
 - How good is the model at classifying cases for which the outcome is known ?

Assumptions



- No assumptions about the distributions of the predictor variables.
- Predictors do not have to be normally distributed
- Does not have to be linearly related.
- Does not have to have equal variance within each group.

Assumptions

- The only “real” limitation on logistic regression is that the outcome must be discrete.

Assumptions

- If the distributional assumptions are met than discriminant function analysis may be more powerful, although it has been shown to overestimate the association using discrete predictors.
- If the outcome is continuous then multiple regression is more powerful given that the assumptions are met

Assumptions

- Ratio of cases to variables – using discrete variables requires that there are enough responses in every given category
 - If there are too many cells with no responses parameter estimates and standard errors will likely blow up
 - Also can make groups perfectly separable (e.g. multicollinear) which will make maximum likelihood estimation impossible.

Assumptions

- Linearity in the logit – the regression equation should have a linear relationship with the logit form of the DV.
- There is no assumption about the predictors being linearly related to each other.

How is relationship formulated?

For linear simplest equation is :

$$y = a + bx + e_i$$

y is the outcome; a is the intercept;

b is the slope related to x the explanatory variable and;

e is the error term or random 'noise'

Can we fit y as a probability range 0 to 1?



$$y = a + bx + e_i$$

Not quite!

y as continuous - any value from $-\infty$ to $+\infty$

Outcome is a probability of event, Π (or p) on scale 0 - 1

Certain transformations of p can give the required scale

Probit is a normal transformation of p but not easy to interpret results

The logit transformation works!



We can now fit p as a probability range 0 to 1

And y in range $-\infty$ to $+\infty$

$$y = \text{logit}(p) = a + bx + e_i$$

$$\log\left(\frac{p}{1-p}\right) = a + bx + e_i$$

Logistic Regression Model


$$\log\left(\frac{P}{1 - P}\right) = a + bx + e_i$$

This has very useful properties

The term $p/(1-p)$ is called the 'Odds' of an event

Note: not the same as the probability of an event p

If x is binary coded 0/1 then -

$$\exp(b) = \text{ODDS RATIO}$$

for the outcome in those coded 1 relative to code 0

e.g. Odds of death in men (1) vs. women (0)

Background

- Odds – like probability. Odds are usually written as “5 to 1 odds” which is equivalent to 1 out of five or .20 probability or 20% chance, etc.
 - The problem with probabilities is that they are non-linear
 - Going from .10 to .20 doubles the probability, but going from .80 to .90 barely increases the probability.

Background

- Odds ratio – the ratio of the odds over 1 – the odds. The probability of winning over the probability of losing. 5 to 1 odds equates to an odds ratio of $.20/.80 = .25$.

Background

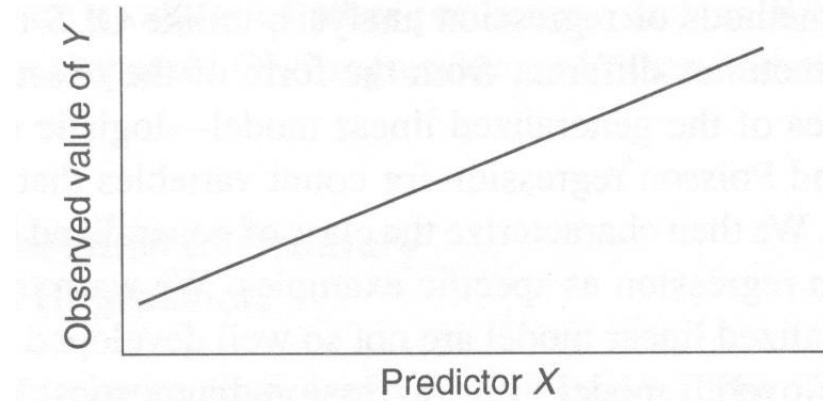
- Logit – this is the natural log of an odds ratio; often called a log odds even though it really is a log odds ratio. The logit scale is linear and functions much like a z-score scale.

The ogive function

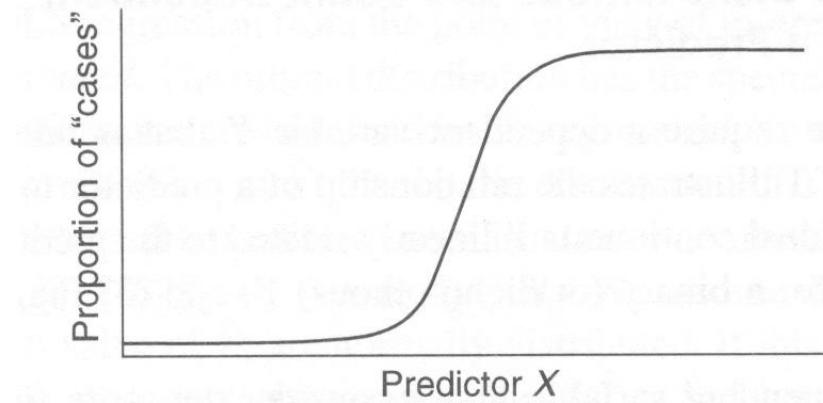
- An ogive function is a curved s-shaped function and the most common is the logistic function which looks like:

The logistic function

(A) For a continuous outcome variable Y , the numerical value of Y at each value of X .



(B) For a binary outcome variable, the proportion of individuals who are “cases” (exhibit a particular outcome property) at each value of X .



The logistic function


$$\widehat{Y}_i = \frac{e^u}{1 + e^u}$$

- Where \widehat{Y}_i is the estimated probability that the i th case is in a category and u is the regular linear regression equation:

$$u = A + B_1 X_1 + B_2 X_2 + \cdots + B_K X_K$$

The logistic function



$$\hat{\pi}_i = \frac{e^{b_0 + b_1 X_1}}{1 + e^{b_0 + b_1 X_1}}$$

The logistic function

- Change in probability is not constant (linear) with constant changes in X
- This means that the probability of a success ($Y = 1$) given the predictor variable (X) is a non-linear function, specifically a logistic function

The logistic function

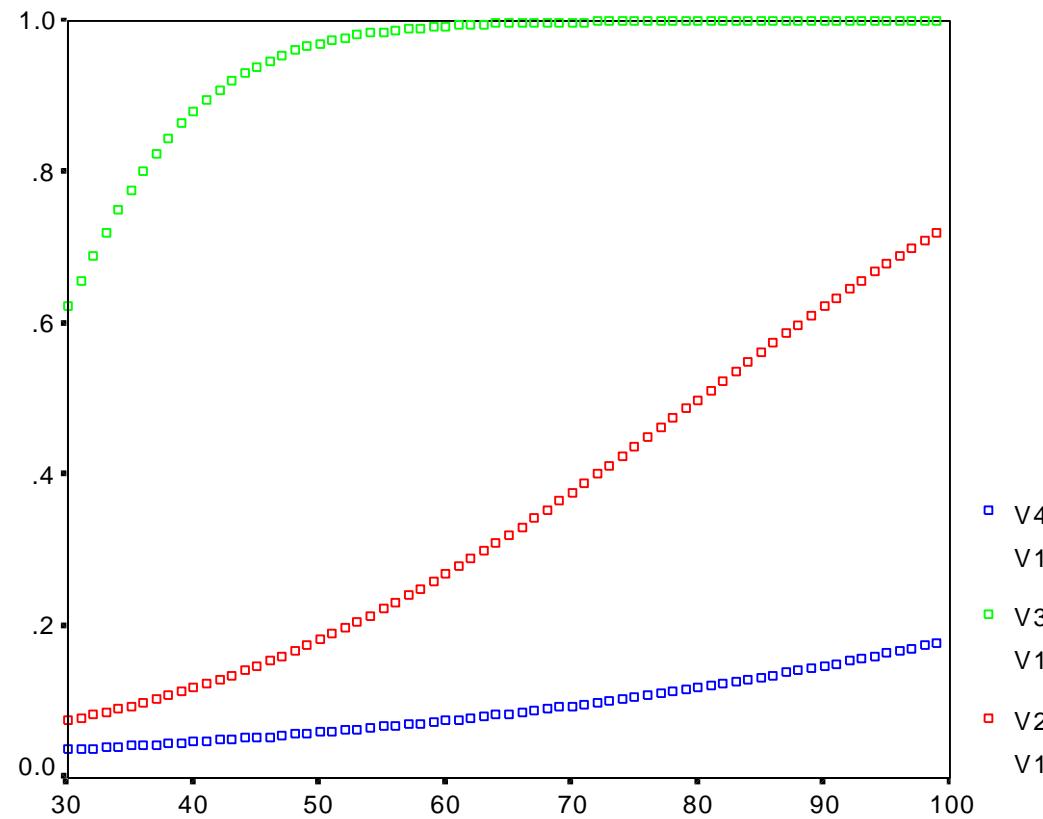
- It is not obvious how the regression coefficients for X are related to changes in the dependent variable (Y) when the model is written this way
- Change in Y(in probability units)|X depends on value of X. Look at S-shaped function

The logistic function

- The values in the regression equation b_0 and b_1 take on slightly different meanings.
 - $b_0 \leftarrow$ The regression constant (moves curve left and right)
 - $b_1 \leftarrow$ The regression slope (steepness of curve)
 - $\frac{-b_0}{b_1} \leftarrow$ The threshold, where probability of success = .50

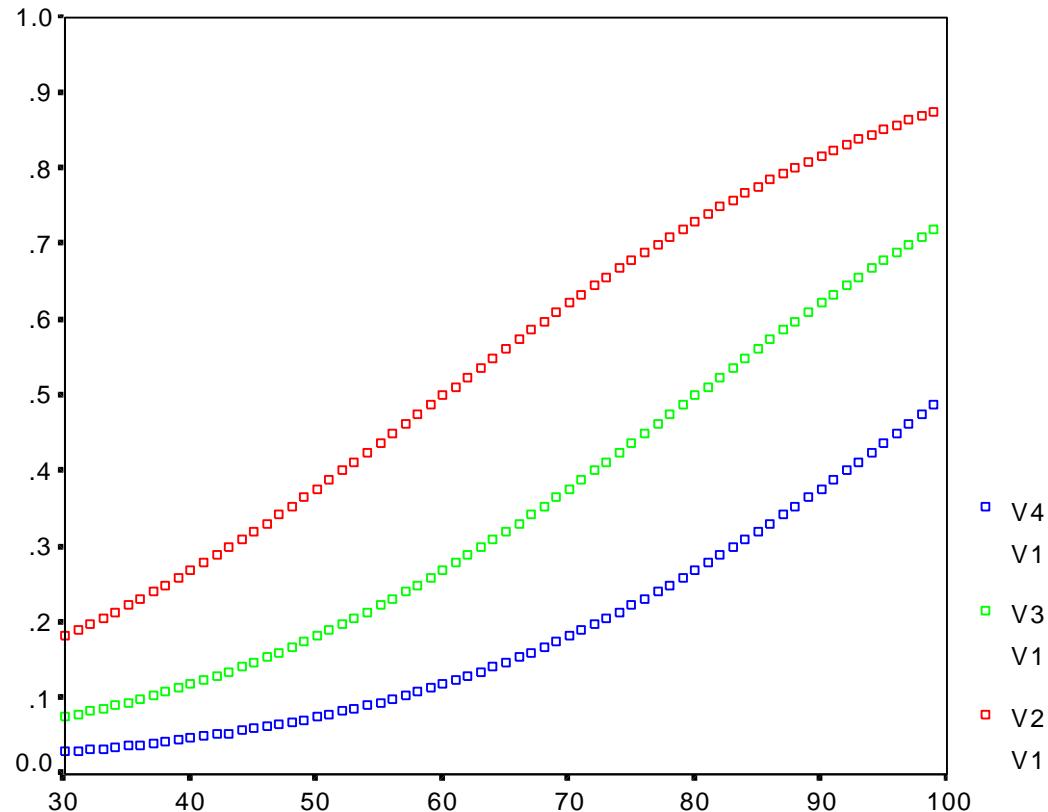
Logistic Function

- Constant regression constant different slopes
 - v2: $b_0 = -4.00$
 $b_1 = 0.05$ (middle)
 - v3: $b_0 = -4.00$
 $b_1 = 0.15$ (top)
 - v4: $b_0 = -4.00$
 $b_1 = 0.025$ (bottom)



Logistic Function

- Constant slopes with different regression constants
 - v2: $b_0 = -3.00$
 $b_1 = 0.05$ (top)
 - v3: $b_0 = -4.00$
 $b_1 = 0.05$ (middle)
 - v4: $b_0 = -5.00$
 $b_1 = 0.05$ (bottom)



The Logit

- By algebraic manipulation, the logistic regression equation can be written in terms of an odds ratio for success:

$$\left[\frac{P(Y=1|X_i)}{(1-P(Y=1|X_i))} \right] = \left[\frac{\hat{\pi}}{(1-\hat{\pi})} \right] = \exp(b_0 + b_1 X_{1i})$$

The Logit

- Odds ratios range from 0 to positive infinity
- Odds ratio: P/Q is an odds ratio; less than 1 = less than .50 probability, greater than 1 means greater than .50 probability

The Logit

- Finally, taking the natural log of both sides, we can write the equation in terms of logits (log-odds):

$$\ln \left[\frac{P(Y=1|X)}{(1-P(Y=1|X))} \right] = \ln \left[\frac{\hat{\pi}}{(1-\hat{\pi})} \right] = b_0 + b_1 X_1$$

For a single predictor

The Logit


$$\ln \left[\frac{\hat{\pi}}{(1 - \hat{\pi})} \right] = b_0 + b_1 X_1 + b_2 X_2 \dots + b_k X_k$$

- For multiple predictors

PRESENTATION ON NAÏVE BAYESIAN CLASSIFICATION

Presented By:

Ashraf Uddin

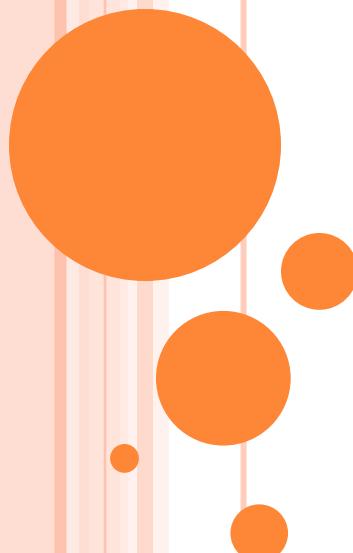
Sujit Singh

Chetanya Pratap Singh

South Asian University

(Master of Computer Application)

<http://ashrafsau.blogspot.in/>



OUTLINE

- Introduction to Bayesian Classification
 - Bayes Theorem
 - Naïve Bayes Classifier
 - Classification Example
- Text Classification – an Application
- Comparison with other classifiers
 - Advantages and disadvantages
 - Conclusions



CLASSIFICATION

- Classification:

- predicts categorical class labels
- classifies data (constructs a model) based on the training set and the values (**class labels**) in a classifying attribute and uses it in classifying new data

- Typical Applications

- credit approval
- target marketing
- medical diagnosis
- treatment effectiveness analysis



A TWO STEP PROCESS

- Model construction: describing a set of predetermined classes
 - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
 - The set of tuples used for model construction: training set
 - The model is represented as classification rules, decision trees, or mathematical formulae
- Model usage: for classifying future or unknown objects
 - Estimate accuracy of the model
 - The known label of test sample is compared with the classified result from the model
 - Accuracy rate is the percentage of test set samples that are correctly classified by the model
 - Test set is independent of training set, otherwise over-fitting will occur

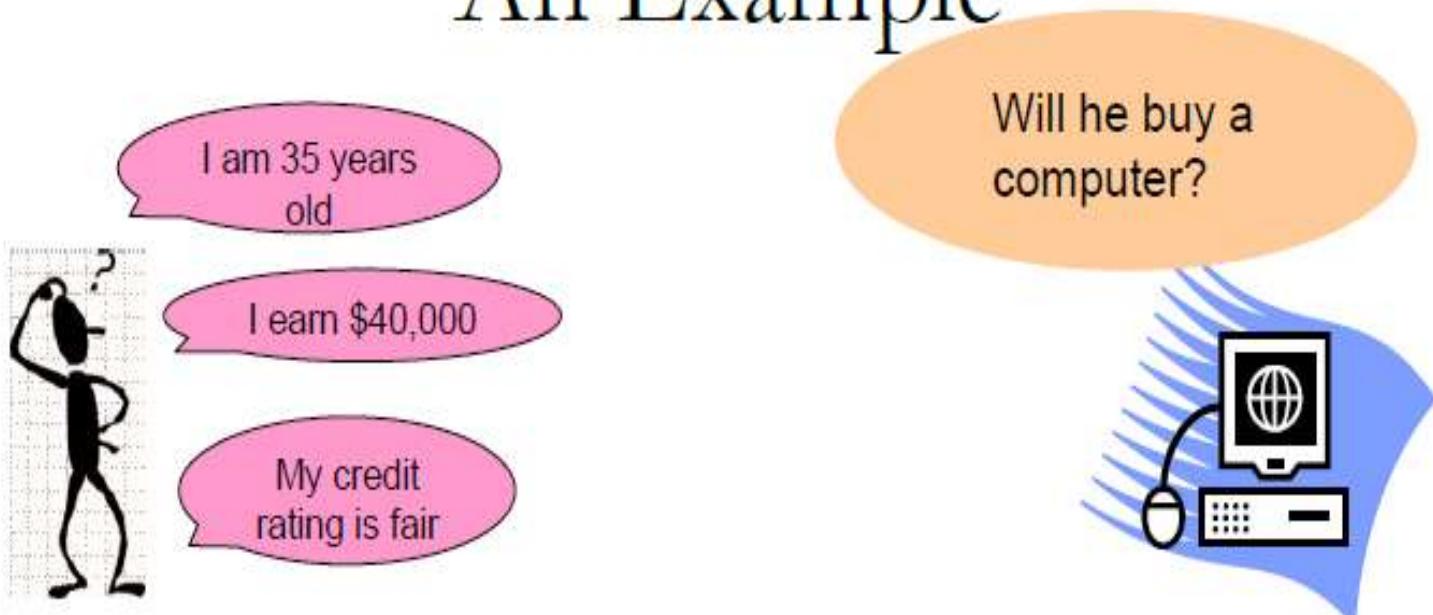
INTRODUCTION TO BAYESIAN CLASSIFICATION

- What is it ?

- Statistical method for classification.
- Supervised Learning Method.
- Assumes an underlying probabilistic model, the Bayes theorem.
- Can solve problems involving both categorical and continuous valued attributes.
- Named after *Thomas Bayes, who proposed the Bayes Theorem.*



An Example



- X : 35 years old customer with an income of \$40,000 and fair credit rating.
- H : Hypothesis that the customer will buy a computer.

THE BAYES THEOREM

- The Bayes Theorem:
 - $P(H|X) = P(X|H) P(H) / P(X)$
- $P(H|X)$: Probability that the customer will buy a computer given that we know his age, credit rating and income. (Posterior Probability of H)
- $P(H)$: Probability that the customer will buy a computer regardless of age, credit rating, income (Prior Probability of H)
- $P(X|H)$: Probability that the customer is 35 yrs old, have fair credit rating and earns \$40,000, given that he has bought our computer (Posterior Probability of X)
- $P(X)$: Probability that a person from our set of customers is 35 yrs old, have fair credit rating and earns \$40,000. (Prior Probability of X)

BAYESIAN CLASSIFIER

- ▶ D : Set of tuples
 - Each Tuple is an ‘n’ dimensional attribute vector
 - $X : (x_1, x_2, x_3, \dots, x_n)$
 - where x_i is the value of attribute A_i
- ▶ Let there are ‘m’ Classes : $C_1, C_2, C_3, \dots, C_m$
- ▶ Bayesian classifier predicts X belongs to Class C_i iff
 - $P(C_i|X) > P(C_j|X)$ for $1 \leq j \leq m, j \neq i$
- ▶ Maximum Posteriori Hypothesis
 - $$P(C_i|X) = \frac{P(X|C_i) P(C_i)}{P(X)}$$
 - Maximize $P(X|C_i) P(C_i)$ as $P(X)$ is constant

NAÏVE BAYESIAN CLASSIFIER...

- With many attributes, it is computationally expensive to evaluate $P(X|C_i)$
- Naïve Assumption of “class conditional independence”

$$\begin{aligned} P(X | C_i) &= P(x_1, x_2, \dots, x_n | C_i) \\ &= P(x_1 | C_i) * P(x_2 | C_i) * \dots * P(x_n | C_i) \\ &= \prod_{k=1}^n P(x_k | C_i) \end{aligned}$$



NAÏVE BAYESIAN CLASSIFIER...

To compute, $P(x_k|C_i)$

- A_k is categorical:

the number of tuples of class C_i in D having the value x_k for A_k

$$P(x_k|C_i) = \frac{\text{_____}}{\text{_____}}$$

the number of tuples of class C_i in D .

- A_k is continuous:

A continuous-valued attribute is typically assumed to have a Gaussian distribution with a mean μ and standard deviation σ

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$



“ZERO” PROBLEM

- What if there is a class, C_i , and X has an attribute value, x_k , such that none of the samples in C_i has that attribute value?
- In that case $P(x_k|C_i) = 0$, which results in $P(X|C_i) = 0$ even though $P(x_k|C_i)$ for all the other attributes in X may be large.

NUMERICAL UNDERFLOW

- When $p(x|Y)$ is often a very small number: the probability of observing any particular high-dimensional vector is small.
- This can lead to numerical under flow.

$$\begin{aligned} P(X | C_i) &= P(x_1, x_2, \dots, x_n | C_i) \\ &= P(x_1 | C_i) * P(x_2 | C_i) * \dots * P(x_n | C_i) \end{aligned}$$



LOG SUM-EXP TRICK

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k | c)].$$

BASIC ASSUMPTION

- The Naïve Bayes assumption is that all the features are conditionally independent given the class label.

$$P(X | C_i) = P(x_1, x_2, \dots, x_n | C_i)$$

$$= P(x_1 | C_i) * P(x_2 | C_i) * \dots * P(x_n | C_i)$$

- Even though this is usually false (some features are usually dependent)



EXAMPLE: CLASS-LABELED TRAINING TUPLES FROM THE *CUSTOMER DATABASE*

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

EXAMPLE...

- ▶ $X = (\text{Age} = \text{'}<=30\text{'}, \text{Income} = \text{medium}, \text{Student} = \text{yes}, \text{Credit_rating} = \text{fair})$

- ▶ $P(C1) = P(\text{Buys_computer} = \text{yes}) = 9/14 = 0.643$

- ▶ $P(C2) = P(\text{Buys_computer} = \text{no}) = 5/14 = 0.357$

- ▶
$$P(\text{Age} = \text{'}<=30\text{'} | \text{Buys_computer} = \text{yes}) = \frac{\text{number of tuples with Buys_computer= yes and Age } <= 30}{\text{number of tuples with Buys_computer = yes}}$$

- ▶ $P(\text{Age} = \text{'}<=30\text{'} | \text{Buys_computer} = \text{yes}) = 2/9 = 0.222$

Similarly,

- ▶ $P(\text{Age} = \text{'}<=30\text{'} | \text{Buys_computer} = \text{no}) = 3/5 = 0.600$

- ▶ $P(\text{Income} = \text{medium} | \text{Buys_computer} = \text{yes}) = 4/9 = 0.444$

- ▶ $P(\text{Income} = \text{medium} | \text{Buys_computer} = \text{no}) = 2/5 = 0.400$

- ▶ $P(\text{Student} = \text{yes} | \text{Buys_computer} = \text{yes}) = 6/9 = 0.667$

- ▶ $P(\text{Student} = \text{yes} | \text{Buys_computer} = \text{no}) = 1/5 = 0.200$

- ▶ $P(\text{Credit_rating} = \text{fair} | \text{Buys_computer} = \text{yes}) = 6/9 = 0.667$

- ▶ $P(\text{Credit_rating} = \text{fair} | \text{Buys_computer} = \text{no}) = 2/5 = 0.400$

EXAMPLE...

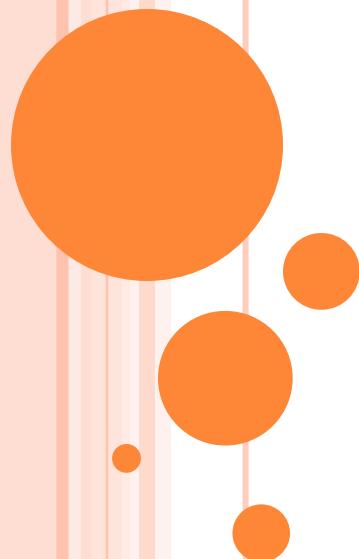
- ▶ $P(X | \text{Buys a computer} = \text{yes})$
= $P(\text{Age}='<=30' | \text{buys_computer} = \text{yes}) * P(\text{Income}=\text{medium} | \text{buys_computer} = \text{yes}) * P(\text{Student}=\text{yes} | \text{buys_computer} = \text{yes}) * P(\text{Credit rating}=\text{fair} | \text{buys_computer} = \text{yes})$
= $0.222 * 0.444 * 0.667 * 0.667 = 0.044$
- ▶ $P(X | \text{Buys a computer} = \text{No})$
= $0.600 * 0.400 * 0.200 * 0.400 = 0.019$
- ▶ Find class C_i that Maximizes $P(X|C_i) * P(C_i)$
→ $P(X | \text{Buys a computer} = \text{yes}) * P(\text{Buys_computer} = \text{yes}) = 0.028$
→ $P(X | \text{Buys a computer} = \text{No}) * P(\text{Buys_computer} = \text{no}) = 0.007$
- ▶ Prediction : Buys a computer for Tuple X

USES OF NAÏVE BAYES CLASSIFICATION

- Text Classification
- Spam Filtering
- Hybrid Recommender System
 - Recommender Systems apply machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource
- Online Application
 - Simple Emotion Modeling



TEXT CLASSIFICATION – AN APPLICATION OF NAIVE BAYES CLASSIFIER



WHY TEXT CLASSIFICATION?

- Learning which articles are of interest
- Classify web pages by topic
- Information extraction
- Internet filters



EXAMPLES OF TEXT CLASSIFICATION

- CLASSES=BINARY

- “spam” / “not spam”

- CLASSES =TOPICS

- “finance” / “sports” / “politics”

- CLASSES =OPINION

- “like” / “hate” / “neutral”

- CLASSES =TOPICS

- “AI” / “Theory” / “Graphics”

- CLASSES =AUTHOR

- “Shakespeare” / “Marlowe” / “Ben Jonson”



EXAMPLES OF TEXT CLASSIFICATION

- Classify news stories as world, business, SciTech, Sports ,Health etc
- Classify email as spam / not spam
- Classify business names by industry
- Classify email to tech stuff as Mac, windows etc
- Classify pdf files as research , other
- Classify movie reviews as favorable, unfavorable, neutral
- Classify documents
- Classify technical papers as Interesting, Uninteresting
- Classify Jokes as Funny, Not Funny
- Classify web sites of companies by Standard Industrial Classification (SIC)



NAÏVE BAYES APPROACH

- Build the Vocabulary as the list of all distinct words that appear in all the documents of the training set.
- Remove stop words and markings
- The words in the vocabulary become the attributes, assuming that classification is independent of the positions of the words
- Each document in the training set becomes a record with frequencies for each word in the Vocabulary.
- Train the classifier based on the training data set, by computing the prior probabilities for each class and attributes.
- Evaluate the results on Test data



REPRESENTING TEXT: A LIST OF WORDS

Agarose gel electrophoresis separates nucleic acids based on their size. The gel has a negative charge, so the DNA migrates toward the positive terminal. The size of the DNA fragments is determined by the length of the gel and the voltage applied. The larger the fragment, the slower it moves. The smaller the fragment, the faster it moves. The bands are visualized by staining the gel with ethidium bromide.

f() = y
(argentine, 1986, 1987, grain, oilseed,
registrations, buenos, aires, feb, 26,
argentine, grain, board, figured, show, crop,
registrations, of, grains, oilseeds, and, their
products, to, february, 11, in, ...)

- Common Refinements: Remove Stop Words, Symbols

TEXT CLASSIFICATION ALGORITHM: NAÏVE BAYES

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B'}$$

- T_{ct} – Number of particular word in particular class
- $T_{ct'}$ – Number of total words in particular class
- B' – Number of distinct words in all class



EXAMPLE

► Table 13.1 Data for parameter estimation examples.

	docID	words in document	in $c = China$?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

$$\hat{P}(\text{Chinese}|c) = (5+1)/(8+6) = 6/14 = 3/7$$

$$\hat{P}(\text{Tokyo}|c) = \hat{P}(\text{Japan}|c) = (0+1)/(8+6) = 1/14$$

$$\hat{P}(\text{Chinese}|\bar{c}) = (1+1)/(3+6) = 2/9$$

$$\hat{P}(\text{Tokyo}|\bar{c}) = \hat{P}(\text{Japan}|\bar{c}) = (1+1)/(3+6) = 2/9$$

EXAMPLE CONT...

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

$$\hat{P}(c|d_5) \propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003.$$

$$\hat{P}(\bar{c}|d_5) \propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001.$$

- Probability of Yes Class is more than that of No Class, Hence this document will go into Yes Class.

Advantages and Disadvantages of Naïve Bayes

Advantages :

- Easy to implement
- Requires a small amount of training data to estimate the parameters
- Good results obtained in most of the cases

Disadvantages:

- Assumption: class conditional independence, therefore loss of accuracy
- Practically, dependencies exist among variables

E.g., hospitals: patients: Profile: age, family history, etc.

Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.

- Dependencies among these cannot be modelled by Naïve Bayesian Classifier



An extension of Naive Bayes for delivering robust classifications

- Naive assumption (statistical independent. of the features given the class)
- NBC computes a single posterior distribution.
- However, the most probable class might depend on the chosen prior, especially on small data sets.
- Prior-dependent classifications might be weak.

Solution via set of probabilities:

- Robust Bayes Classifier (Ramoni and Sebastiani, 2001)
- Naive Credal Classifier (Zaffalon, 2001)

Relevant Issues

- Violation of Independence Assumption
- Zero conditional probability Problem



VIOLATION OF INDEPENDENCE ASSUMPTION

- Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered “naive.”

IMPROVEMENT

- Bayesian belief network are graphical models, which unlike naïve Bayesian classifiers, allow the representation of dependencies among subsets of attributes.
- Bayesian belief networks can also be used for classification.



ZERO CONDITIONAL PROBABILITY PROBLEM

- ✓ If a given class and feature value never occur together in the training set then the frequency-based probability estimate will be zero.
- ✓ This is problematic since it will wipe out all information in the other probabilities when they are multiplied.
- ✓ It is therefore often desirable to incorporate a small-sample correction in all probability estimates such that no probability is ever set to be exactly zero.



CORRECTION

- To eliminate zeros, we use add-one or Laplace smoothing, which simply adds one to each count

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}},$$

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B},$$

EXAMPLE

- Suppose that for the class buys computer D (yes) in some training database, D, containing 1000 tuples.
 - we have 0 tuples with income D low,
 - 990 tuples with income D medium, and
 - 10 tuples with income D high.
- The probabilities of these events, without the Laplacian correction, are 0, 0.990 (from 990/1000), and 0.010 (from 10/1000), respectively.
- Using the Laplacian correction for the three quantities, we pretend that we have 1 more tuple for each income-value pair. In this way, we instead obtain the following probabilities :

$$\frac{1}{1003} = 0.001, \frac{991}{1003} = 0.988, \text{ and } \frac{11}{1003} = 0.011,$$

respectively. The “corrected” probability estimates are close to their “uncorrected” counterparts, yet the zero probability value is avoided.

Remarks on the Naive Bayesian Classifier

- Studies comparing classification algorithms have found that the naive Bayesian classifier to be comparable in performance with decision tree and selected neural network classifiers.
- Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.



Conclusions

- ✓ The naive Bayes model is tremendously appealing because of its simplicity, elegance, and robustness.
- ✓ It is one of the oldest formal classification algorithms, and yet even in its simplest form it is often surprisingly effective.
- ✓ It is widely used in areas such as text classification and spam filtering.
- ✓ A large number of modifications have been introduced, by the statistical, data mining, machine learning, and pattern recognition communities, in an attempt to make it more flexible.
- ✓ but some one has to recognize that such modifications are necessarily complications, which detract from its basic simplicity.



REFERENCES

- http://en.wikipedia.org/wiki/Naive_Bayes_classifier
- <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/mlbook/ch6.pdf>
- Data Mining: Concepts and Techniques, 3rd Edition, Han & Kamber & Pei ISBN: 9780123814791



Machine Learning

Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University

Estimation using Maximum Likelihood

- Let D be a set of data generated from some distribution parameterized by θ .
 - We want to *estimate* the unknown parameter θ .
 - What we can do?
-
- Essentially, we want to find a most likely value of θ given D , that is $\text{argmax } P(\theta|D)$.

Estimation using Maximum Likelihood

- According to Bayes Rule, we have

$$P(\theta|D) = P(D|\theta)P(\theta) / P(D)$$

and the terms have the following meanings:

- $P(\theta|D)$: Posterior
- $P(D|\theta)$: Likelihood
- $P(\theta)$: Prior
- $P(D)$: Evidence

Estimation using Maximum Likelihood

- An easy way out is to use the MLE method.
- We want to find a θ the *best explains* the data.
- That is, we maximize $P(D|\theta)$.
- Denote such a value as $\hat{\theta}$.
- We have

$$\hat{\theta} = \operatorname{argmax}_{\theta} P(D|\theta) = \operatorname{argmax}_{\theta} P(X_1, X_2, X_3, \dots, X_N | \theta)$$

Estimation using Maximum Likelihood

- We have

$$\hat{\theta} = \operatorname{argmax}_{\theta} P(D|\theta) = \operatorname{argmax}_{\theta} P(X_1, X_2, X_3, \dots, X_N | \theta)$$

- Note that the above P is a joint distribution over the data.
- We usually assume the observations are *independent*.
- Thus, we have

$$P(X_1, \dots, X_N | \theta) = \prod_{i=1}^N P(x_i | \theta)$$

- We usually use logarithm to simplify the computation, as logarithm is monotonically increasing. Thus, we write:

$$L(D|\theta) = \sum_{i=1}^N \log P(x_i | \theta)$$

Estimation using Maximum Likelihood

- Finally, we seek for the ML solution:

$$\hat{\theta} = \operatorname{argmax}_{\theta} L(D|\theta)$$

- If we know the distribution P , we can usually solve the above by setting derivative of θ to 0 and solve for θ , that is,

$$\frac{\partial L}{\partial \theta} = 0$$

Estimation using Maximum Likelihood

- In Bayesian statistics, a maximum a posteriori probability (MAP) estimate is an estimate of an unknown quantity, that equals the mode of the posterior distribution.
- Maximum A Posteriori (MAP):
In MAP, we maximize $P(D|\theta)$ directly.
- Note that, due to the evidence (data) D is constant, and thus can be omitted in argmax.
- At this step, we notice that the only difference between MLE and MAP is the prior term $P(\theta)$.
- The extra prior term has the effect that we are essentially ‘pulling’ the θ distribution towards prior value. This makes sense as we are putting our domain knowledge as *prior* and intuitively the estimation is biased towards the *prior* value.



Learning from Hidden Variables and Missing Data

**Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University**

Course Outline

- Missing data and Hidden variables
- Why do we want hidden variables?
- EM algorithm
- Bayesian networks with hidden variables

Hidden Variables and Missing Data

- Latent or hidden variables in the model are never observed
 - We may or may not be interested in their values, but their existence is crucial to the model
- Some observations in a particular sample may be missing
 - Missing information on surveys or medical records (quite common)
 - We may need to model how the variables are missing

Missing Data

- Data can be missing from the model in many different ways
 - Missing completely at random: the probability that a data item is missing is independent of the observed data and the other missing data
 - Missing at random: the probability that a data item is missing can depend on the observed data
 - Missing not at random: the probability that a data item is missing can depend on the observed data and the other missing data

Handling Missing Data

- Discard all incomplete observations
 - Can introduce bias
- Imputation: actual values are substituted for missing values so that all of the data is fully observed
 - E.g., find the most probable assignments for the missing data and substitute them in (not possible if the model is unknown)
 - Use the sample mean/mode
- Explicitly model the missing data
 - For example, could expand the state space
 - The most sensible solution, but may be non-trivial if we don't know how/why the data is missing

Learning probability distribution

Basic learning settings:

- A set of random variables $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$
- **A model of the distribution** over variables in X with parameters Θ
- **Data** $D = \{D_1, D_2, \dots, D_N\}$
s.t. $D_i = (x_1^i, x_2^i, \dots, x_n^i)$

Objective: find parameters $\hat{\Theta}$ that describe the data

Assumptions considered so far:

- Known parameterizations
- No hidden variables
- No-missing values

Hidden variables

Modeling assumption:

Variables $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ are related through hidden variables

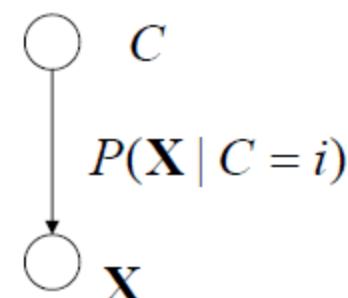
Why to add hidden variables?

- More flexibility in describing the distribution $P(\mathbf{X})$
- Smaller parameterization of $P(\mathbf{X})$
 - New independences can be introduced via hidden variables

Example:

- Latent variable models
 - hidden classes (categories)

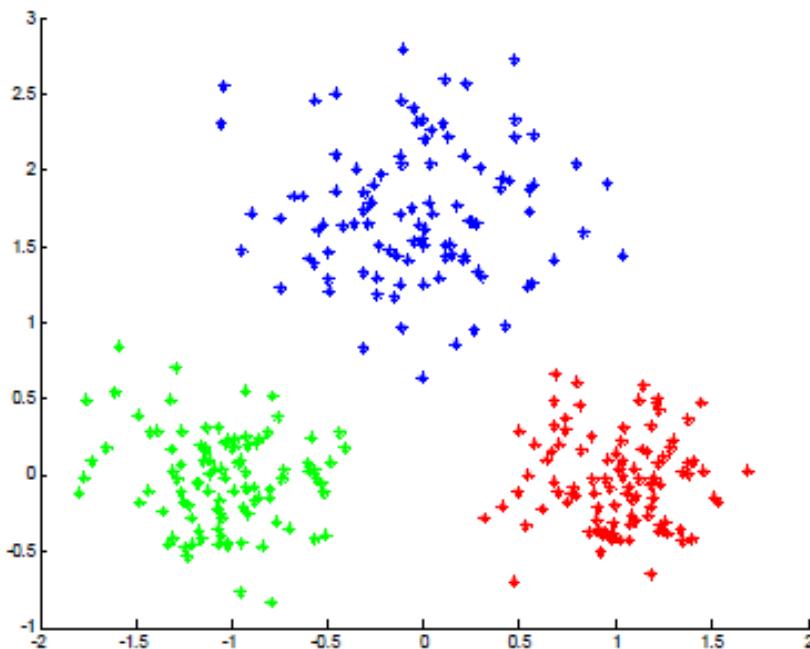
Hidden class variable



Hidden variable model. Example.

- We want to represent the probability model of a population in a two dimensional space $\mathbf{X} = \{X_1, X_2\}$

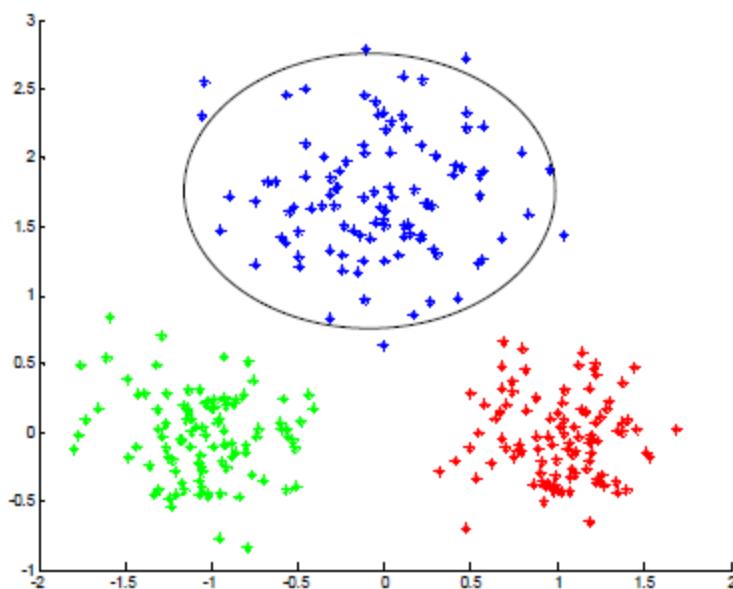
Observed data



Hidden variable model

- We want to represent the probability model of a population in a two dimensional space $\mathbf{X} = \{X_1, X_2\}$

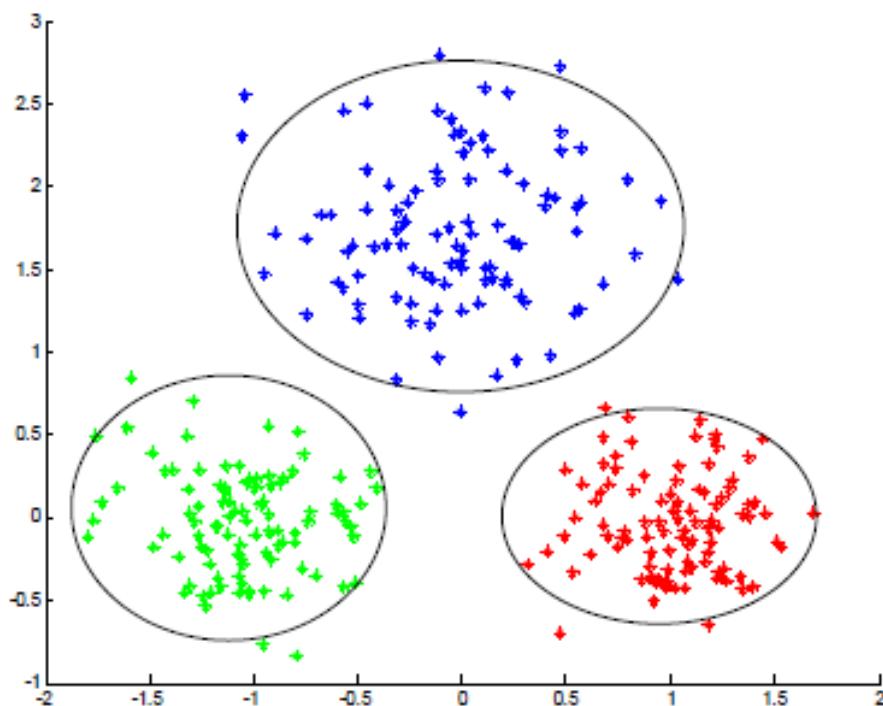
Observed data



Hidden variable model

- We want to represent a model of a population in a two dimensional space $\mathbf{X} = \{X_1, X_2\}$

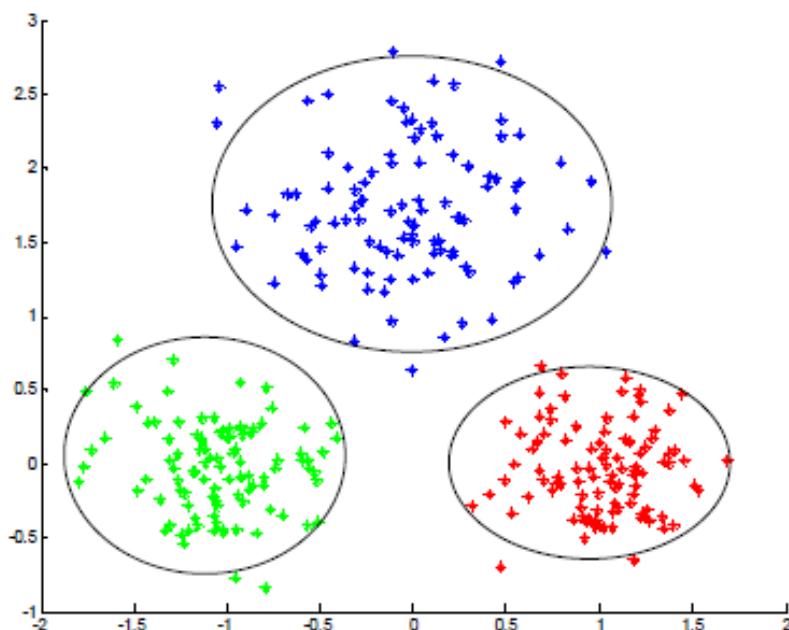
Observed data



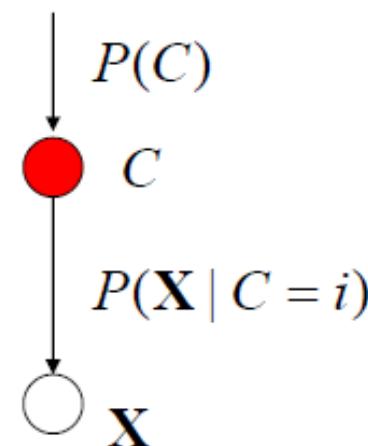
Hidden variable model

- We want to represent the probability model of a population in a two dimensional space $\mathbf{X} = \{X_1, X_2\}$

Observed data



Model : 3 Gaussians with
a hidden class variable



Mixture of Gaussians

Probability of the occurrence of a data point \mathbf{x} is modeled as

$$p(\mathbf{x}) = \sum_{i=1}^k p(C = i) p(\mathbf{x} | C = i)$$

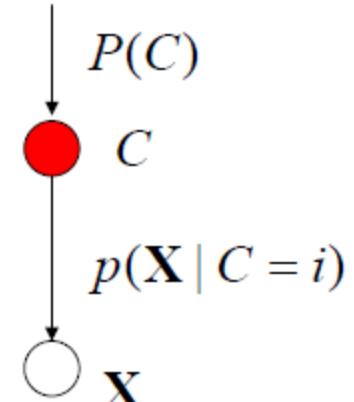
where

$$p(C = i)$$

= probability of a data point coming from class $C=i$

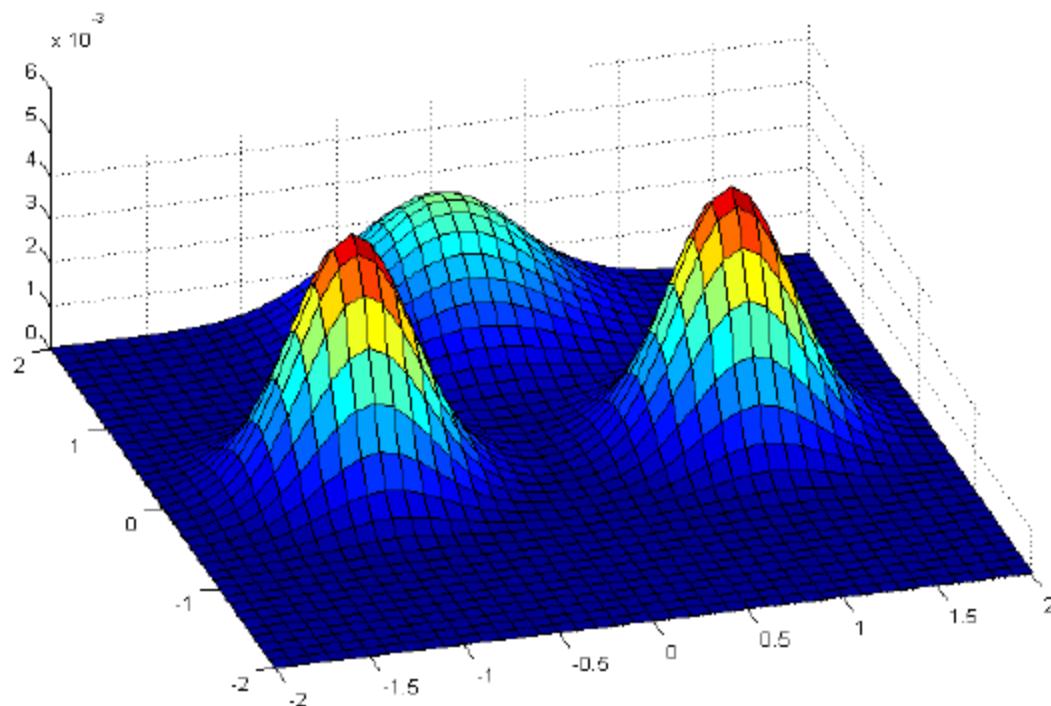
$$p(\mathbf{x} | C = i) \approx N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

= class-conditional density (modeled as Gaussian) for class i



Mixture of Gaussians

- Density function for the Mixture of Gaussians model



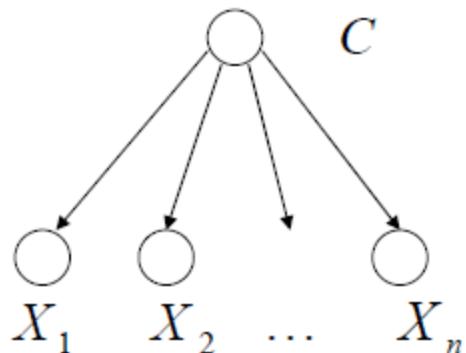
Naïve Bayes with a hidden class variable

Introduction of a hidden variable can reduce the number of parameters defining $P(\mathbf{X})$

Example:

- Naïve Bayes model with a hidden class variable

Hidden class variable



Attributes are independent given the class

- Useful in customer profiles
 - Class value = type of customers

Missing values

A set of random variables $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$

- **Data** $D = \{D_1, D_2, \dots, D_N\}$
- **But some values are missing**

$$D_i = (x_1^i, x_3^i, \dots, x_n^i)$$

Missing value of x_2^i

$$D_{i+1} = (x_3^i, \dots, x_n^i)$$

Missing values of x_1^i, x_2^i

Etc.

- **Example: medical records**
- **We still want to estimate parameters of $P(\mathbf{X})$**

Density estimation

Goal: Find the set of parameters $\hat{\Theta}$

Estimation criteria:

- **ML** $\max_{\Theta} p(D | \Theta, \xi)$
- **Bayesian** $p(\Theta | D, \xi)$

Optimization methods for ML: gradient-ascent, conjugate gradient, Newton-Raphson, etc.

- **Problem:** No or very small advantage from the structure of the corresponding belief network

Expectation-maximization (EM) method

- An alternative optimization method
- Suitable when there are missing or hidden values
- **Takes advantage of the structure of the belief network**

General EM

The key idea of a method:

Compute the parameter estimates iteratively by performing the following two steps:

Two steps of the EM:

1. **Expectation step.** Complete all hidden and missing variables with expectations for the current set of parameters Θ'
2. **Maximization step.** Compute the new estimates of Θ for the completed data

Stop when no improvement possible

EM

Let H – be a set of all variables with hidden or missing values

Derivation

$$P(H, D | \Theta, \xi) = P(H | D, \Theta, \xi)P(D | \Theta, \xi)$$

$$\log P(H, D | \Theta, \xi) = \log P(H | D, \Theta, \xi) + \log P(D | \Theta, \xi)$$

$$\underline{\log P(D | \Theta, \xi)} = \log P(H, D | \Theta, \xi) - \log P(H | D, \Theta, \xi)$$



Log-likelihood of data

Average both sides with $P(H | D, \Theta', \xi)$ for Θ'

$$E_{H|D,\Theta'} \log P(D | \Theta, \xi) = E_{H|D,\Theta'} \log P(H, D | \Theta, \xi) - E_{H|D,\Theta'} \log P(H | \Theta, \xi)$$

$$\underbrace{\log P(D | \Theta, \xi)}_{\text{Log-likelihood of data}} = Q(\Theta | \Theta') + H(\Theta | \Theta')$$

Log-likelihood of data

EM algorithm

Algorithm (general formulation)

Initialize parameters Θ

Repeat

Set $\Theta' = \Theta$

1. Expectation step

$$Q(\Theta | \Theta') = E_{H|D, \Theta'} \log P(H, D | \Theta, \xi)$$

2. Maximization step

$$\Theta = \arg \max_{\Theta} Q(\Theta | \Theta')$$

until no or small improvement in Θ ($\Theta = \Theta'$)

Questions: Why this leads to the ML estimate ?

What is the advantage of the algorithm?

EM algorithm

- Why is the EM algorithm correct?
- **Claim: maximizing Q improves the log-likelihood**

$$l(\Theta) = Q(\Theta | \Theta') + H(\Theta | \Theta')$$

Difference in log-likelihoods (current and next step)

$$l(\Theta) - l(\Theta') = Q(\Theta | \Theta') - Q(\Theta' | \Theta') + H(\Theta | \Theta') - H(\Theta' | \Theta')$$

Subexpression $H(\Theta | \Theta') - H(\Theta' | \Theta') \geq 0$

Kullback-Leibler (KL) divergence (distance between 2 distributions)

$$KL(P | R) = \sum_i P_i \log \frac{P_i}{R_i} \geq 0 \quad \text{Is always positive !!!}$$

$$H(\Theta | \Theta') = -E_{H|D,\Theta'} \log P(H | \Theta, \xi) = -\sum_{\{H\}} p(H | D, \Theta') \log P(H | \Theta, \xi)$$

$$H(\Theta | \Theta') - H(\Theta' | \Theta') = \sum_i P(H | D, \Theta') \log \frac{P(H | \Theta', \xi)}{P(H | \Theta, \xi)} \geq 0$$

EM algorithm

Difference in log-likelihoods

$$l(\Theta) - l(\Theta') = Q(\Theta | \Theta') - Q(\Theta' | \Theta') + H(\Theta | \Theta') - H(\Theta' | \Theta')$$

$$l(\Theta) - l(\Theta') \geq Q(\Theta | \Theta') - Q(\Theta' | \Theta')$$

Thus

by **maximizing Q** we **maximize the log-likelihood**

$$l(\Theta) = Q(\Theta | \Theta') + H(\Theta | \Theta')$$

EM is a first-order optimization procedure

- **Climbs the gradient**
- **Automatic learning rate**

No need to adjust the learning rate !!!!

EM advantages

Key advantages:

- In many problems (e.g. Bayesian belief networks)

$$Q(\Theta | \Theta') = E_{H|D,\Theta'} \log P(H, D | \Theta, \xi)$$

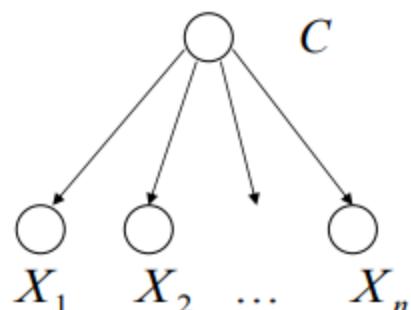
- has a nice form and the maximization of Q can be carried in the closed form
- No need to compute Q before maximizing
- We directly optimize
 - use quantities corresponding to expected counts

Naïve Bayes with a hidden class and missing values

Assume:

- $P(\mathbf{X})$ is modeled using a Naïve Bayes model with hidden class variable
- Missing entries (values) for attributes in the dataset D

Hidden class variable



Attributes are independent given the class

EM for the Naïve Bayes

- We can use EM to learn the parameters

$$Q(\Theta | \Theta') = E_{H|D, \Theta'} \log P(H, D | \Theta, \xi)$$

- Parameters:

π_j prior on class j

θ_{ijk} probability of an attribute i having value k given class j

- Indicator variables:

δ_j^l for example l , the class is j ; if true (=1) else false (=0)

δ_{ijk}^l for example l , the class is j and the value of attrib i is k

- because the class is hidden and some attributes are missing, the values (0,1) of indicator variables are not known; they are hidden

H – a collection of all indicator variables

EM for the Naïve Bayes model

- We can use EM to do the learning of parameters

$$Q(\Theta | \Theta') = E_{H|D,\Theta'} \log P(H, D | \Theta, \xi)$$

$$\begin{aligned} \log P(H, D | \Theta, \xi) &= \log \prod_{l=1}^N \prod_j \pi_j^{\delta_j^l} \prod_i \prod_k \theta_{ijk}^{\delta_{ijk}^l} \\ &= \sum_{l=1}^N \sum_j (\delta_j^l \log \pi_j + \sum_i \sum_k \delta_{ijk}^l \log \theta_{ijk}) \end{aligned}$$

$$E_{H|D,\Theta'} \log P(H, D | \Theta, \xi) = \sum_{l=1}^N \sum_j (E_{H|D,\Theta'}(\delta_j^l) \log \pi_j + \sum_i \sum_k E_{H|D,\Theta'}(\delta_{ijk}^l) \log \theta_{ijk})$$

$$E_{H|D,\Theta'}(\delta_j^l) = p(C_l = j | D_l, \Theta')$$

Substitutes 0,1

$$E_{H|D,\Theta'}(\delta_{ijk}^l) = p(X_{il} = k, C_l = j | D_l, \Theta')$$

with expected value

EM for the Naïve Bayes model

- Computing derivatives of \mathcal{Q} for parameters and setting it to 0 we get:

$$\pi_j = \frac{\tilde{N}_j}{N} \quad \theta_{ijk} = \frac{\tilde{N}_{ijk}}{\sum_{k=1}^{r_i} \tilde{N}_{ijk}}$$

$$\tilde{N}_j = \sum_{l=1}^N E_{H|D, \Theta'}(\delta_j^l) = \sum_{l=1}^N p(C_l = j | D_l, \Theta')$$

$$\tilde{N}_{ijk} = \sum_{l=1}^N E_{H|D, \Theta'}(\delta_{ijk}^l) = \sum_{l=1}^N p(X_{il} = k, C_l = j | D_l, \Theta')$$

- **Important:**
 - Use expected counts instead of counts !!!
 - Re-estimate the parameters using expected counts



Thank you



K-Nearest-Neighbor

Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University

Course Outline

- Non parametric learning
- K-nearest neighbours
- K-NN for prediction
- Merits and demerits
- Summary

Parametric vs Non parametric Models

- In the models that we have seen, we select a hypothesis space and adjust a fixed set of parameters with the training data ($h_\alpha(x)$)
- We assume that the parameters α summarize the training and we can forget about it
- These methods are called parametric models
- When we have a small amount of data it makes sense to have a small set of parameters and to constraint the complexity of the model (avoiding overfitting)

Parametric vs Non parametric Models

- When we have a large quantity of data, overfitting is less an issue
- If data shows that the hypothesis has to be complex, we can try to adjust to that complexity
- A **non parametric model** is one that can not be characterized by a fixed set of parameters
- A family of non parametric models is **Instance Based Learning**

Instance Based Learning

- Instance based learning is based on the memorization of the dataset
- The number of parameters is unbounded and grows with the size of the data
- There is not a model associated to the learned concepts
- The classification is obtained by looking into the memorized examples
- The cost of the learning process is 0, all the cost is in the computation of the prediction
- This kind learning is also known as **lazy learning**

Characteristics

- Data-driven, not model-driven
- Makes no assumptions about the data

Basic Idea

For a given record to be classified, identify nearby records

“Near” means records with similar predictor values
 X_1, X_2, \dots, X_p

Classify the record as whatever the predominant class is among the nearby records (the “neighbors”)

K-nearest neighbours

- K-nearest neighbours uses the local neighborhood to obtain a prediction
- The K memorized examples more similar to the one that is being classified are retrieved
- A distance function is needed to compare the examples similarity
- This means that if we change the distance function, we change how examples are classified

How to measure “nearby”?

The most popular distance measure is
Euclidean distance

- Euclidean distance ($d(x_j, x_k) = \sqrt{\sum_i (x_{j,i} - x_{k,i})^2}$)
- Mahnattan distance ($d(x_j, x_k) = \sum_i |x_{j,i} - x_{k,i}|$)

K-nearest neighbours - Algorithm

- Training: Store all the examples
- Prediction: $h(x_{new})$
 - Let be x_1, \dots, x_k the k more similar examples to x_{new}
 - $h(x_{new}) = \text{combine_predictions}(x_1, \dots, x_k)$
- The parameters of the algorithm are the number k of neighbours and the procedure for combining the predictions of the k examples
- The value of k has to be adjusted (crossvalidation)
 - We can overfit (k too low)
 - We can underfit (k too high)

Choosing k

K is the number of nearby neighbors to be used to classify the new record

$K=1$ means use the single nearest record

$K=5$ means use the 5 nearest records

Typically choose that value of k which has lowest error rate in validation data

Low k vs. High k

Low values of k ($1, 3, \dots$) capture local structure in data (but also noise)

High values of k provide more smoothing, less noise, but may miss local structure

Note: the extreme case of $k = n$ (i.e., the entire data set) is the same as the “naïve rule” (classify all records according to majority class)

Example: Riding Mowers

Data: 24 households classified as owning or not owning riding mowers

Predictors: Income, Lot Size

Income	Lot_Size	Ownership
60.0	18.4	owner
85.5	16.8	owner
64.8	21.6	owner
61.5	20.8	owner
87.0	23.6	owner
110.1	19.2	owner
108.0	17.6	owner
82.8	22.4	owner
69.0	20.0	owner
93.0	20.8	owner
51.0	22.0	owner
81.0	20.0	owner
75.0	19.6	non-owner
52.8	20.8	non-owner
64.8	17.2	non-owner
43.2	20.4	non-owner
84.0	17.6	non-owner
49.2	17.6	non-owner
59.4	16.0	non-owner
66.0	18.4	non-owner
47.4	16.4	non-owner
33.0	18.8	non-owner
51.0	14.0	non-owner
63.0	14.8	non-owner

XLMiner Output

For each record in validation data (6 records) XLMiner finds neighbors amongst training data (18 records).

The record is scored for $k=1, k=2, \dots k=18$.

Best k appears to be $k=8$.

$k = 9, k = 10, k=14$ also share low error rate, but best to choose lowest k .

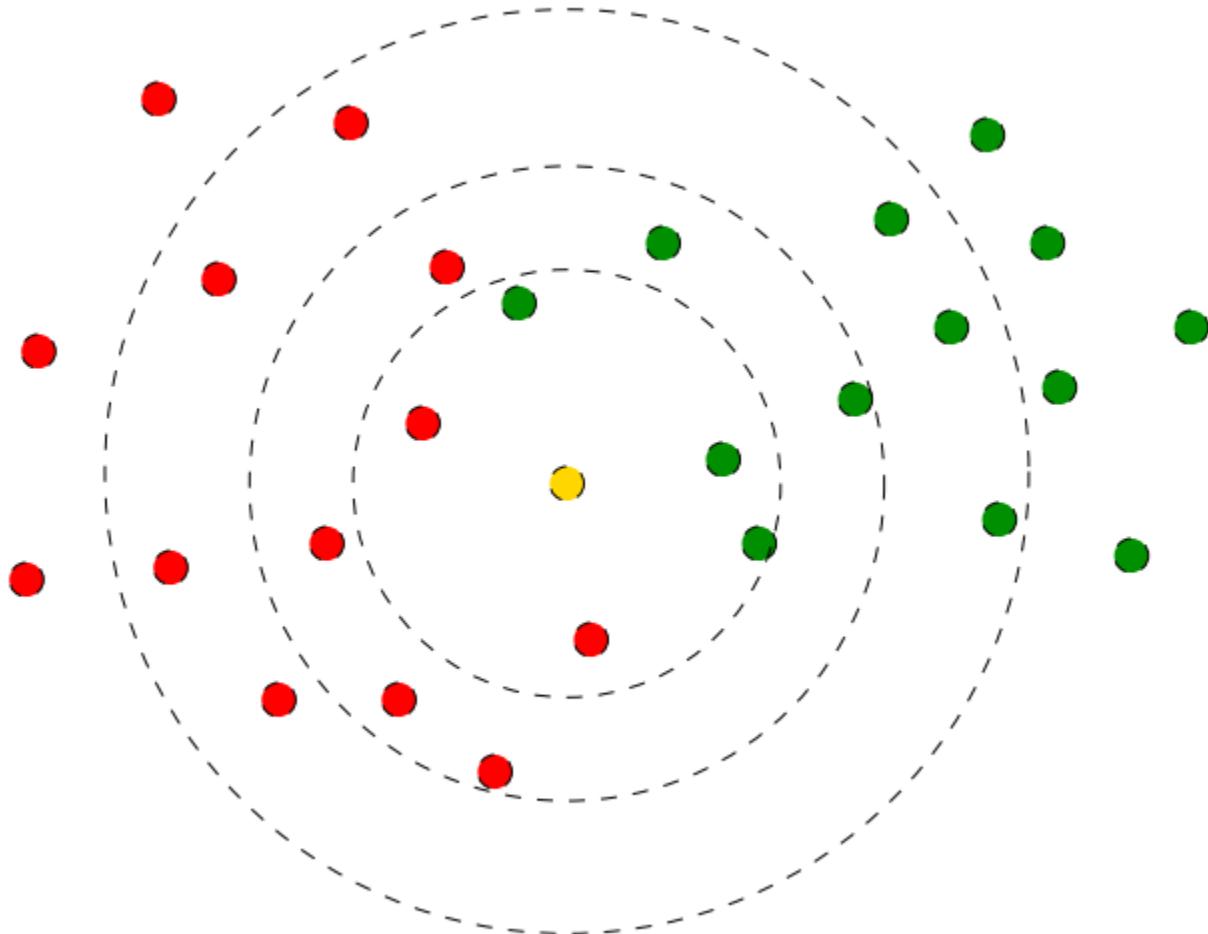
Value of k	% Error Training	% Error Validation
1	0.00	33.33
2	16.67	33.33
3	11.11	33.33
4	22.22	33.33
5	11.11	33.33
6	27.78	33.33
7	22.22	33.33
8	22.22	16.67
9	22.22	16.67
10	22.22	16.67
11	16.67	33.33
12	16.67	16.67
13	11.11	33.33
14	11.11	16.67
15	5.56	33.33
16	16.67	33.33
17	11.11	33.33
18	50.00	50.00

<--- Best k

Using K-NN for Prediction (for Numerical Outcome)

- Instead of “majority vote determines class” use average of response values
- May be a weighted average, weight decreasing with distance

Using K-NN for Prediction (for Numerical Outcome)



Looking for neighbours

- Looking for the K-nearest examples for a new example can be expensive
- The straightforward algorithm has a cost $O(n \log(k))$, not good if the dataset is large
- We can use indexing with *k-d trees* (multidimensional binary search trees)
 - They are good only if we have around 2^{\dim} examples, so not good for high dimensionality
- We can use *locality sensitive hashing* (approximate k-nn)
 - Examples are inserted in multiple hash tables that use hash functions that with high probability put together examples that are close
 - We retrieve from all the hash tables the examples that are in the bin of the query example
 - We compute the k-nn only with these examples

K-nearest neighbours - Variants

- There are different possibilities for computing the class from the k nearest neighbours
 - Majority vote
 - Distance weighted vote
 - Inverse of the distance
 - Inverse of the square of the distance
 - Kernel functions (gaussian kernel, tricube kernel, ...)
- Once we use weights for the prediction we can relax the constraint of using only k neighbours
 - ① We can use k examples (local model)
 - ② We can use all examples (global model)

K-nearest neighbours - Regression

- We can extend this method from classification to regression
- Instead of combining the discrete predictions of k-neighbours we have to combine continuous predictions
- These predictions can be obtained in different ways:
 - Simple interpolation
 - Averaging
 - Local linear regression
 - Local weighted regression
- The time complexity of the prediction will depend on the method

Advantages

- Simple
- Effective at capturing complex interactions among variables without having to define a statistical model
- The cost of the learning process is zero
- No assumptions about the characteristics of the concepts to learn have to be done
- Complex concepts can be learned by local approximation using simple procedures

Shortcomings

- The model can not be interpreted (there is no description of the learned concepts)
- Required size of training set increases exponentially with # of predictors, p
This is because expected distance to nearest neighbor increases with p (with large vector of predictors, all records end up “far away” from each other)
- In a large training set, it takes a long time to find distances to all the neighbors and then identify the nearest one(s)
- Performance depends on the number of dimensions that we have
 - (curse of dimensionality) \Rightarrow Attribute Selection

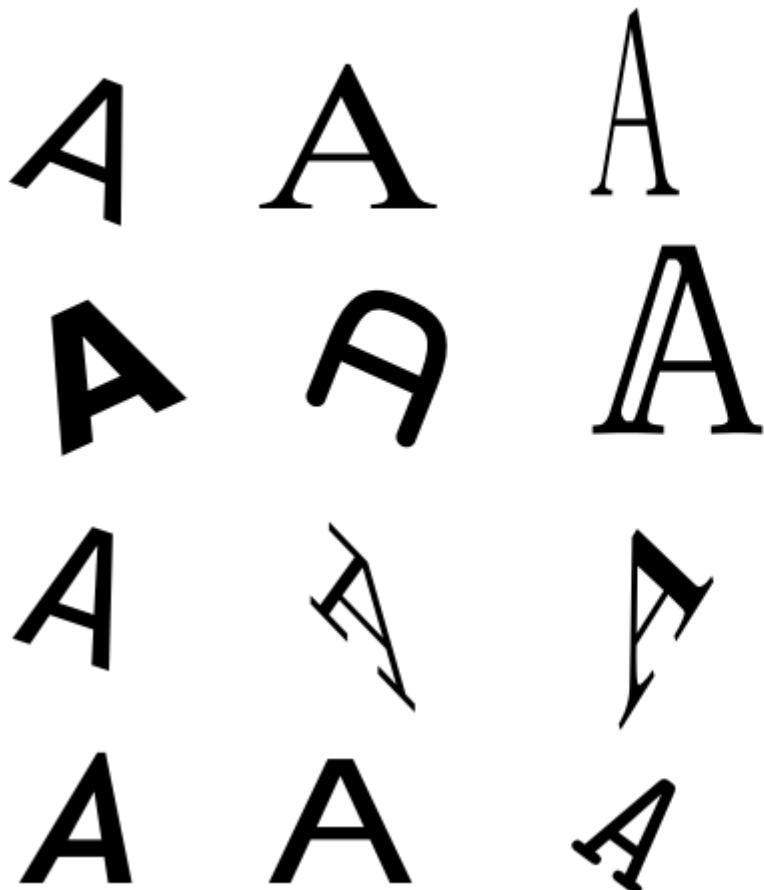
The Curse of dimensionality

- The more dimensions we have, the more examples we need to approximate a hypothesis
- The number of examples that we have in a volume of space decreases exponentially with the number of dimensions
- This is specially bad for k-nearest neighbors
 - If the number of dimensions is very high the nearest neighbours can be very far away

Dealing with the Curse

- Reduce dimension of predictors (e.g., with PCA)
- Computational shortcuts that settle for “almost nearest neighbors”

Optical Character Recognition



- OCR capital letters
- 14 Attributes (All continuous)
- Attributes: horizontal position of box, vertical position of box, width of box, height of box, total num on pixels, mean x of on pixels in box, ...
- 20000 instances
- 26 classes (A-Z)
- Validation: 10 fold cross validation

Optical Character Recognition: Models

- K-nn 1 (Euclidean distance, weighted): accuracy 96.0%
- K-nn 5 (Manhattan distance, weighted): accuracy 95.9%
- K-nn 1 (Correlation distance, weighted): accuracy 95.1%

Summary

- Find distance between record-to-be-classified and all other records
- Select k-nearest records
 - Classify it according to majority vote of nearest neighbors
 - Or, for prediction, take the average of the nearest neighbors
- “Curse of dimensionality” – need to limit # of predictors

Linear Classification Models

Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University

Classification

- The goal of classification is to use an object's characteristics to identify which class (or group) it belongs to.
- A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics.
- An object's characteristics are also known as feature values and are typically presented to the machine in a vector called a feature vector.
- Such classifiers work well for practical problems such as document classification, and more generally for problems with many variables (features), reaching accuracy levels comparable to non-linear classifiers while taking less time to train and use.

Classification vs. Prediction

- Classification
 - predicts categorical class labels (discrete or nominal)
 - classifies data (constructs a model) based on the training set and the values (**class labels**) in a classifying attribute and uses it in classifying new data
- Prediction
 - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical applications
 - Credit approval
 - Target marketing
 - Medical diagnosis
 - Fraud detection

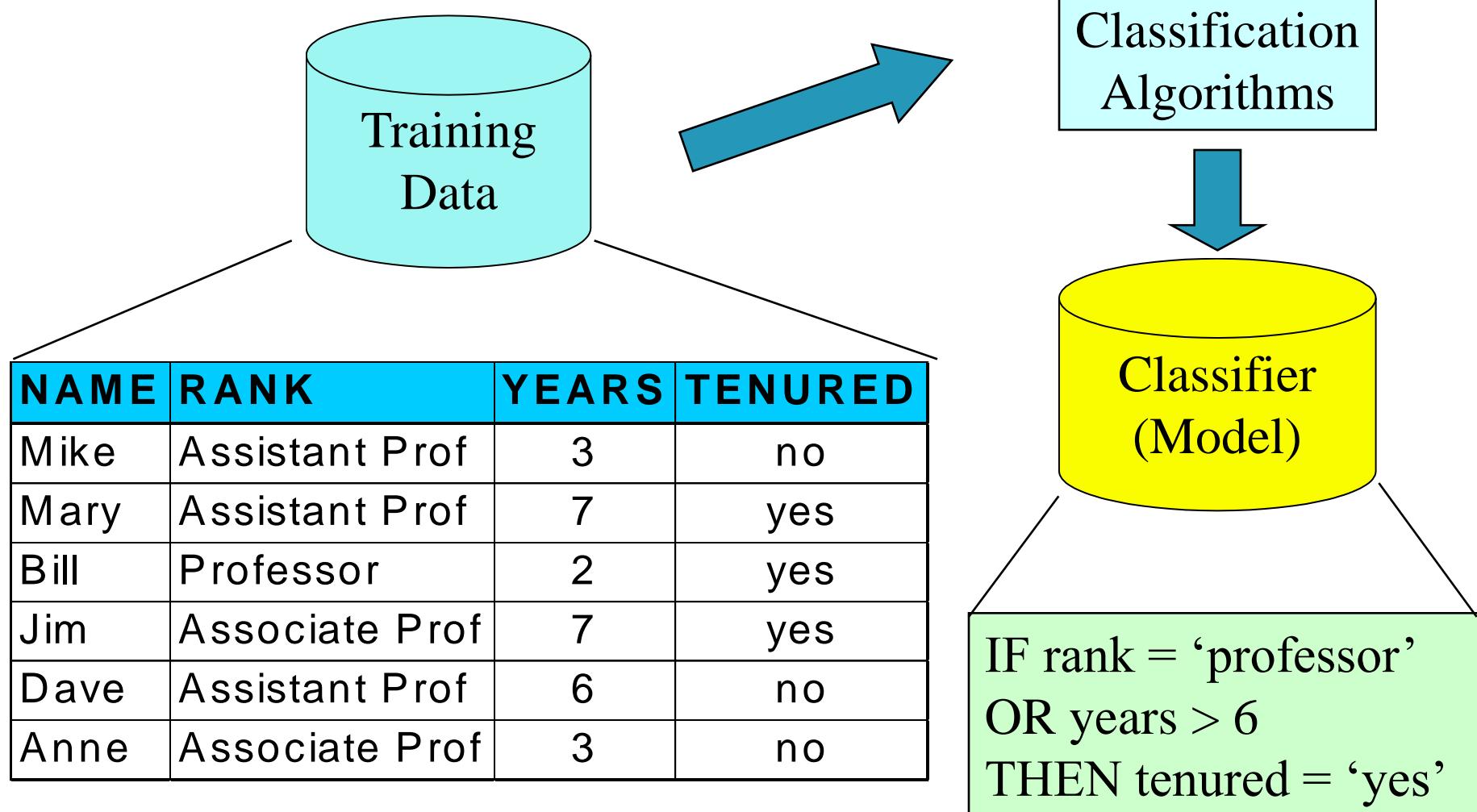
Regression vs Classification

- In *Regression* we assign input vector x to one or more continuous target variables t
 - Linear regression has simple analytical and computational properties
- In *Classification* we assign input vector x to one of K discrete classes C_k , $k = 1, \dots, K$.
 - We discuss here linear models for Classification
 - Ordinal Regression is a form of classification where discrete classes have an ordering
 - E.g., relevance score regression

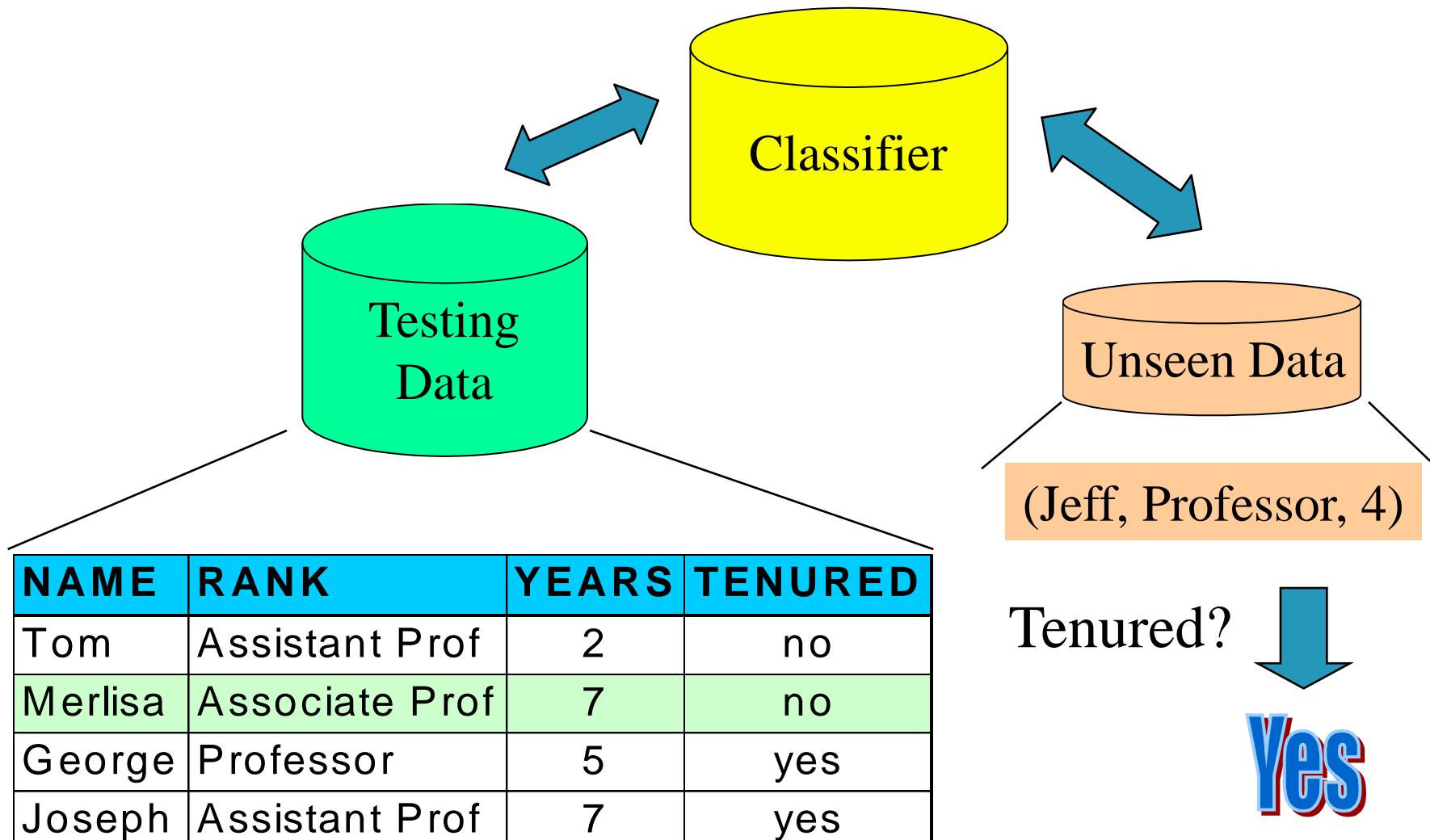
Classification—A Two-Step Process

- **Model construction:** describing a set of predetermined classes
 - Each tuple/sample is assumed to belong to a predefined class, as determined by the **class label attribute**
 - The set of tuples used for model construction is **training set**
 - The model is represented as classification rules, decision trees, or mathematical formulae
- **Model usage:** for classifying future or unknown objects
 - **Estimate accuracy** of the model
 - The known label of test sample is compared with the classified result from the model
 - Accuracy rate is the percentage of test set samples that are correctly classified by the model
 - Test set is independent of training set, otherwise over-fitting will occur
 - If the accuracy is acceptable, use the model to **classify data** tuples whose class labels are not known

Process (1): Model Construction



Process (2): Using the Model in Prediction



Supervised vs. Unsupervised Learning

- Supervised learning (classification)
 - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
 - New data is classified based on the training set
- Unsupervised learning (clustering)
 - The class labels of training data is unknown
 - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Issues: Data Preparation

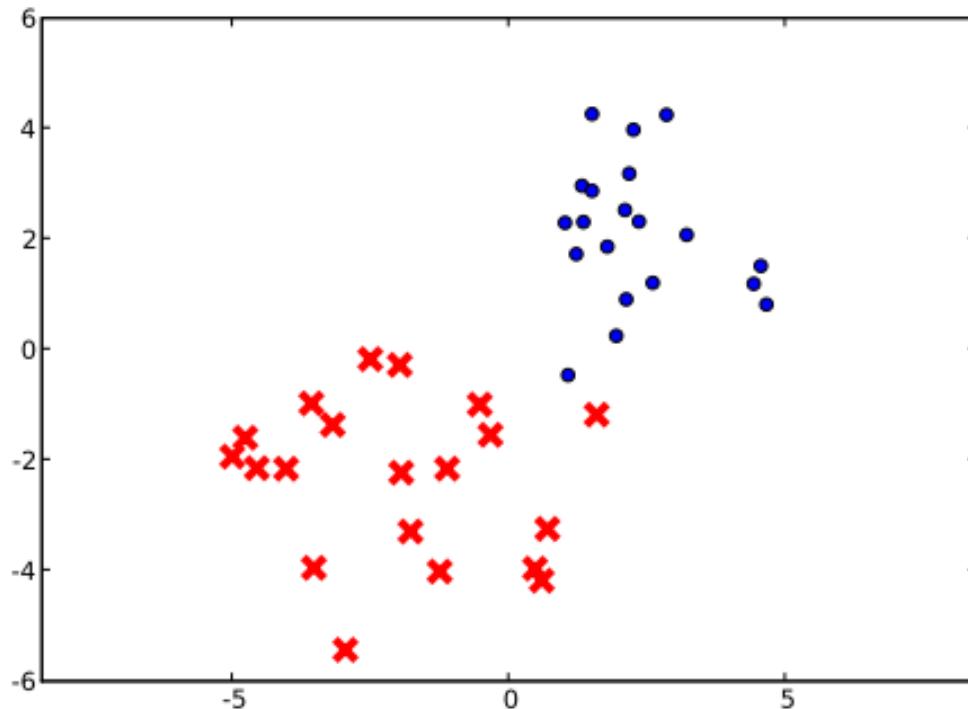
- Data cleaning
 - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
 - Remove the irrelevant or redundant attributes
- Data transformation
 - Generalize and/or normalize data

Issues: Evaluating Classification Methods

- Accuracy
 - classifier accuracy: predicting class label
 - predictor accuracy: guessing value of predicted attributes
- Speed
 - time to construct the model (training time)
 - time to use the model (classification/prediction time)
- Robustness: handling noise and missing values
- Scalability: efficiency in disk-resident databases
- Interpretability
 - understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

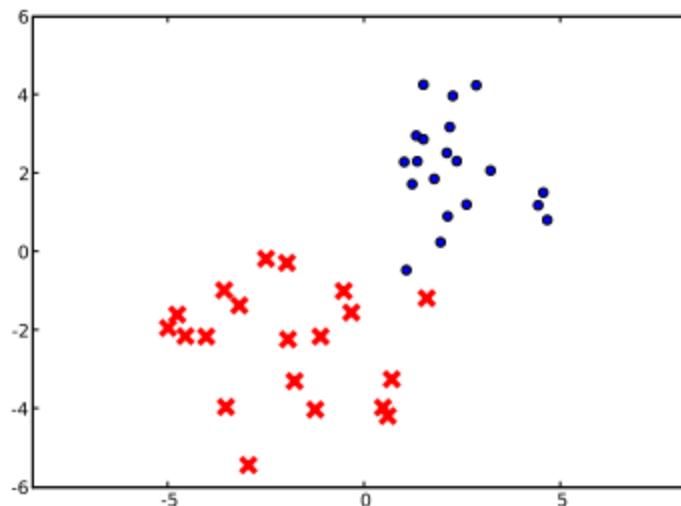
Basic Setup

- ▶ We want to separate the X's and the O's
- ▶ Today, we will see how to solve this (seemingly) simple task mathematically



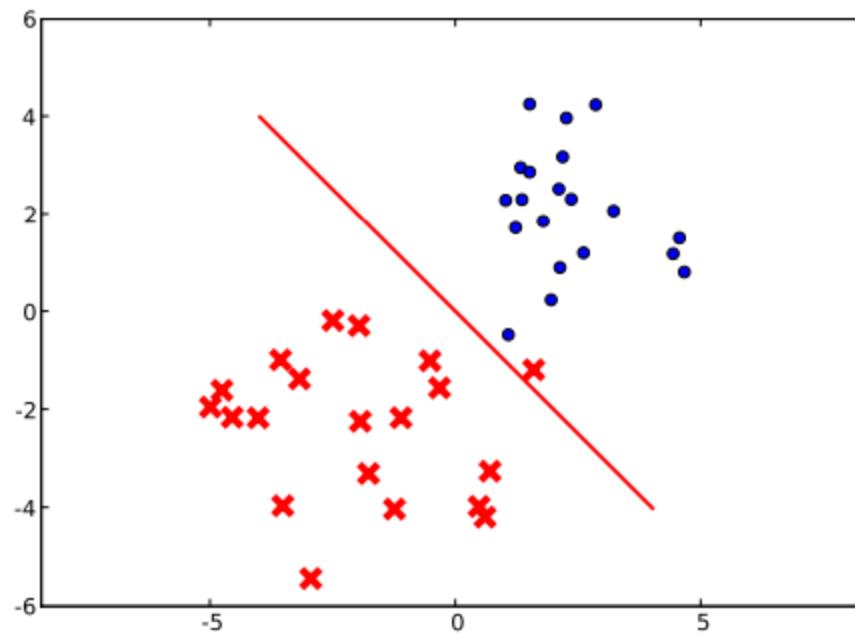
Basic Vocabulary

- ▶ We'll call each thing that we are classifying a *point*
- ▶ Each point is described by a set of numbers that we call *features*
 - ▶ It is your job to decide on the features. You need to pick features that help you do your job.
 - ▶ For instance, if you are looking for edges, you'll probably want to look at the response of different derivative filters.
- ▶ In the graph below, every point is described by two features.
- ▶ The point at index i in the dataset will be described as \mathbf{x}_i



Separating Points Linearly

- ▶ In building mathematical models for classifying, we are going to focus on dividing these points with a straight line.
- ▶ This is called linear classification



Expressing Linear Separation Mathematically

- Given a single point $\mathbf{x} = (x, y)$, we can express the classification of the point as

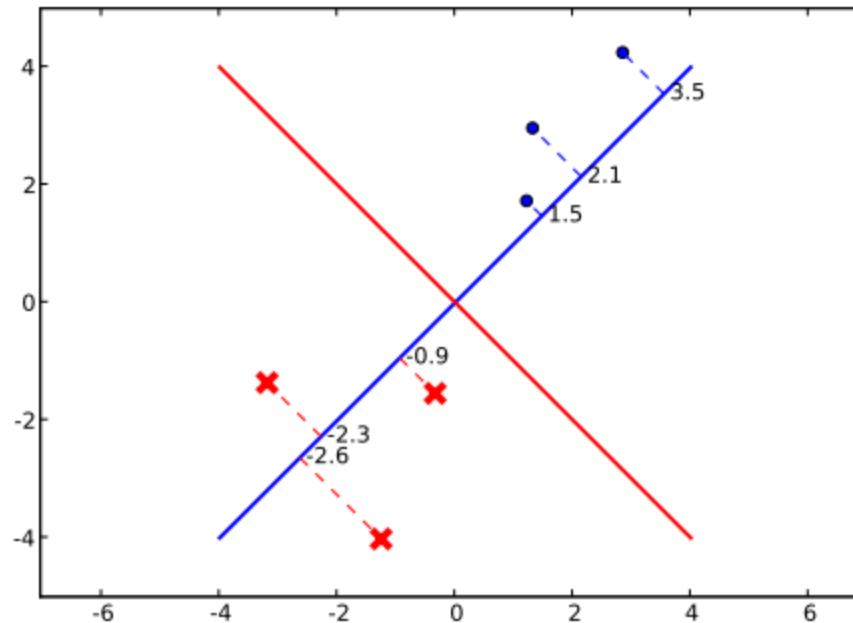
$$\text{sign}(ax + by + c)$$

where a , b , and c are constants that define a line. We'll have to choose these somehow.

- This function will return a $+1$ if $ax + by + c$ is positive and -1 otherwise.

How this Translates Graphically

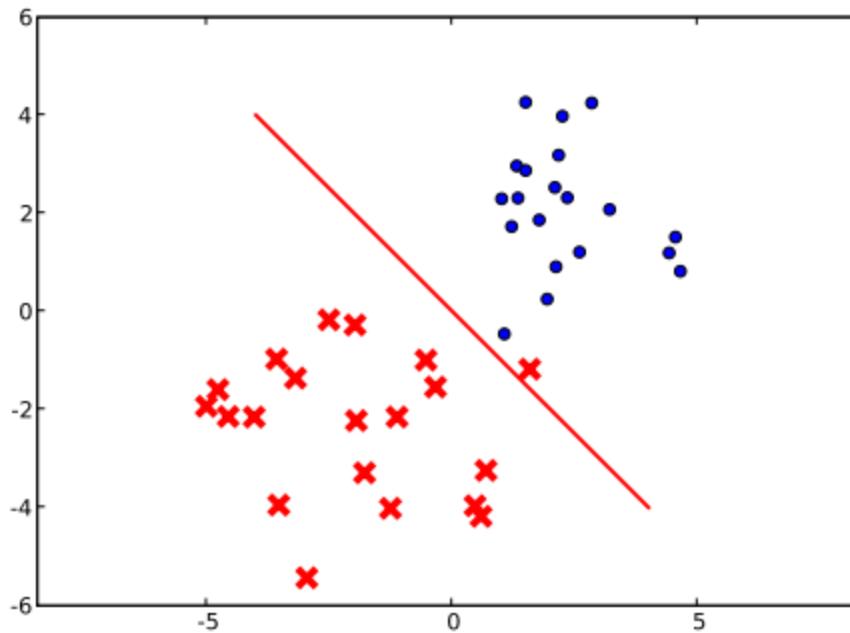
- ▶ Effectively, we are projecting every point onto a line.
- ▶ Every point projects to some point on the line. The sign of the location along the lines determines the classification of the point



How can we find the separating line?

- ▶ This separating line can be found by looking at the line

$$ax + by + c = 0$$

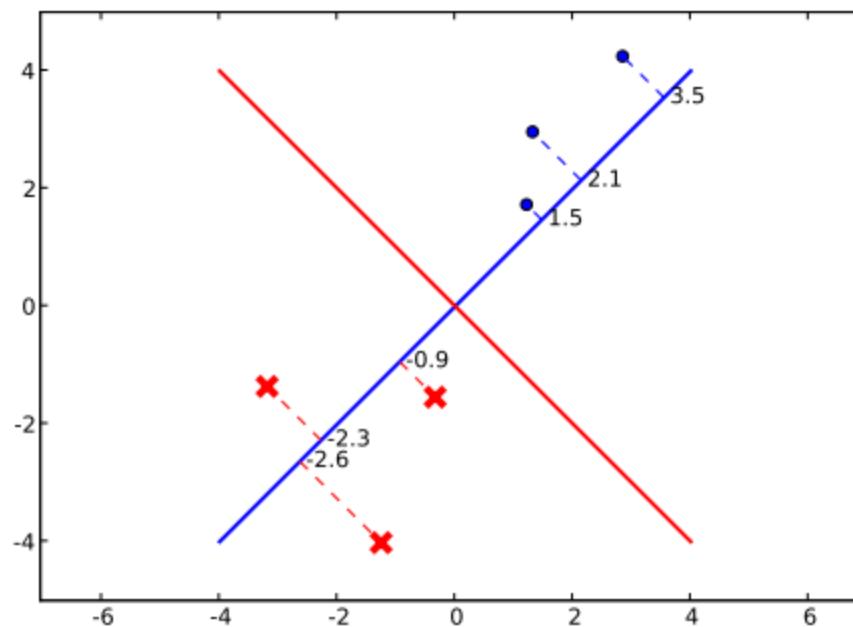


How do we get the parameters of this line?

- ▶ We'll find the parameters using a machine learning approach
- ▶ We'll form a *training set* of examples, along with the correct classification.
- ▶ We'll find the line that best separates the training examples.
- ▶ Now, we have to form a set of mathematical steps for finding the line that "best separates" the training set.

Optimizing the parameters of the line

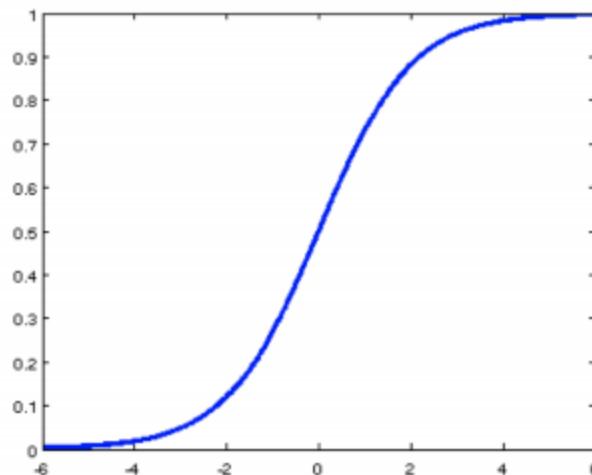
- ▶ Notice that the points that are the farthest from the red separating line have the largest response.
- ▶ In some sense, we can be more confident in a point's classification as the point gets farther from the separating line.
- ▶ So, the bigger the magnitude of a response, the more confident in the classification we can be.



Looking at the confidence in classification

- ▶ We can translate this response into a probability.
- ▶ We will use the *logistic function* to transform the response into a probability of the correct classification
- ▶ We will assume that each point will be labeled with a label l . l can take values $+1$ or -1 .

$$P[l = +1 | \mathbf{x}] = \frac{1}{1 + \exp(-(ax + by + c))}$$



Finding the line parameters

- ▶ Now, for any set of line constants, we can find out the probability assigned to the correct label of each item in the training set.

$$\prod_{i=1}^N P[l_i | \mathbf{x}_i] = \prod_{i=1}^N \frac{1}{1 + \exp(-l_i(ax_i + by_i + c))}$$

- ▶ We've inserted l_i into the exponent because, if $l_i = -1$

$$P[l = -1 | \mathbf{x}] = \frac{1}{1 + \exp((ax + by + c))}$$

- ▶ We multiply the probabilities because we believe the points are drawn independently.
- ▶ Note that for each item, this number tells us how much the model defined by that particular line believes that the ground-truth right answer is the actual right answer.

Linear vs Nonlinear Models

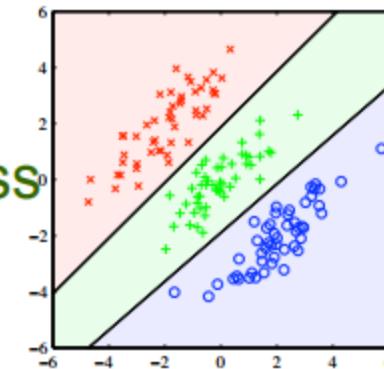
- A linear classifier is based on the value of a linear combination of the characteristics.
- At the most fundamental point, linear methods can only solve problems that are linearly separable (usually via a hyperplane).
- If you can solve it with a linear method, you're usually better off.
- However, if linear isn't working for your particular problem, the next step is to use a nonlinear method, which typically involves applying some type of transformation to your input dataset.
- After the transformation, many techniques then try to use a linear method for separation.

Linear vs Nonlinear Models

- Linear:
 - LDA
 - Decision tree
 - Naïve bayes
 - Regression
 - KNN
- Nonlinear:
 - SVM
 - ANN

Linear Classification Models

- Common classification scenario:
classes considered disjoint
 - Each input assigned to only one class
- Input space divided into decision regions
- Decision surfaces are linear functions of input x
 - Defined by $(D - 1)$ dimensional hyperplanes within D dim. input space



Straight line is 1-D in 2-D
A plane is 2-D in 3-D

Data sets whose classes can be separated exactly by linear decision surfaces are said to be Linearly separable

Representing the target in Classification

- In *regression* target variable t is a real number (or vector of real numbers \mathbf{t}) which we wish to predict
- In *classification* there are various ways of using target values to represent class labels, depending on whether
 - Model is probabilistic
 - Model is non-probabilistic

Representing Class in Probabilistic Model

- Two class: Binary representation is convenient
 - Discrete $t \in \{0, 1\}$, $t = 1$ represents C_1 ,
 $t = 0$ means class C_2
 - Can interpret value of t as probability that class is C_1
 - Probabilities taking only extreme values of 0 and 1
- For $K > 2$: Use a 1-of- K coding scheme.
 - \mathbf{t} is a vector of length K
 - Eg. if $K = 5$, a pattern of class 2 has $\mathbf{t} = (0, 1, 0, 0, 0)^T$
 - Value of t_k interpreted as probability of class C_k
 - If t_k assume real values then we allow different class probabilities

Representing Class: Nonprobabilistic

- For non-probabilistic models, e.g, nearest neighbor
 - other choices of target variable representation used

Two Approaches to Classification

1. Discriminant function

- Directly assign x to a specific class
 - E.g., Fisher's Linear Disc, Perceptron

2. Probabilistic Models

1. Model $p(C_k|x)$ in *inference* stage (direct or $p(x|C_k)$)
2. Use it to make *optimal* decisions

Separating Inference from Decision is better:

- Minimize risk (loss function can change in financial app)
- Reject option (minimize expected loss)
- Compensate for unbalanced data
 - use modified balanced data & scale by class fractions
- Combine models

8

Probabilistic Models: Generative/Discriminative

- Model $p(C_k|x)$ in an *inference* stage and use it to make optimal decisions
- Two approaches to computing the $p(C_k|x)$
 - **Generative**
 - Model class conditional densities by $p(x|C_k)$ together with prior probabilities $p(C_k)$
 - Then use Bayes rule to compute posterior
$$p(C_k|x) = p(x|C_k)p(C_k)/p(x)$$
 - **Discriminative**
 - Directly model conditional probabilities $p(C_k|x)$

From Regression to Classification

- Linear Regression model $y(\mathbf{x}, \mathbf{w})$ is a linear function of parameters \mathbf{w}
 - In simple case model is also a linear function of \mathbf{x}
 - Thus has the form $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$ where y is a real no.
- For classification we need to predict class labels or posterior probabilities in range $(0,1)$
 - For this, we use a generalization where we transform the linear function of \mathbf{w} using a nonlinear function $f(\cdot)$, so that

$$y(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x} + w_0)$$

- $f(\cdot)$ is known as an *activation function*
- Whereas its inverse is called a *link function* in statistics
 - link function provides relationship between the linear predictor and the mean of the distribution function

Overview of Linear Classifiers

1. Discriminant Functions

- Two class and Multi class
- Least squares for classification
- Fisher's linear discriminant
- Perceptron algorithm

2. Probabilistic Generative Models

- Continuous inputs and max likelihood
- Discrete inputs, Exponential Family

3. Probabilistic Discriminative Models

- Logistic regression for single and multi class
- Laplace approximation
- Bayesian logistic regression

Bayesian Classification: Why?

- A statistical classifier: performs *probabilistic prediction*, i.e., predicts class membership probabilities
- Foundation: Based on Bayes' Theorem.
- Performance: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Bayesian Theorem: Basics

- Let \mathbf{X} be a data sample ("*evidence*"): class label is unknown
- Let H be a *hypothesis* that X belongs to class C
- Classification is to determine $P(H|\mathbf{X})$, the probability that the hypothesis holds given the observed data sample \mathbf{X}
- $P(H)$ (*prior probability*), the initial probability
 - E.g., \mathbf{X} will buy computer, regardless of age, income, ...
- $P(\mathbf{X})$: probability that sample data is observed
- $P(\mathbf{X}|H)$ (*posteriori probability*), the probability of observing the sample \mathbf{X} , given that the hypothesis holds
 - E.g., Given that \mathbf{X} will buy computer, the prob. that X is 31..40, medium income

Bayesian Theorem

- Given training data \mathbf{X} , *posteriori probability of a hypothesis H*, $P(H|\mathbf{X})$, follows the Bayes theorem

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

- Informally, this can be written as
posteriori = likelihood x prior/evidence
- Predicts \mathbf{X} belongs to C_2 iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|\mathbf{X})$ for all the k classes
- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Towards Naïve Bayesian Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n -D attribute vector $\mathbf{X} = (x_1, x_2, \dots, x_n)$
- Suppose there are m classes C_1, C_2, \dots, C_m .
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Since $P(X)$ is constant for all classes, only

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

needs to be maximized

Derivation of Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X} | C_i) = \prod_{k=1}^n P(x_k | C_i) = P(x_1 | C_i) \times P(x_2 | C_i) \times \dots \times P(x_n | C_i)$$

- This greatly reduces the computation cost: Only counts the class distribution
- If A_k is categorical, $P(x_k | C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i,D}|$ (# of tuples of C_i in D)
- If A_k is continuous-valued, $P(x_k | C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

and $P(x_k | C_i)$ is

$$P(\mathbf{X} | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

Naïve Bayesian Classifier: Training Dataset

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Class:

C1:buys_computer = 'yes'
C2:buys_computer = 'no'

Data sample

X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)

Naïve Bayesian Classifier: An Example

- $P(C_i)$:
 $P(\text{buys_computer} = \text{"yes"}) = 9/14 = 0.643$
 $P(\text{buys_computer} = \text{"no"}) = 5/14 = 0.357$
- Compute $P(X|C_i)$ for each class
 $P(\text{age} = \text{"<=30"} | \text{buys_computer} = \text{"yes"}) = 2/9 = 0.222$
 $P(\text{age} = \text{"<= 30"} | \text{buys_computer} = \text{"no"}) = 3/5 = 0.6$
 $P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"yes"}) = 4/9 = 0.444$
 $P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$
 $P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$
 $P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"no"}) = 1/5 = 0.2$
 $P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$
 $P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$$\mathbf{P(X|C_i)} : P(X|\text{buys_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$
$$P(X|\text{buys_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$

$$\mathbf{P(X|C_i)*P(C_i)} : P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) = 0.028$$
$$P(X|\text{buys_computer} = \text{"no"}) * P(\text{buys_computer} = \text{"no"}) = 0.007$$

Therefore, X belongs to class ("buys_computer = yes")

Avoiding the 0-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be non-zero. Otherwise, the predicted prob. will be zero

$$P(X | C_i) = \prod_{k=1}^n P(x_k | C_i)$$

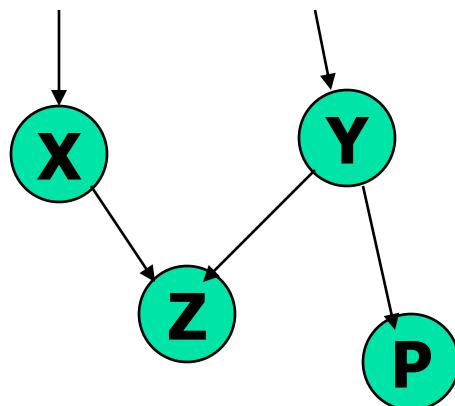
- Ex. Suppose a dataset with 1000 tuples, income=low (0), income=medium (990), and income = high (10),
- Use Laplacian correction (or Laplacian estimator)
 - Adding 1 to each case
 - Prob(income = low) = 1/1003
 - Prob(income = medium) = 991/1003
 - Prob(income = high) = 11/1003
 - The “corrected” prob. estimates are close to their “uncorrected” counterparts

Naïve Bayesian Classifier: Comments

- Advantages
 - Easy to implement
 - Good results obtained in most of the cases
- Disadvantages
 - Assumption: class conditional independence, therefore loss of accuracy
 - Practically, dependencies exist among variables
 - E.g., hospitals: patients: Profile: age, family history, etc.
 - Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
 - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
 - Bayesian Belief Networks

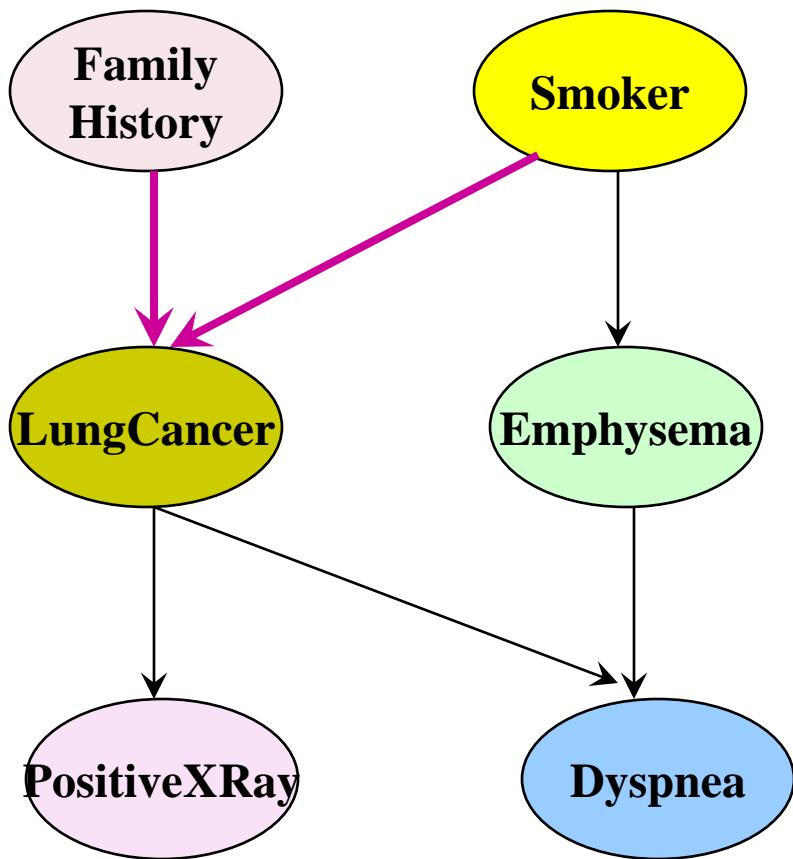
Bayesian Belief Networks

- Bayesian belief network allows a *subset* of the variables conditionally independent
- A graphical model of causal relationships
 - Represents dependency among the variables
 - Gives a specification of joint probability distribution



- Nodes: random variables
- Links: dependency
- X and Y are the parents of Z, and Y is the parent of P
- No dependency between Z and P
- Has no loops or cycles

Bayesian Belief Network: An Example



The **conditional probability table (CPT)** for variable LungCancer:

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

CPT shows the conditional probability for each possible combination of its parents

Derivation of the probability of a particular combination of values of \mathbf{X} , from CPT:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | Parents(Y_i))$$

Bayesian Belief Networks

Training Bayesian Networks

- Several scenarios:
 - Given both the network structure and all variables observable: *learn only the CPTs*
 - Network structure known, some hidden variables: *gradient descent* (greedy hill-climbing) method, analogous to neural network learning
 - Network structure unknown, all variables observable: search through the model space to *reconstruct network topology*
 - Unknown structure, all hidden variables: No good algorithms known for this purpose
- Ref. D. Heckerman: Bayesian networks for data mining

Discriminative Classifiers

- Advantages
 - prediction accuracy is generally high
 - As compared to Bayesian methods – in general
 - robust, works when training examples contain errors
 - fast evaluation of the learned target function
 - Bayesian networks are normally slow
- Criticism
 - long training time
 - difficult to understand the learned function (weights)
 - Bayesian networks can be used easily for pattern discovery
 - not easy to incorporate domain knowledge
 - Easy in the form of priors on the data or distributions

Neural Network as a Classifier

■ Weakness

- Long training time
- Require a number of parameters typically best determined empirically, e.g., the network topology or ``structure."
- Poor interpretability: Difficult to interpret the symbolic meaning behind the learned weights and of ``hidden units" in the network

■ Strength

- High tolerance to noisy data
- Ability to classify untrained patterns
- Well-suited for continuous-valued inputs and outputs
- Successful on a wide array of real-world data
- Algorithms are inherently parallel
- Techniques have recently been developed for the extraction of rules from trained neural networks

SVM—Support Vector Machines

- A new classification method for both linear and nonlinear data
- It uses a nonlinear mapping to transform the original training data into a higher dimension
- With the new dimension, it searches for the linear optimal separating hyperplane (i.e., “decision boundary”)
- With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane
- SVM finds this hyperplane using support vectors (“essential” training tuples) and margins (defined by the support vectors)

Why Is SVM Effective on High Dimensional Data?

- The complexity of trained classifier is characterized by the # of support vectors rather than the dimensionality of the data
- The support vectors are the essential or critical training examples — they lie closest to the decision boundary (MMH)
- If all other training examples are removed and the training is repeated, the same separating hyperplane would be found
- The number of support vectors found can be used to compute an (upper) bound on the expected error rate of the SVM classifier, which is independent of the data dimensionality
- Thus, an SVM with a small number of support vectors can have good generalization, even when the dimensionality of the data is high

SVM vs. Neural Network

- SVM
 - Relatively new concept
 - Deterministic algorithm
 - Nice Generalization properties
 - Hard to learn – learned in batch mode using quadratic programming techniques
 - Using kernels can learn very complex functions
- Neural Network
 - Relatively old
 - Nondeterministic algorithm
 - Generalizes well but doesn't have strong mathematical foundation
 - Can easily be learned in incremental fashion
 - To learn complex functions—use multilayer perceptron (not that trivial)

Lazy vs. Eager Learning

- Lazy vs. eager learning
 - Lazy learning (e.g., instance-based learning): Simply stores training data (or only minor processing) and waits until it is given a test tuple
 - Eager learning (the above discussed methods): Given a set of training set, constructs a classification model before receiving new (e.g., test) data to classify
- Lazy: less time in training but more time in predicting
- Accuracy
 - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
 - Eager: must commit to a single hypothesis that covers the entire instance space

Lazy Learner: Instance-Based Methods

- Instance-based learning:
 - Store training examples and delay the processing ("lazy evaluation") until a new instance must be classified
- Typical approaches
 - k -nearest neighbor approach
 - Instances represented as points in a Euclidean space.
 - Locally weighted regression
 - Constructs local approximation
 - Case-based reasoning
 - Uses symbolic representations and knowledge-based inference

Thank You!

Clustering

Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University

Cluster Analysis

1. What is Cluster Analysis? 
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary

What is Cluster Analysis?

- Cluster: a collection of data objects
 - Similar to one another within the same cluster
 - Dissimilar to the objects in other clusters
- Cluster analysis
 - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- **Unsupervised learning**: no predefined classes
- Typical applications
 - As a **stand-alone tool** to get insight into data distribution
 - As a **preprocessing step** for other algorithms

Clustering: Rich Applications and Multidisciplinary Efforts

- Pattern Recognition
- Spatial Data Analysis
 - Create thematic maps in GIS by clustering feature spaces
 - Detect spatial clusters or for other spatial mining tasks
- Image Processing
- Economic Science (especially market research)
- WWW
 - Document classification
 - Cluster Weblog data to discover groups of similar access patterns

Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earthquake studies: Observed earth quake epicenters should be clustered along continent faults

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters with
 - high intra-class similarity
 - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- **Dissimilarity/Similarity metric:** Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
- There is a separate “quality” function that measures the “goodness” of a cluster.
- The definitions of **distance functions** are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define “similar enough” or “good enough”
 - the answer is typically highly subjective.

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary



Data Structures

- Data matrix
 - (two modes)

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
 - (one mode)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Type of data in clustering analysis

- Interval-scaled variables
- Binary variables
- Nominal, ordinal, and ratio variables
- Variables of mixed types

Interval-valued variables

- Standardize data

- Calculate the mean absolute deviation:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|)$$

where $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \dots + x_{nf})$.

- Calculate the standardized measurement (*z-score*)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q)}$$

where $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ are two p -dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

Similarity and Dissimilarity Between Objects (Cont.)

- If $q = 2$, d is Euclidean distance:

$$d(i, j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \dots + |x_{i_p} - x_{j_p}|^2)}$$

- Properties

- $d(i, j) \geq 0$
- $d(i, i) = 0$
- $d(i, j) = d(j, i)$
- $d(i, j) \leq d(i, k) + d(k, j)$

- Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures

Binary Variables

- A contingency table for binary data

		Object <i>j</i>		
		1	0	<i>sum</i>
Object <i>i</i>	1	<i>a</i>	<i>b</i>	<i>a+b</i>
	0	<i>c</i>	<i>d</i>	<i>c+d</i>
	<i>sum</i>	<i>a+c</i>	<i>b+d</i>	<i>p</i>

- Distance measure for symmetric binary variables:
- Distance measure for asymmetric binary variables:
- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

$$d(i, j) = \frac{b + c}{a + b + c + d}$$

$$d(i, j) = \frac{b + c}{a + b + c}$$

$$\text{sim}_{\text{Jaccard}}(i, j) = \frac{a}{a + b + c}$$

Dissimilarity between Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- gender is a symmetric attribute
- the remaining attributes are asymmetric binary
- let the values Y and P be set to 1, and the value N be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
 - m : # of matches, p : total # of variables

$$d(i, j) = \frac{p - m}{p}$$

- Method 2: use a large number of binary variables
 - creating a new binary variable for each of the M nominal states

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank
- Can be treated like interval-scaled
 - replace x_{if} by their rank $r_{if} \in \{1, \dots, M_f\}$
 - map the range of each variable onto [0, 1] by replacing i -th object in the f -th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- compute the dissimilarity using methods for interval-scaled variables

Ratio-Scaled Variables

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt}
- Methods:
 - treat them like interval-scaled variables—*not a good choice!* (why?—the scale can be distorted)
 - apply logarithmic transformation
$$y_{if} = \log(x_{if})$$
 - treat them as continuous ordinal data treat their rank as interval-scaled

Variables of Mixed Types

- A database may contain all the six types of variables
 - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio
- One may use a weighted formula to combine their effects

$$d(i, j) = \frac{\sum_{f=1}^P \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^P \delta_{ij}^{(f)}}$$

- f is binary or nominal:
 $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise
- f is interval-based: use the normalized distance
- f is ordinal or ratio-scaled
 - compute ranks r_{if} and
 - and treat z_{if} as interval-scaled

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

Vector Objects

- Vector objects: keywords in documents, gene features in micro-arrays, etc.
- Broad applications: information retrieval, biologic taxonomy, etc.
- Cosine measure

$$s(\vec{X}, \vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{|\vec{X}| |\vec{Y}|},$$

\vec{X}^t is a transposition of vector \vec{X} , $|\vec{X}|$ is the Euclidean norm of vector \vec{X} ,

- A variant: Tanimoto coefficient

$$s(\vec{X}, \vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{\vec{X}^t \cdot \vec{X} + \vec{Y}^t \cdot \vec{Y} - \vec{X}^t \cdot \vec{Y}},$$

Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary



Major Clustering Approaches (I)

- Partitioning approach:
 - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
 - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
 - Create a hierarchical decomposition of the set of data (or objects) using some criterion
 - Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON
- Density-based approach:
 - Based on connectivity and density functions
 - Typical methods: DBSCAN, OPTICS, DenClue

Major Clustering Approaches (II)

- Grid-based approach:
 - based on a multiple-level granularity structure
 - Typical methods: STING, WaveCluster, CLIQUE
- Model-based:
 - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
 - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
 - Based on the analysis of frequent patterns
 - Typical methods: pCluster
- User-guided or constraint-based:
 - Clustering by considering user-specified or application-specific constraints
 - Typical methods: COD (obstacles), constrained clustering

Typical Alternatives to Calculate the Distance between Clusters

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $\text{dis}(K_i, K_j) = \min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $\text{dis}(K_i, K_j) = \max(t_{ip}, t_{jq})$
- Average: avg distance between an element in one cluster and an element in the other, i.e., $\text{dis}(K_i, K_j) = \text{avg}(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters, i.e., $\text{dis}(K_i, K_j) = \text{dis}(C_i, C_j)$
- Medoid: distance between the medoids of two clusters, i.e., $\text{dis}(K_i, K_j) = \text{dis}(M_i, M_j)$
 - Medoid: one chosen, centrally located object in the cluster

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: the “middle” of a cluster

$$C_m = \frac{\sum_{i=1}^N (t_{ip})}{N}$$

- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^N (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^N (t_{ip} - t_{iq})^2}{N(N-1)}}$$

Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary



Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database \mathcal{D} of n objects into a set of k clusters, s.t., min sum of squared distance

$$\sum_{m=1}^k \sum_{t_{mi} \in K_m} (C_m - t_{mi})^2$$

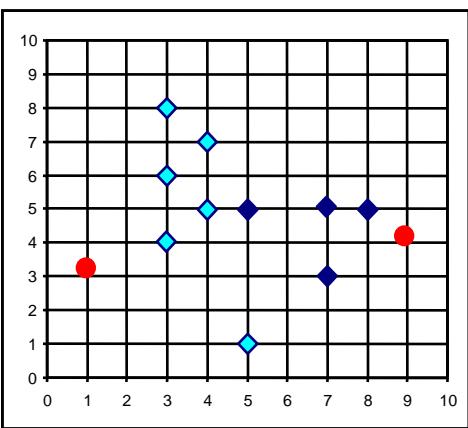
- Given a k , find a partition of k *clusters* that optimizes the chosen partitioning criterion
 - Global optimal: exhaustively enumerate all partitions
 - Heuristic methods: *k-means* and *k-medoids* algorithms
 - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
 - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

- Given k , the *k-means* algorithm is implemented in four steps:
 - Partition objects into k nonempty subsets
 - Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
 - Assign each object to the cluster with the nearest seed point
 - Go back to Step 2, stop when no more new assignment

The *K*-Means Clustering Method

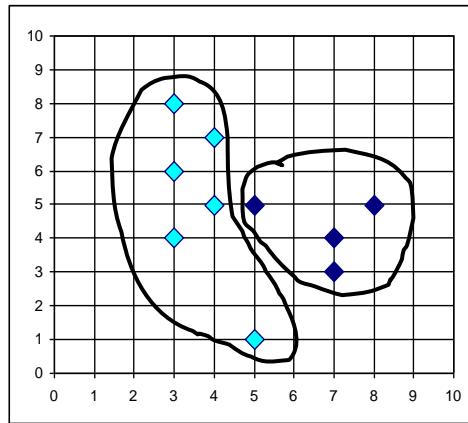
■ Example



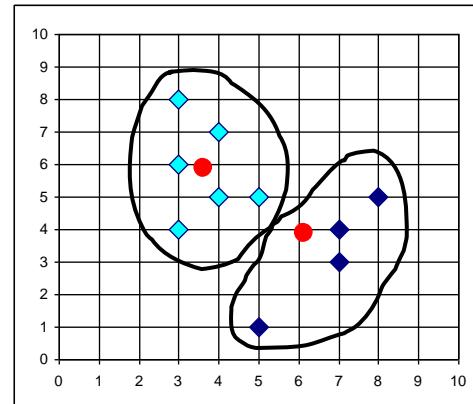
K=2

Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

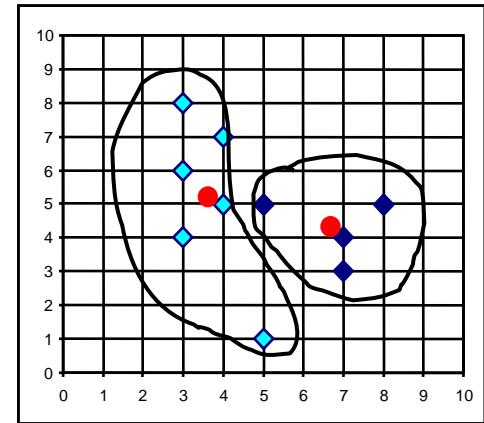


Update the cluster means

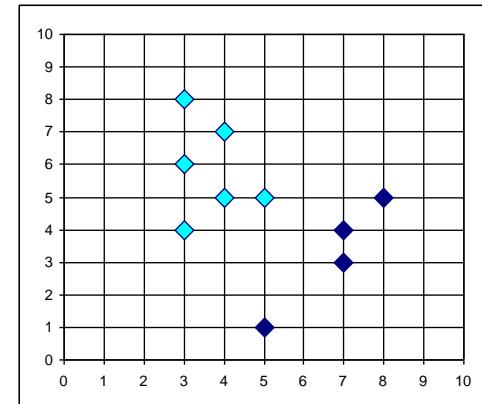


Update the cluster means

reassign



reassign



Comments on the *K-Means* Method

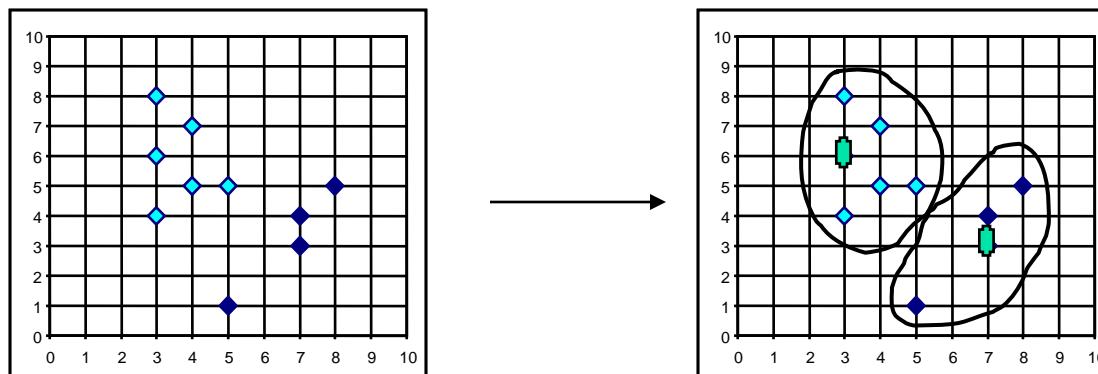
- Strength: *Relatively efficient*: $\mathcal{O}(t kn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.
 - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
 - Applicable only when *mean* is defined, then what about categorical data?
 - Need to specify k , the *number* of clusters, in advance
 - Unable to handle noisy data and *outliers*
 - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
 - Selection of the initial k means
 - Dissimilarity calculations
 - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
 - Replacing means of clusters with modes
 - Using new dissimilarity measures to deal with categorical objects
 - Using a frequency-based method to update modes of clusters
 - A mixture of categorical and numerical data: *k-prototype* method

What Is the Problem of the K-Means Method?

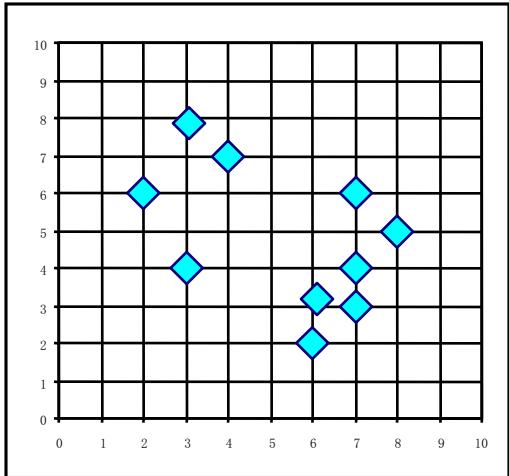
- The k-means algorithm is sensitive to outliers !
 - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
 - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
 - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)

A Typical K-Medoids Algorithm (PAM)

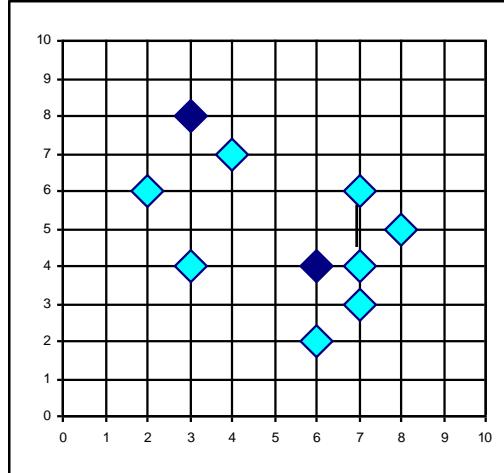


K=2

Do loop
Until no change

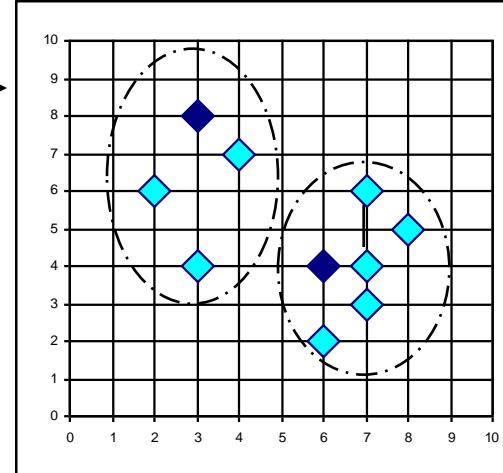
Swapping O and O_{random}
If quality is improved.

Arbitrary choose k object as initial medoids



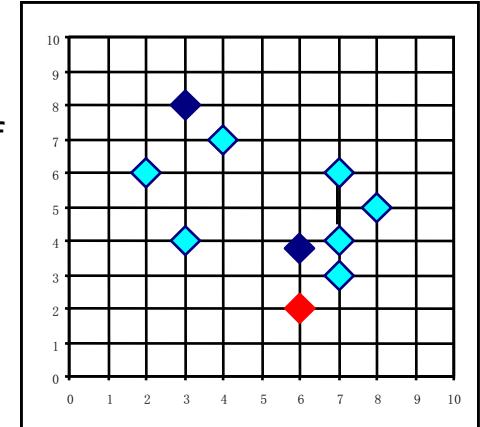
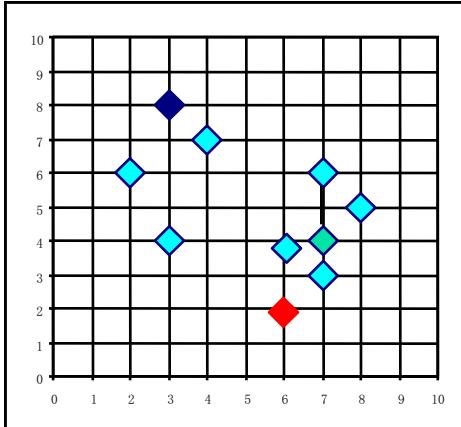
Total Cost = 26

Assign each remaining object to nearest medoids



Total Cost = 20
Randomly select a nonmedoid object, O_{random}

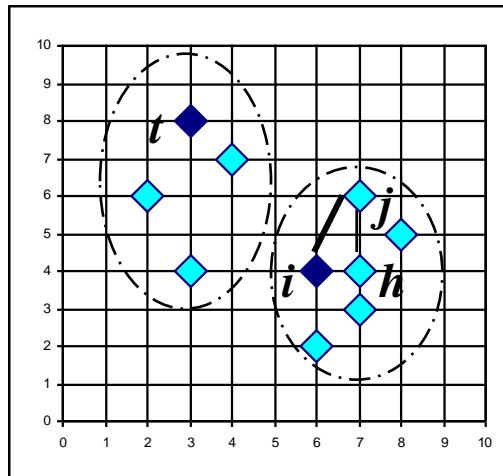
Compute total cost of swapping



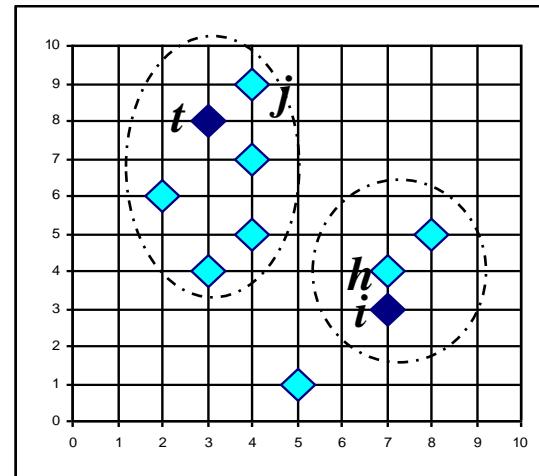
PAM (Partitioning Around Medoids) (1987)

- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
 - Select k representative objects arbitrarily
 - For each pair of non-selected object h and selected object i , calculate the total swapping cost TC_{ih}
 - For each pair of i and h ,
 - If $TC_{ih} < 0$, i is replaced by h
 - Then assign each non-selected object to the most similar representative object
 - repeat steps 2-3 until there is no change

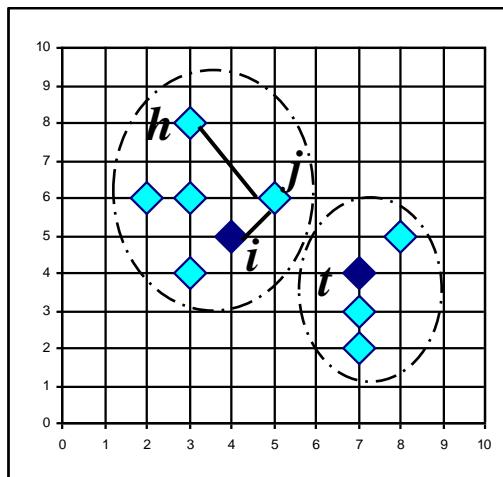
PAM Clustering: Total swapping cost $TC_{ih} = \sum_j C_{jih}$



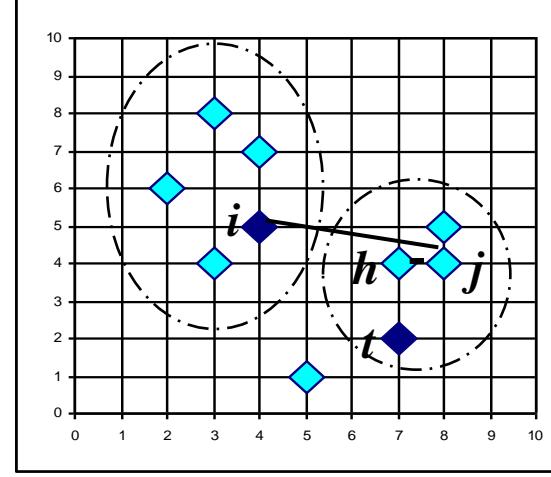
$$C_{jih} = d(j, h) - d(j, i)$$



$$C_{jih} = 0$$



$$C_{jih} = d(j, t) - d(j, i)$$



$$C_{jih} = d(j, h) - d(j, t)$$

What Is the Problem with PAM?

- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
 - $O(k(n-k)^2)$ for each iteration

where n is # of data, k is # of clusters

→ Sampling based method,

CLARA(Clustering LARge Applications)

CLARA (Clustering Large Applications) (1990)

- *CLARA* (Kaufmann and Rousseeuw in 1990)
 - Built in statistical analysis packages, such as S+
- It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
 - Efficiency depends on the sample size
 - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

CLARANS ("Randomized" CLARA) (1994)

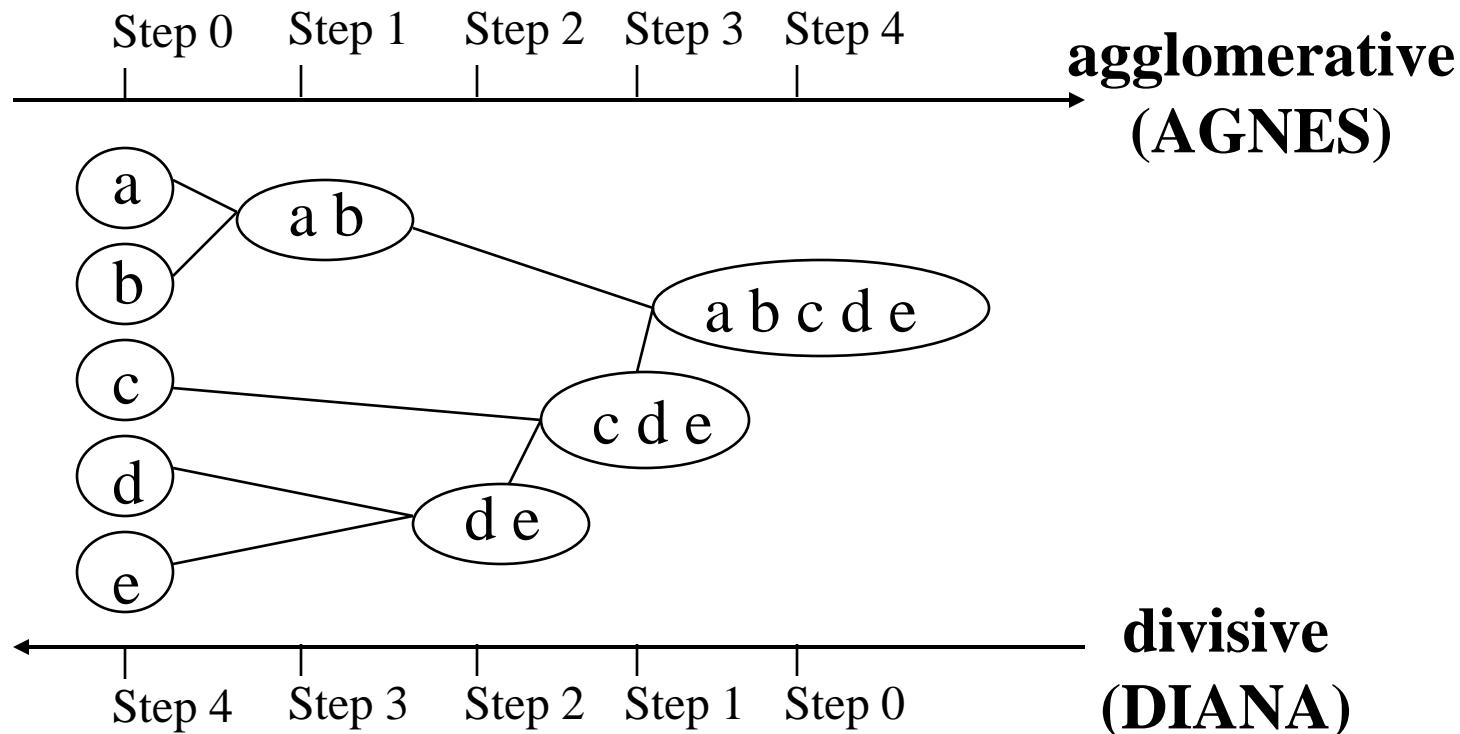
- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
- CLARANS draws sample of neighbors dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of k medoids
- If the local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum
- It is more efficient and scalable than both *PAM* and *CLARA*
- Focusing techniques and spatial access structures may further improve its performance (Ester et al.'95)

Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary

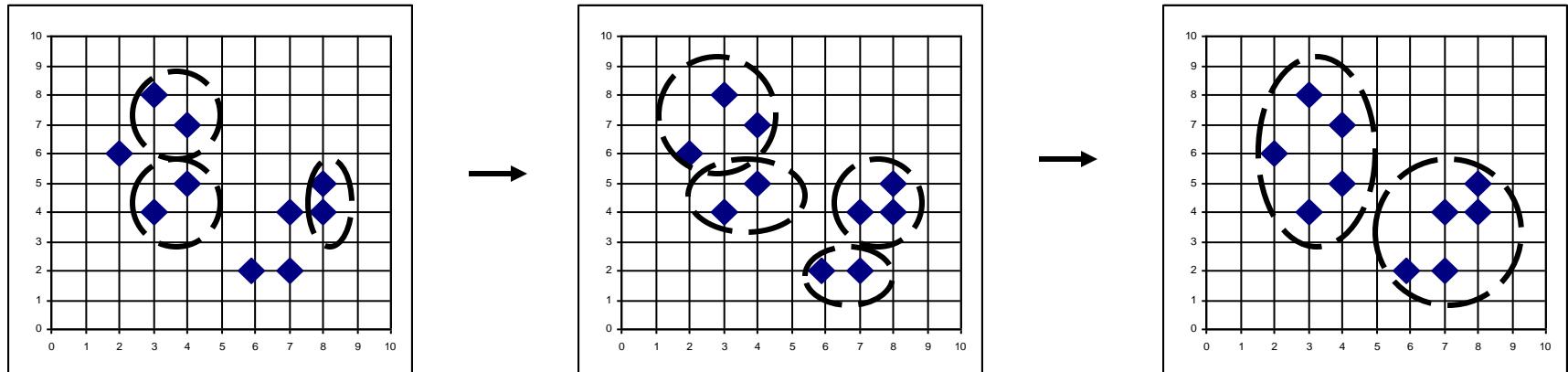
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters k as an input, but needs a termination condition



AGNES (Agglomerative Nesting)

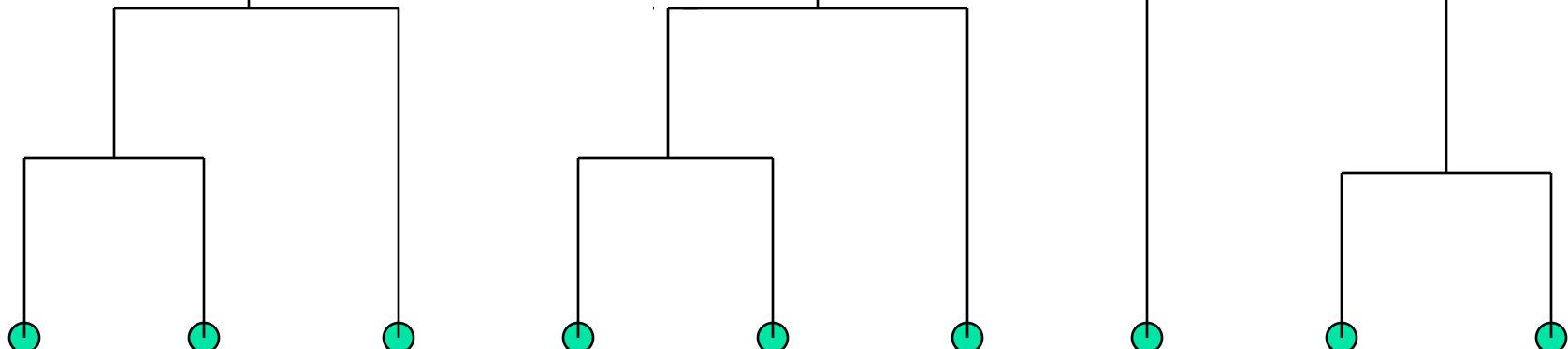
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How the Clusters are Merged

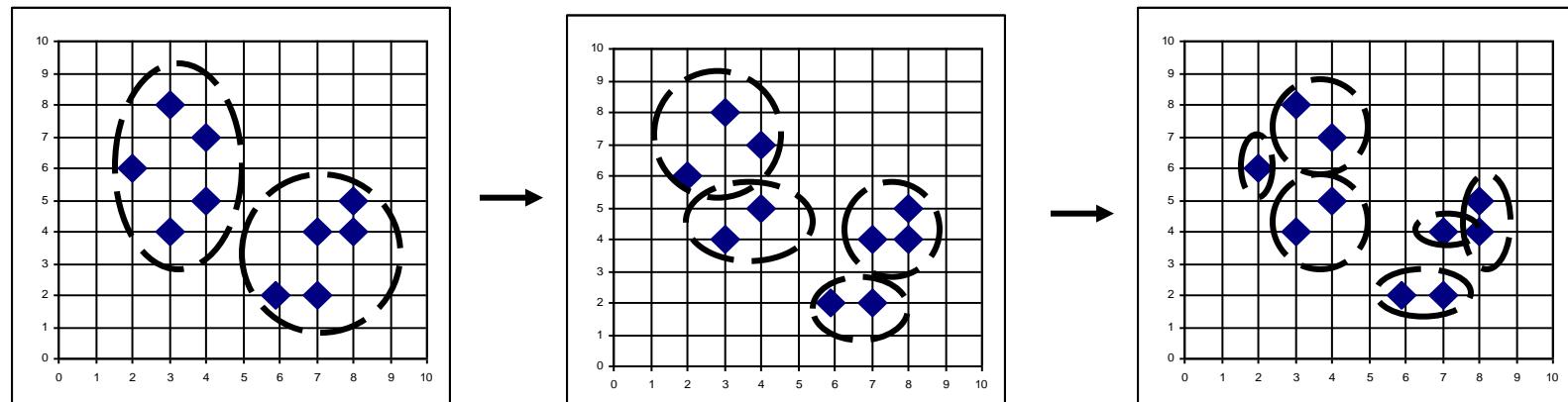
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Recent Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
 - do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
 - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
 - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
 - ROCK (1999): clustering categorical data by neighbor and link analysis
 - CHAMELEON (1999): hierarchical clustering using dynamic modeling

BIRCH (1996)

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies (Zhang, Ramakrishnan & Livny, SIGMOD'96)
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
 - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
 - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness*: handles only numeric data, and sensitive to the order of the data record.

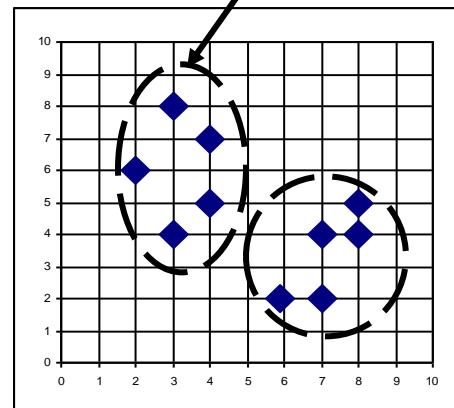
Clustering Feature Vector in BIRCH

Clustering Feature: $CF = (N, \overrightarrow{LS}, SS)$

N: Number of data points

$LS: \sum_{i=1}^N X_i$

$SS: \sum_{i=1}^N X_i^2$



$$CF = (5, (16,30),(54,190))$$

(3,4)

(2,6)

(4,5)

(4,7)

(3,8)

CF-Tree in BIRCH

- Clustering feature:
 - summary of the statistics for a given subcluster: the 0-th, 1st and 2nd moments of the subcluster from the statistical point of view.
 - registers crucial measurements for computing cluster and utilizes storage efficiently
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
 - A nonleaf node in a tree has descendants or “children”
 - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
 - Branching factor: specify the maximum number of children.
 - threshold: max diameter of sub-clusters stored at the leaf nodes

The CF Tree Structure

Root

$B = 7$

$L = 6$

CF_1	CF_2	CF_3	CF_6
$child_1$	$child_2$	$child_3$		$child_6$

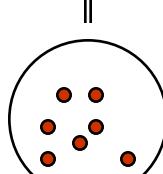
Non-leaf node

CF_1	CF_2	CF_3	CF_5
$child_1$	$child_2$	$child_3$		$child_5$

Leaf node

prev	CF_1	CF_2	CF_6	next
------	--------	--------	-------	--------	------

prev	CF_1	CF_2	CF_4	next
------	--------	--------	-------	--------	------



Clustering Categorical Data: The ROCK Algorithm

- ROCK: RObust Clustering using linKs
 - S. Guha, R. Rastogi & K. Shim, ICDE'99
- Major ideas
 - Use links to measure similarity/proximity
 - Not distance-based
 - Computational complexity: $O(n^2 + nm_m m_a + n^2 \log n)$
- Algorithm: sampling-based clustering
 - Draw random sample
 - Cluster with links
 - Label data in disk
- Experiments
 - Congressional voting, mushroom data

Similarity Measure in ROCK

- Traditional measures for categorical data may not work well, e.g., Jaccard coefficient
- Example: Two groups (clusters) of transactions
 - C₁. <a, b, c, d, e>: {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}
 - C₂. <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}
- Jaccard co-efficient may lead to wrong clustering result
 - C₁: 0.2 ({a, b, c}, {b, d, e}) to 0.5 ({a, b, c}, {a, b, d})
 - C₁ & C₂: could be as high as 0.5 ({a, b, c}, {a, b, f})
- Jaccard co-efficient-based similarity function:
 - Ex. Let T₁ = {a, b, c}, T₂ = {c, d, e}

$$Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

$$Sim(T_1, T_2) = \frac{|\{c\}|}{|\{a, b, c, d, e\}|} = \frac{1}{5} = 0.2$$

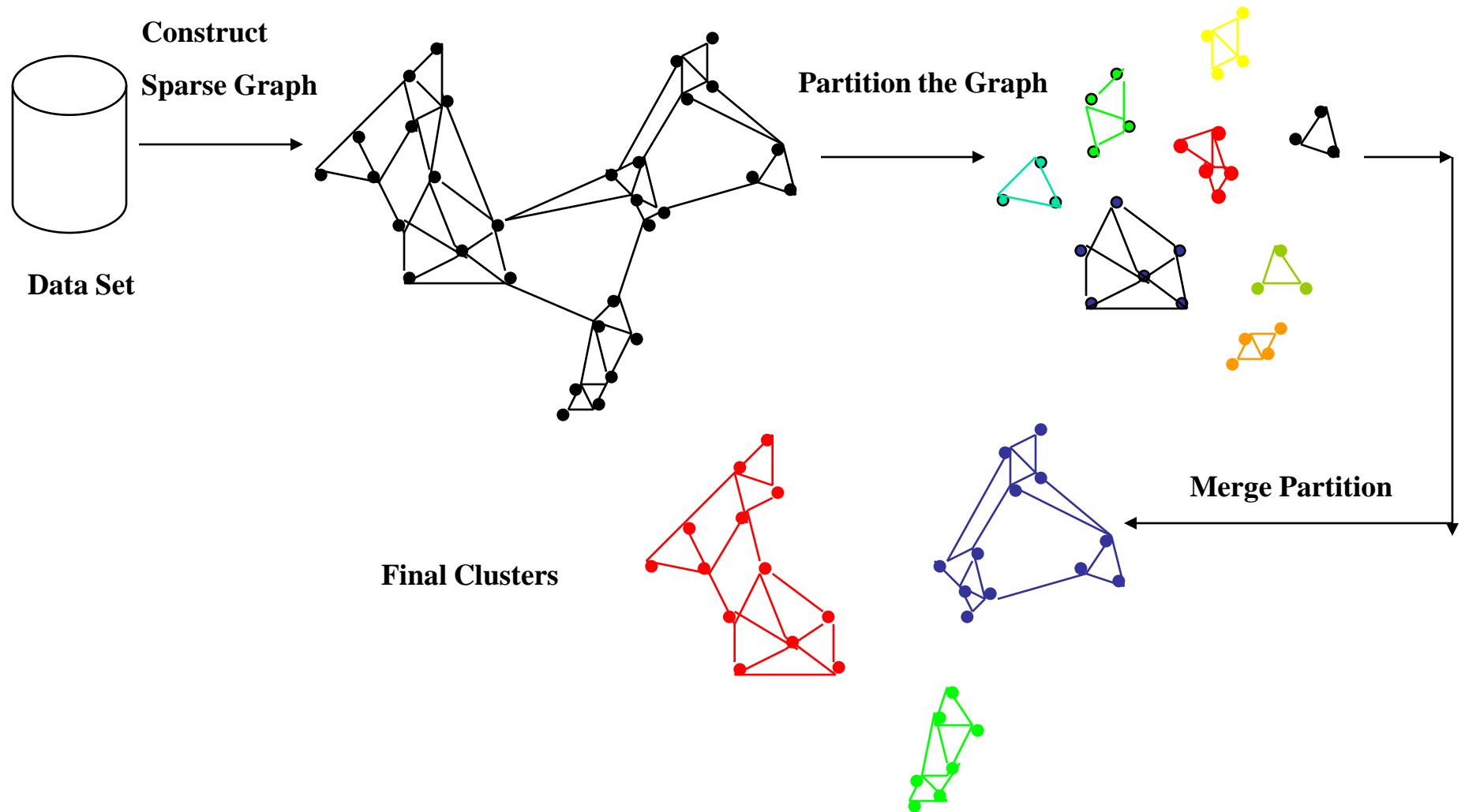
Link Measure in ROCK

- Links: # of common neighbors
 - $C_1 <a, b, c, d, e>$: $\{\underline{a}, \underline{b}, \underline{c}\}$, $\{a, b, d\}$, $\{a, b, e\}$, $\{a, c, d\}$, $\{a, c, e\}$, $\{a, d, e\}$, $\{b, c, d\}$, $\{b, c, e\}$, $\{b, d, e\}$, $\{\underline{c}, \underline{d}, \underline{e}\}$
 - $C_2 <a, b, f, g>$: $\{\underline{a}, \underline{b}, \underline{f}\}$, $\{a, b, g\}$, $\{a, f, g\}$, $\{b, f, g\}$
- Let $T_1 = \{a, b, c\}$, $T_2 = \{c, d, e\}$, $T_3 = \{a, b, f\}$
 - $\text{link}(T_1, T_2) = 4$, since they have 4 common neighbors
 - $\{a, c, d\}$, $\{a, c, e\}$, $\{b, c, d\}$, $\{b, c, e\}$
 - $\text{link}(T_1, T_3) = 3$, since they have 3 common neighbors
 - $\{a, b, d\}$, $\{a, b, e\}$, $\{a, b, g\}$
- Thus link is a better measure than Jaccard coefficient

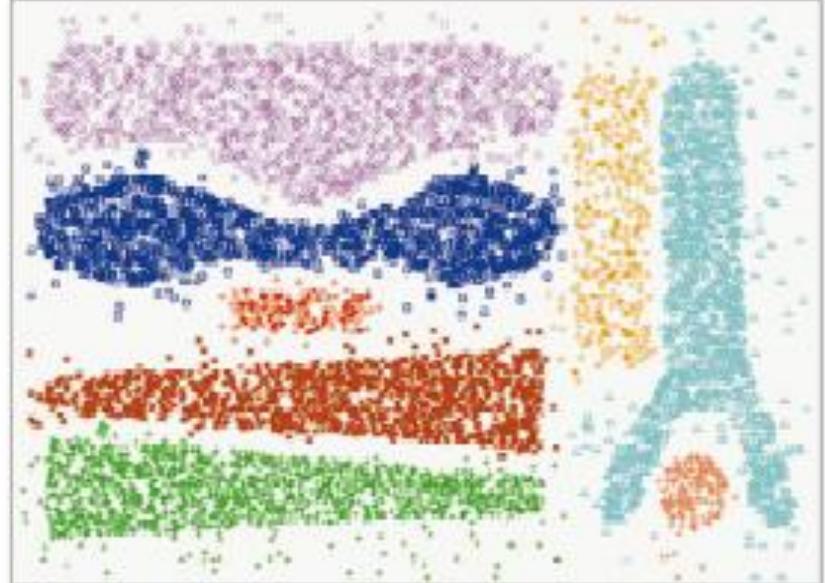
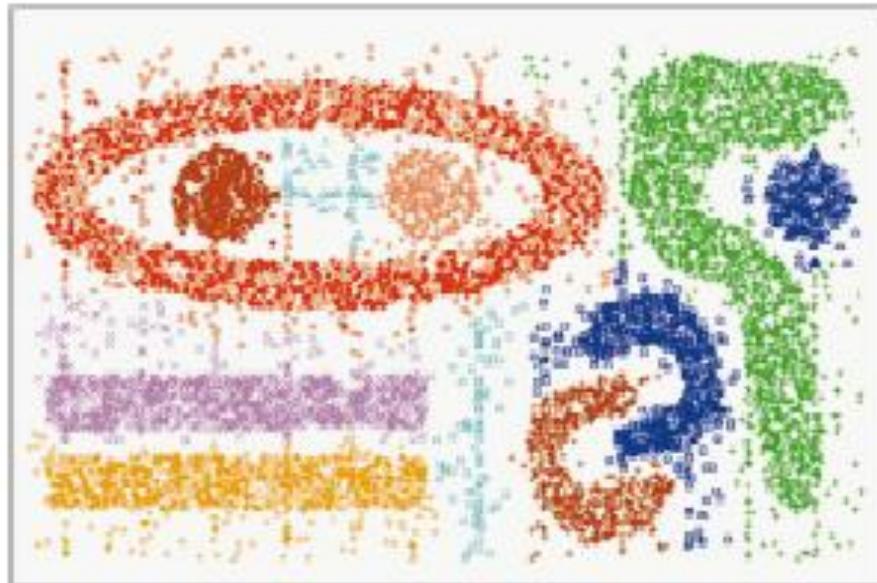
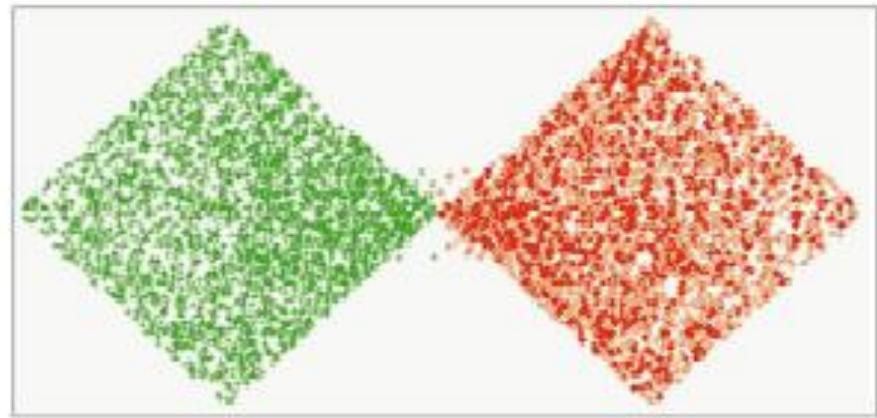
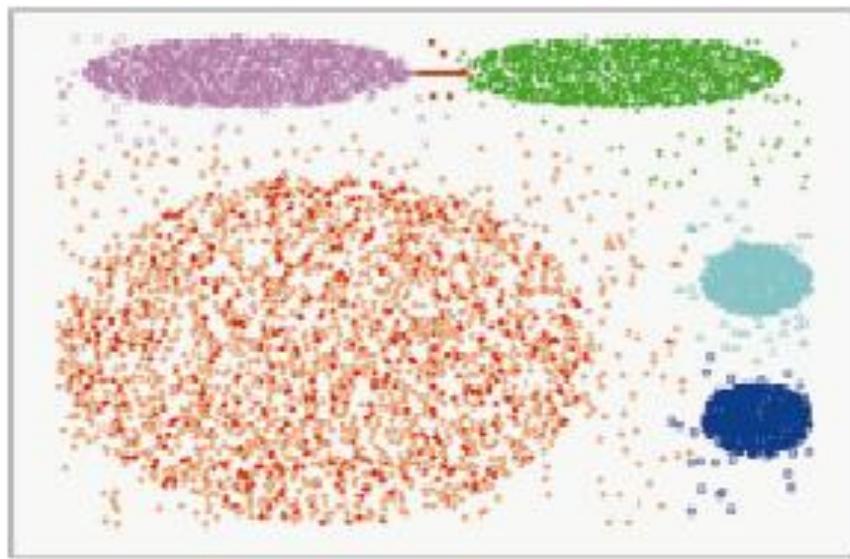
CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)

- CHAMELEON: by G. Karypis, E.H. Han, and V. Kumar'99
- Measures the similarity based on a dynamic model
 - Two clusters are merged only if the *interconnectivity* and *closeness* (*proximity*) between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
 - **Cure** ignores information about **interconnectivity** of the objects, **Rock** ignores information about the **closeness** of two clusters
- A two-phase algorithm
 1. Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
 2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

Overall Framework of CHAMELEON



CHAMELEON (Clustering Complex Objects)



Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods 
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary

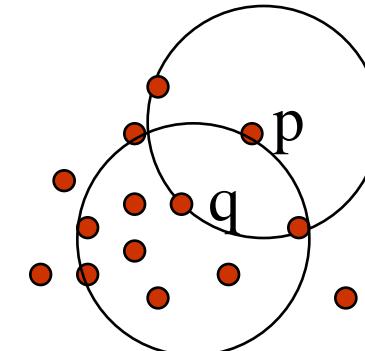
Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
 - Discover clusters of arbitrary shape
 - Handle noise
 - One scan
 - Need density parameters as termination condition
- Several interesting studies:
 - DBSCAN: Ester, et al. (KDD'96)
 - OPTICS: Ankerst, et al (SIGMOD'99).
 - DENCLUE: Hinneburg & D. Keim (KDD'98)
 - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

Density-Based Clustering: Basic Concepts

- Two parameters:
 - *Eps*: Maximum radius of the neighbourhood
 - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid \text{dist}(p,q) \leq Eps\}$
- **Directly density-reachable**: A point p is directly density-reachable from a point q w.r.t. *Eps*, *MinPts* if
 - p belongs to $N_{Eps}(q)$
 - core point condition:

$$|N_{Eps}(q)| \geq MinPts$$



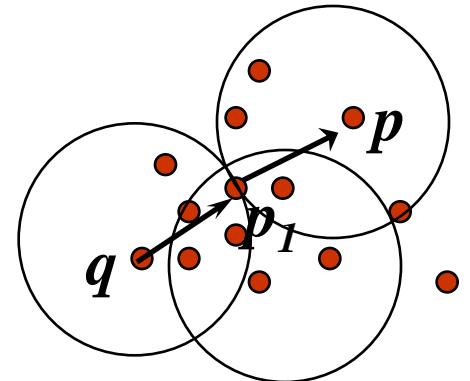
$MinPts = 5$

$Eps = 1 \text{ cm}$

Density-Reachable and Density-Connected

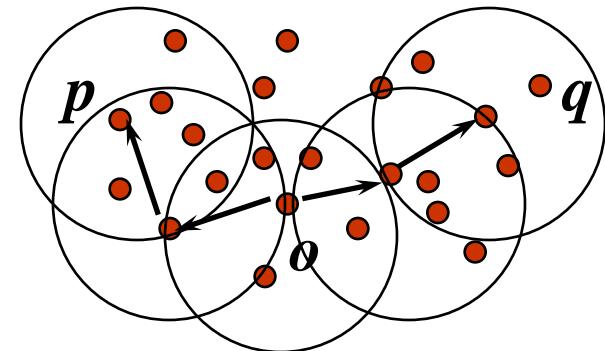
- Density-reachable:

- A point p is **density-reachable** from a point q w.r.t. $Eps, MinPts$ if there is a chain of points p_1, \dots, p_n , $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i .



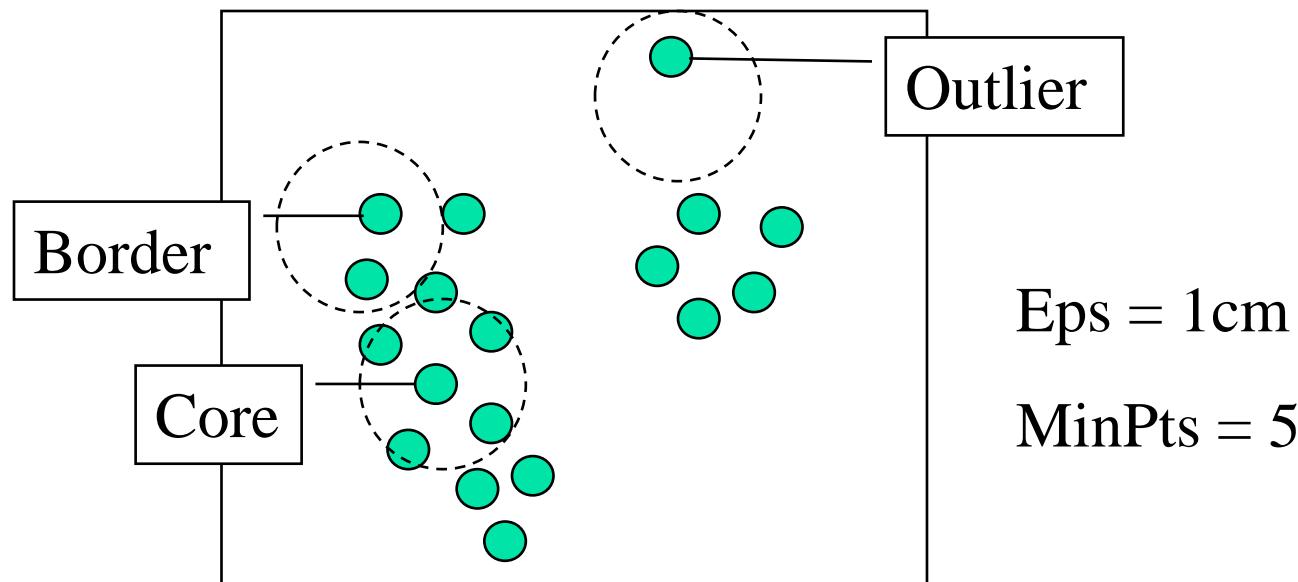
- Density-connected

- A point p is **density-connected** to a point q w.r.t. $Eps, MinPts$ if there is a point o such that both, p and q are density-reachable from o w.r.t. Eps and $MinPts$



DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

- Arbitrary select a point p
- Retrieve all points density-reachable from p w.r.t. Eps and $MinPts$.
- If p is a core point, a cluster is formed.
- If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

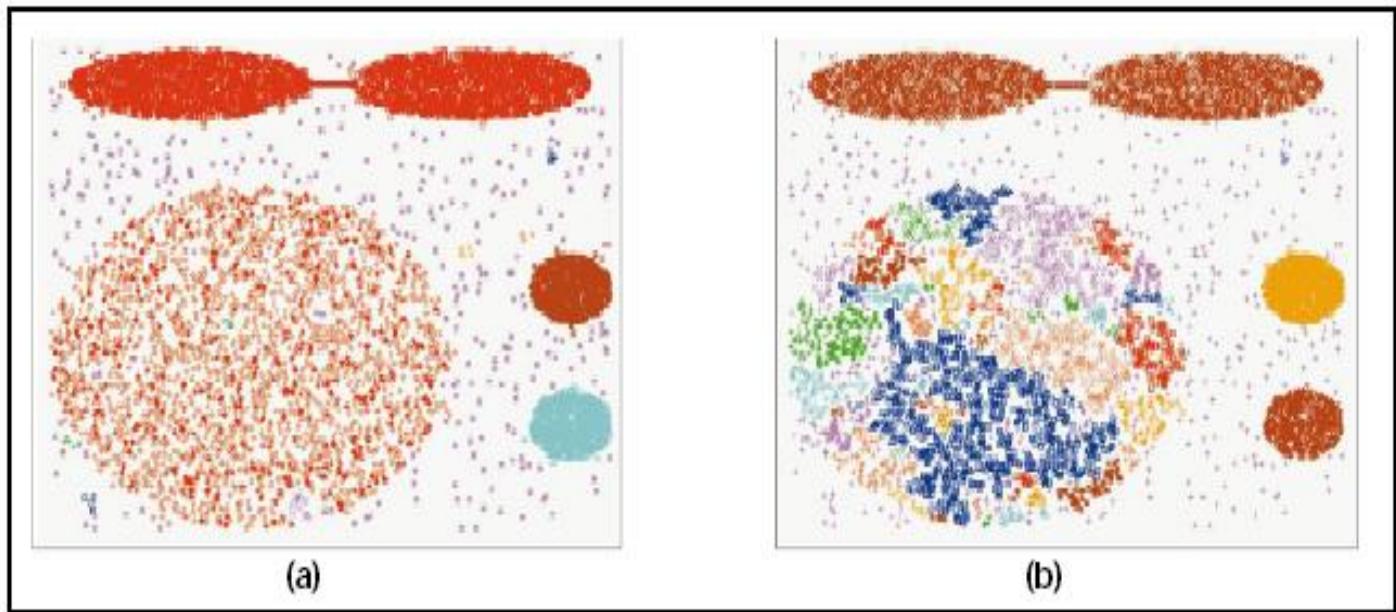
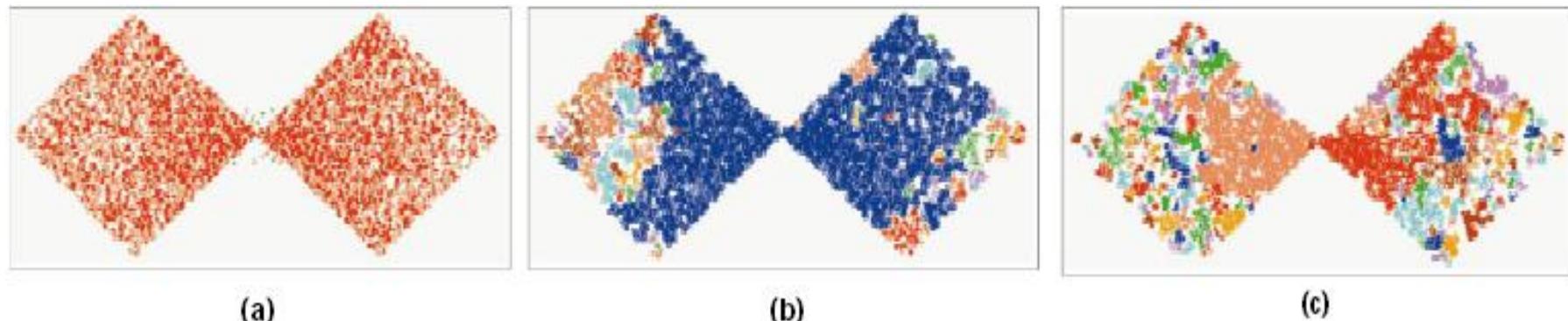
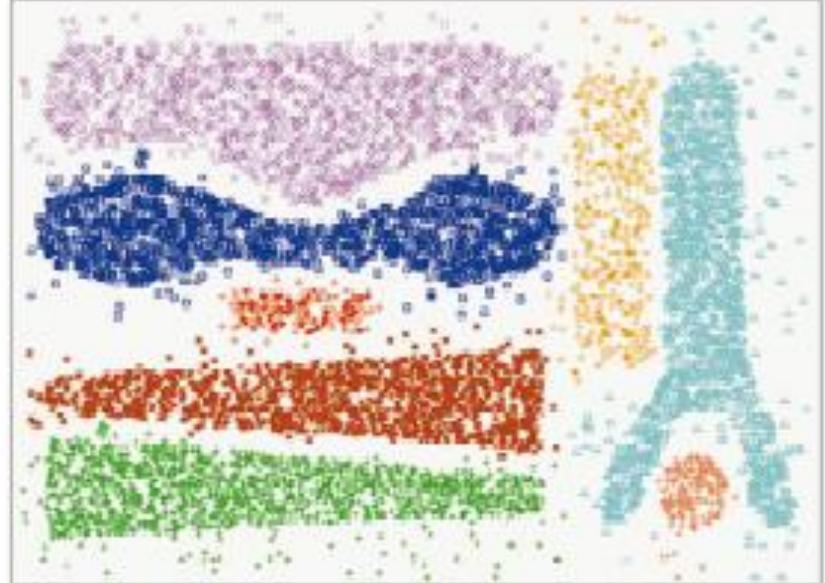
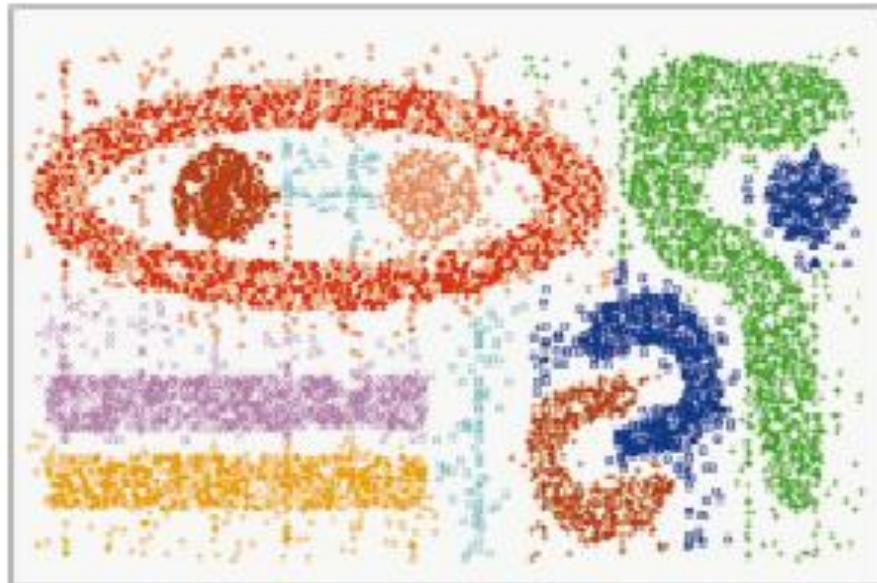
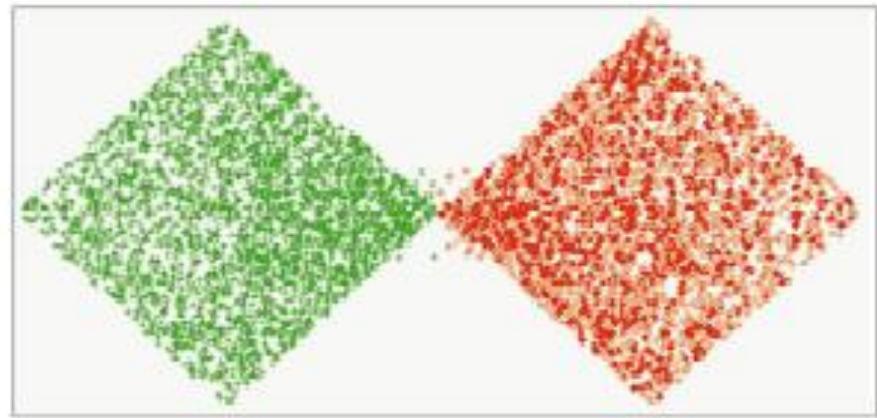
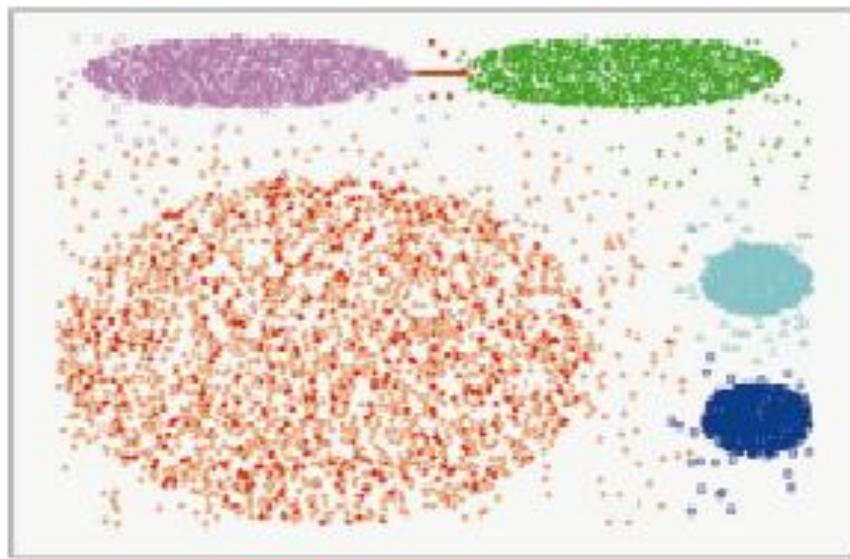


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



CHAMELEON (Clustering Complex Objects)



OPTICS: A Cluster-Ordering Method (1999)

- OPTICS: Ordering Points To Identify the Clustering Structure
 - Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)
 - Produces a special order of the database wrt its density-based clustering structure
 - This cluster-ordering contains info equiv to the density-based clusterings corresponding to a broad range of parameter settings
 - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
 - Can be represented graphically or using visualization techniques

OPTICS: Some Extension from DBSCAN

- Index-based:
 - $k = \text{number of dimensions}$
 - $N = 20$
 - $p = 75\%$
 - $M = N(1-p) = 5$
- Complexity: $O(kN^2)$

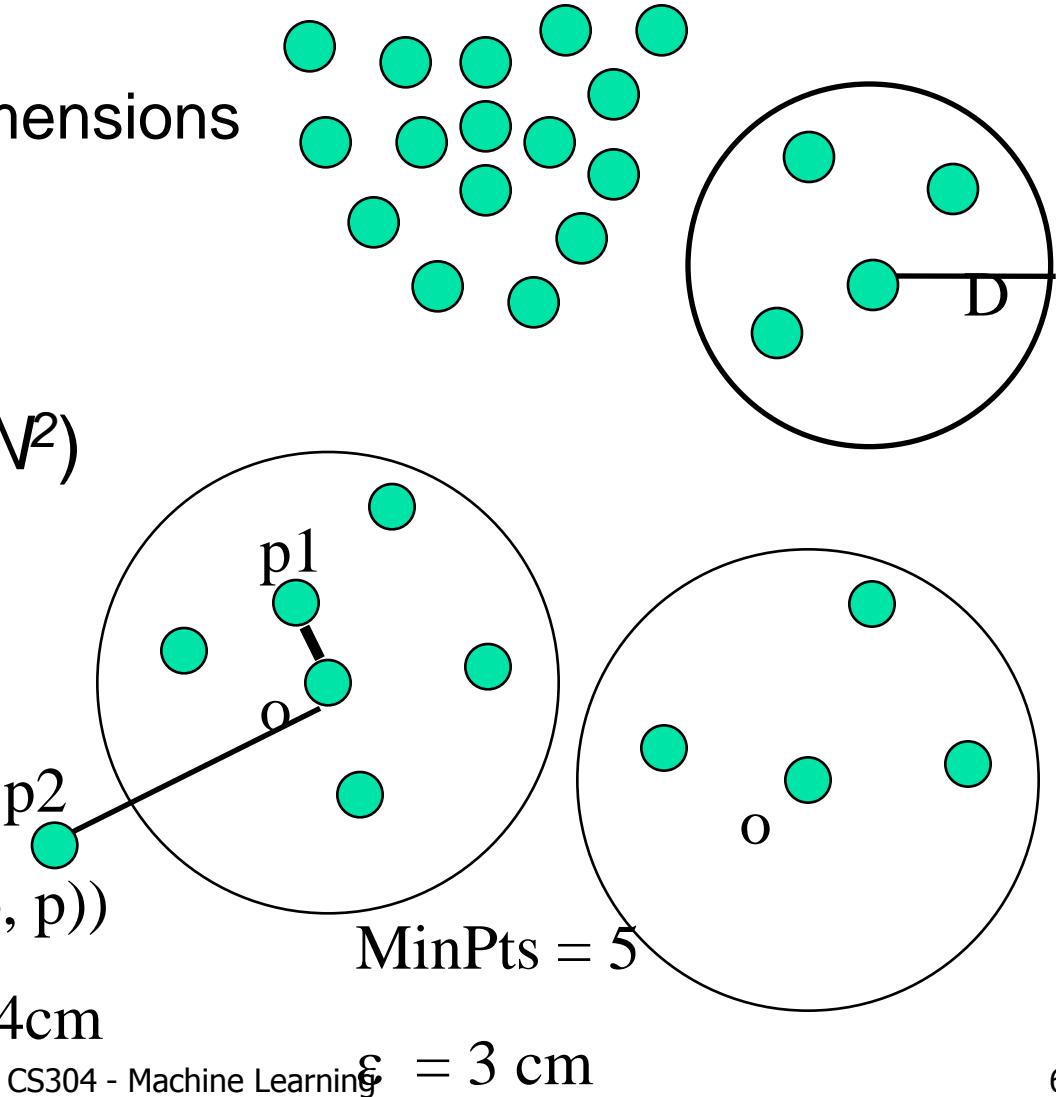
- Core Distance

- Reachability Distance

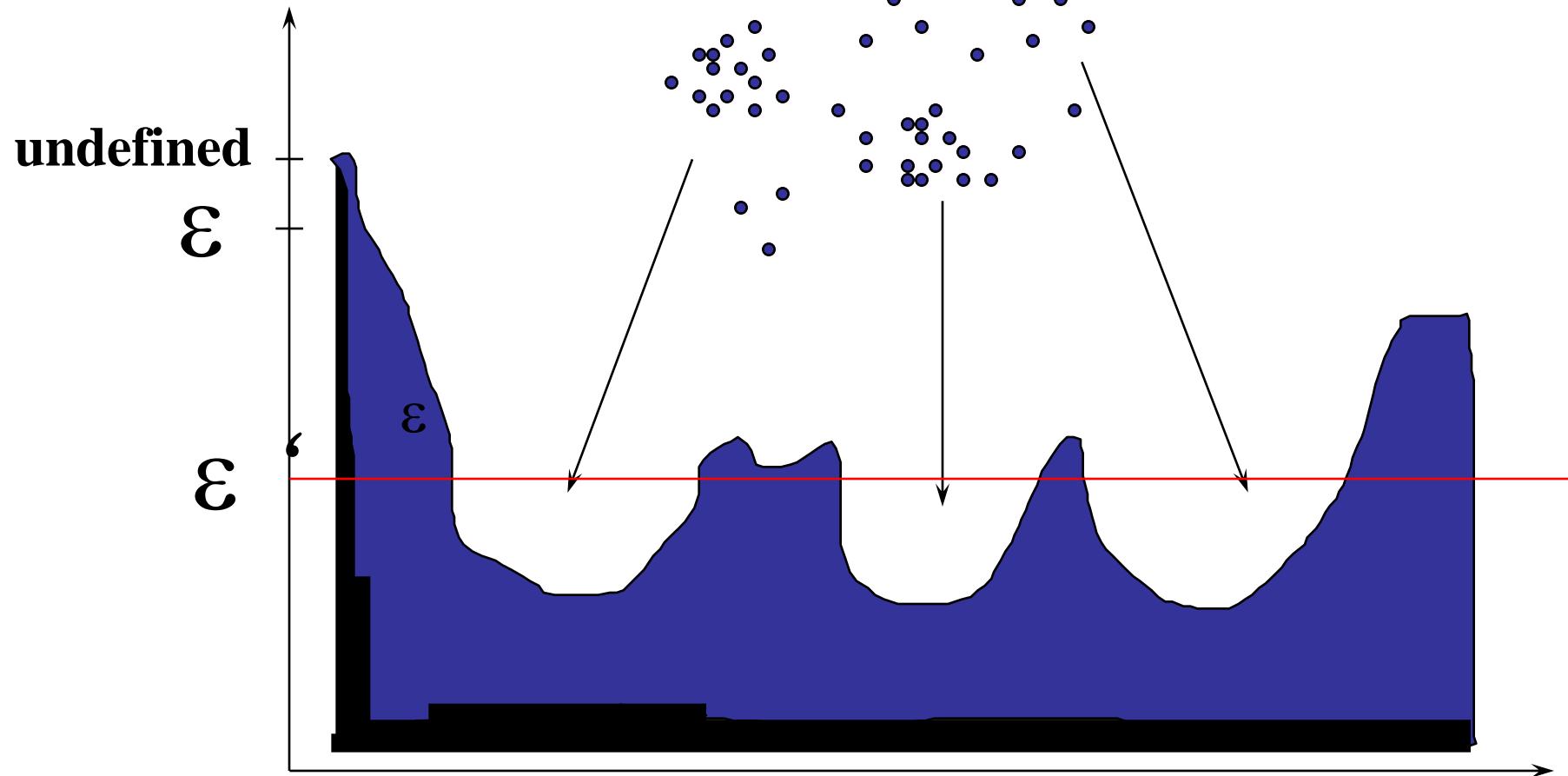
Max (core-distance (o), $d(o, p)$)

$$r(p_1, o) = 2.8\text{cm. } r(p_2, o) = 4\text{cm}$$

$$\epsilon = 3 \text{ cm}$$

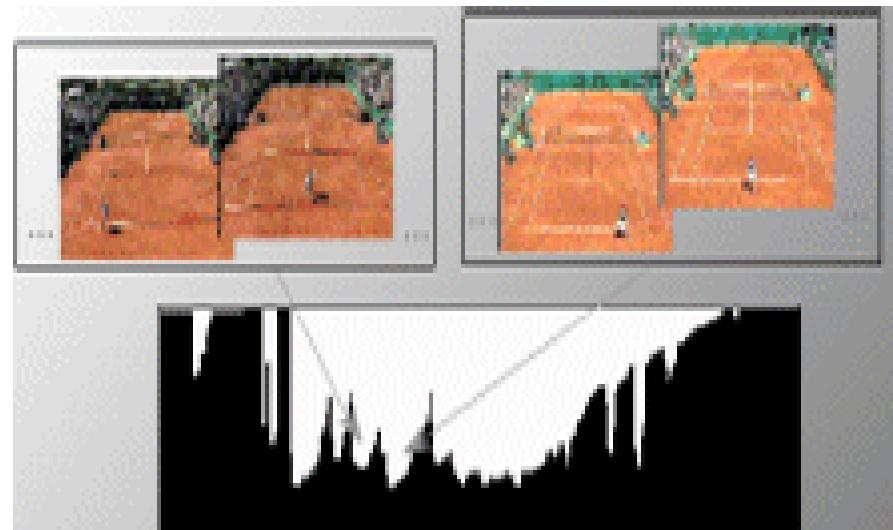
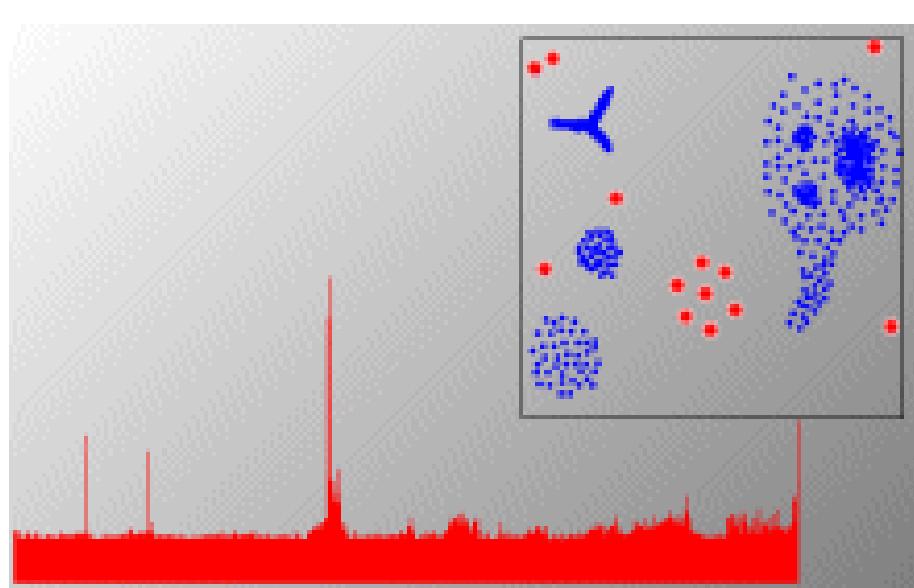
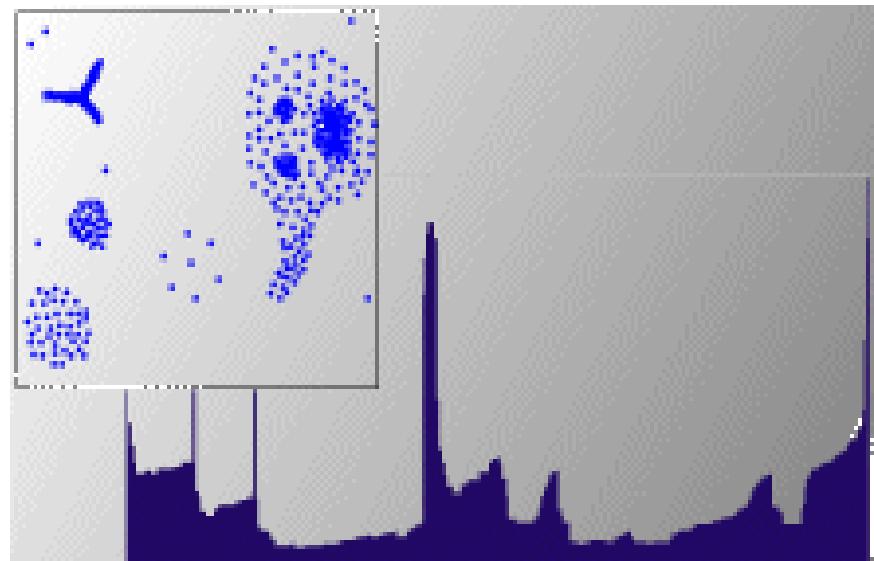
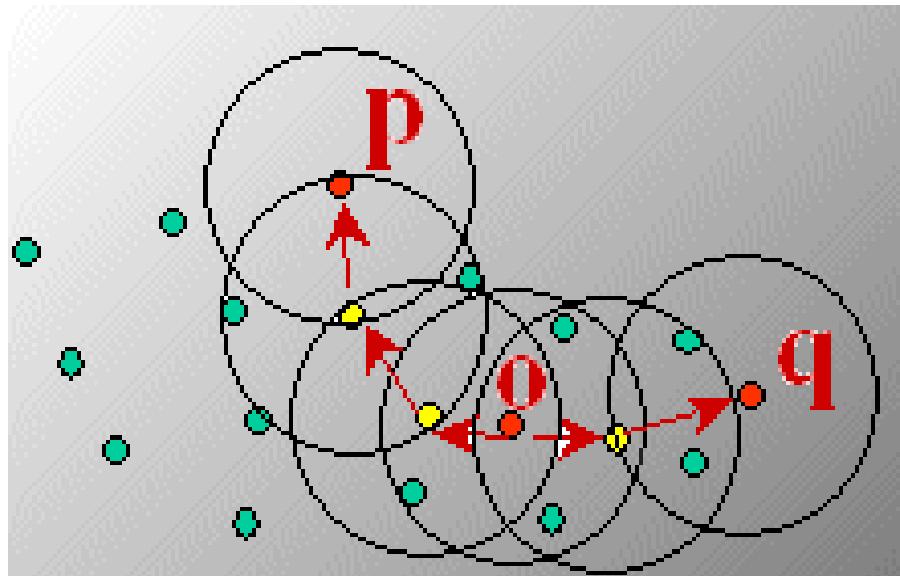


Reachability -distance



Cluster-order
of the objects

Density-Based Clustering: OPTICS & Its Applications



DENCLUE: Using Statistical Density Functions

- DENsity-based CLUstEring by Hinneburg & Keim (KDD'98)
- Using statistical density functions:

$$f_{Gaussian}(x, y) = e^{-\frac{d(x, y)^2}{2\sigma^2}}$$

$$f_{Gaussian}^D(x) = \sum_{i=1}^N e^{-\frac{d(x, x_i)^2}{2\sigma^2}}$$

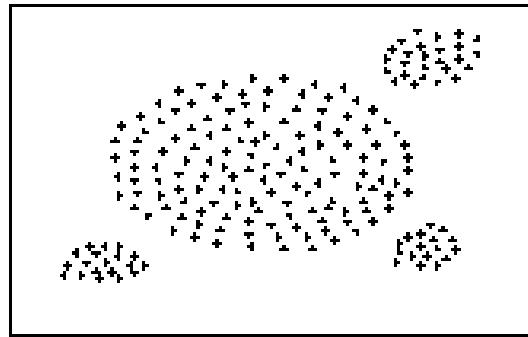
- Major features

- Solid mathematical foundation
- Good for data sets with large amounts of noise
- Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
- Significant faster than existing algorithm (e.g., DBSCAN)
- But needs a large number of parameters

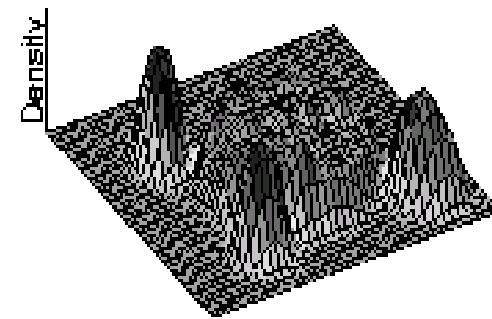
Denclue: Technical Essence

- Uses grid cells but only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure
- Influence function: describes the impact of a data point within its neighborhood
- Overall density of the data space can be calculated as the sum of the influence function of all data points
- Clusters can be determined mathematically by identifying density attractors
- Density attractors are local maximal of the overall density function

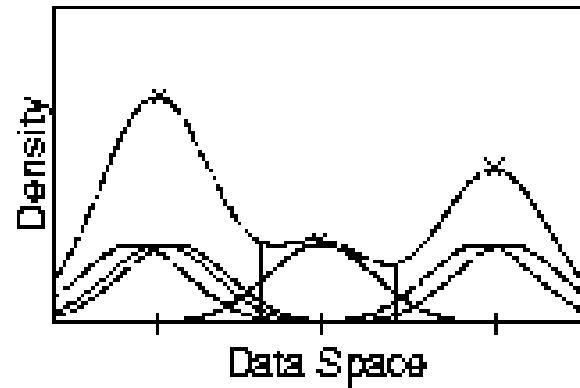
Density Attractor



(a) Data Set



(c) Gaussian



Center-Defined and Arbitrary

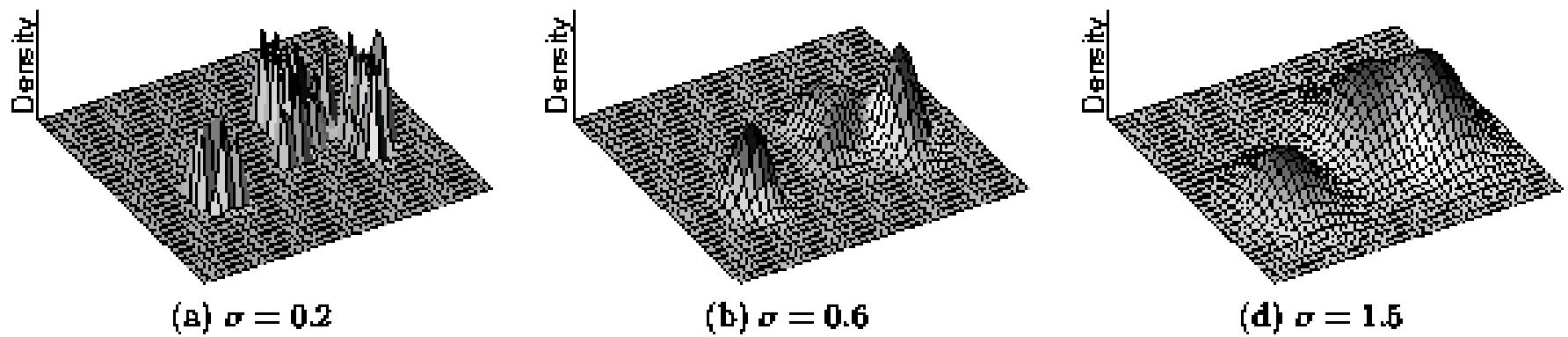


Figure 3: Example of Center-Defined Clusters for different σ

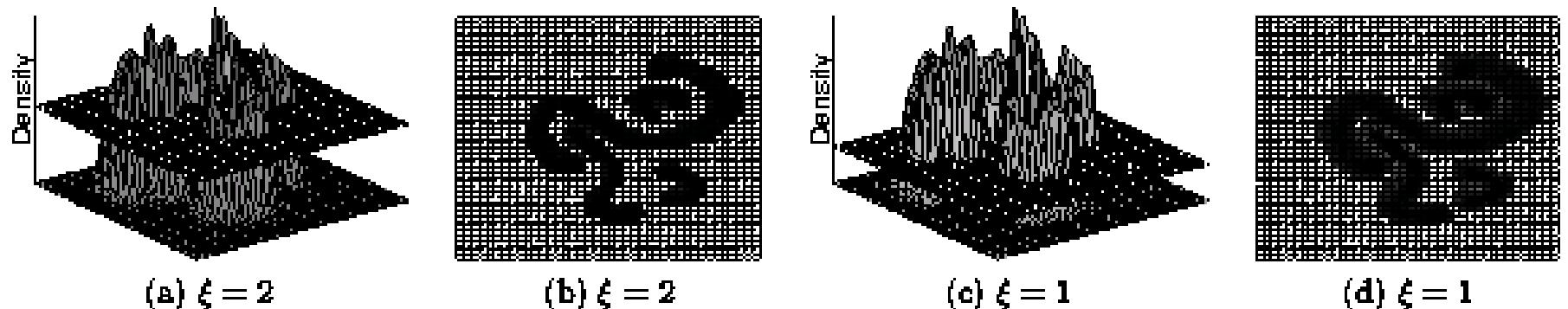


Figure 4: Example of Arbitrary-Shape Clusters for different ξ

Cluster Analysis

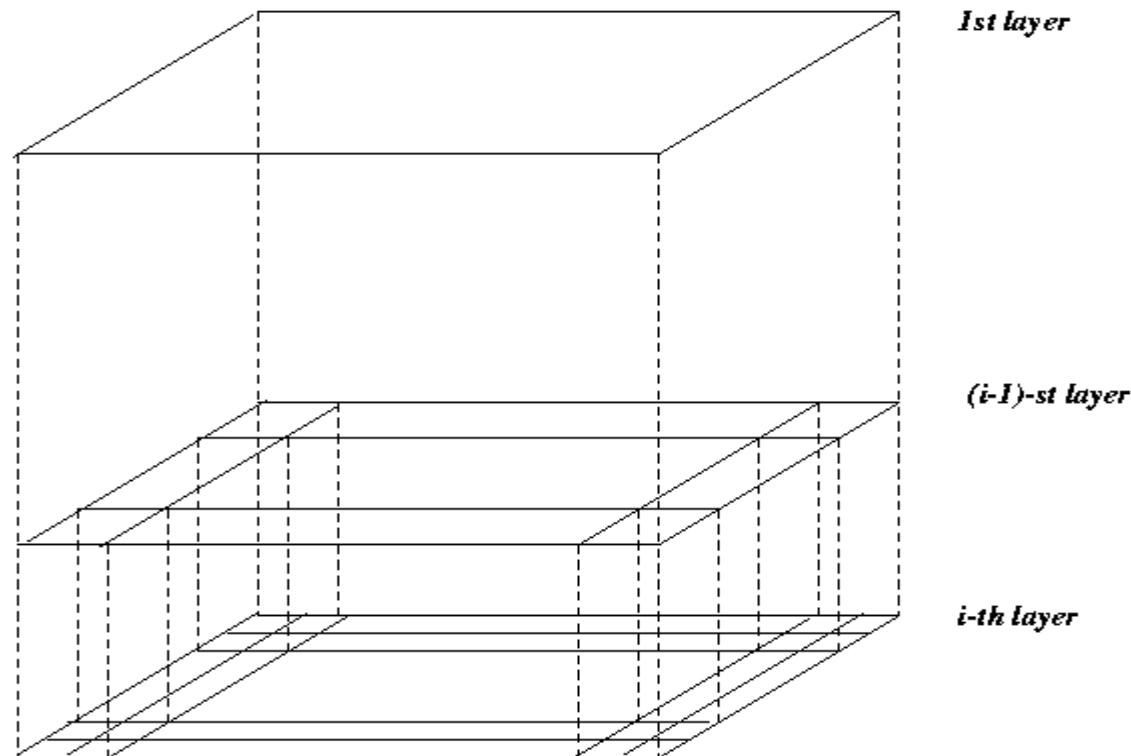
1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary

Grid-Based Clustering Method

- Using multi-resolution grid data structure
- Several interesting methods
 - **STING** (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
 - **WaveCluster** by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
 - A multi-resolution clustering approach using wavelet method
 - **CLIQUE**: Agrawal, et al. (SIGMOD'98)
 - On high-dimensional data (thus put in the section of clustering high-dimensional data)

STING: A Statistical Information Grid Approach

- Wang, Yang and Muntz (VLDB'97)
- The spatial area area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution



The STING Clustering Method

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell
 - *count, mean, s, min, max*
 - type of distribution—normal, *uniform*, etc.
- Use a top-down approach to answer spatial data queries
- Start from a pre-selected layer—typically with a small number of cells
- For each cell in the current level compute the confidence interval

Comments on STING

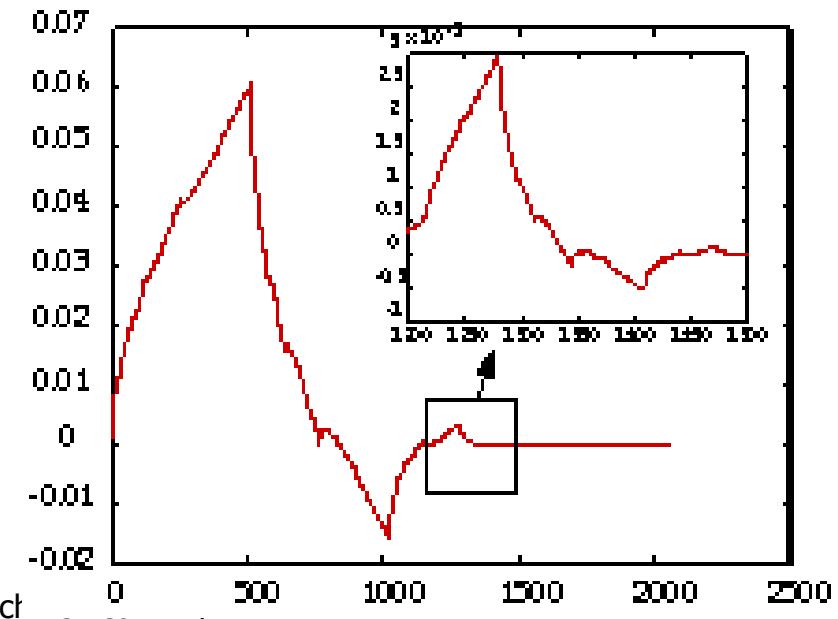
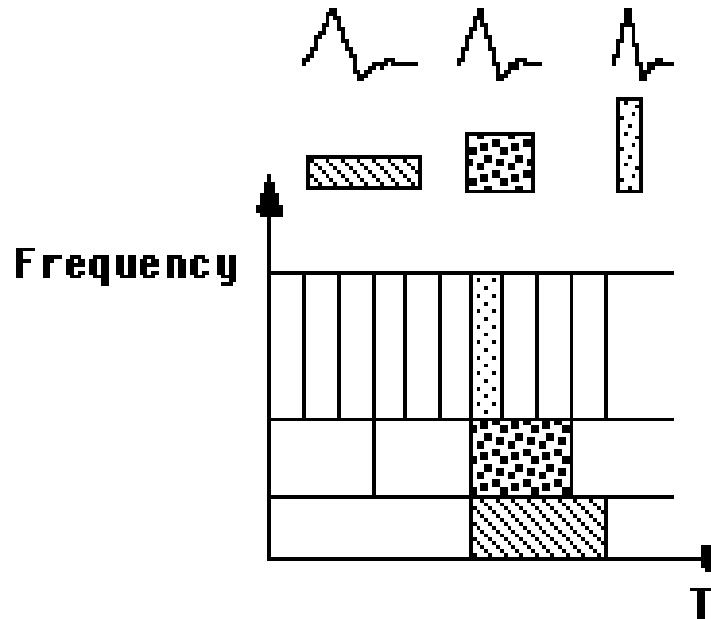
- Remove the irrelevant cells from further consideration
- When finish examining the current layer, proceed to the next lower level
- Repeat this process until the bottom layer is reached
- Advantages:
 - Query-independent, easy to parallelize, incremental update
 - $O(K)$, where K is the number of grid cells at the lowest level
- Disadvantages:
 - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

WaveCluster: Clustering by Wavelet Analysis (1998)

- Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
- A multi-resolution clustering approach which applies wavelet transform to the feature space
- How to apply wavelet transform to find clusters
 - Summarizes the data by imposing a multidimensional grid structure onto data space
 - These multidimensional spatial data objects are represented in a n-dimensional feature space
 - Apply wavelet transform on feature space to find the dense regions in the feature space
 - Apply wavelet transform multiple times which result in clusters at different scales from fine to coarse

Wavelet Transform

- Wavelet transform: A signal processing technique that decomposes a signal into different frequency sub-band (can be applied to n-dimensional signals)
 - Data are transformed to preserve relative distance between objects at different levels of resolution
 - Allows natural clusters to become more distinguishable



The WaveCluster Algorithm

- Input parameters
 - # of grid cells for each dimension
 - the wavelet, and the # of applications of wavelet transform
- Why is wavelet transformation useful for clustering?
 - Use hat-shape filters to emphasize region where points cluster, but simultaneously suppress weaker information in their boundary
 - Effective removal of outliers, multi-resolution, cost effective
- Major features:
 - Complexity $O(N)$
 - Detect arbitrary shaped clusters at different scales
 - Not sensitive to noise, not sensitive to input order
 - Only applicable to low dimensional data
- Both grid-based and density-based

Quantization & Transformation

- First, quantize data into m-D grid structure, then wavelet transform
 - a) scale 1: high resolution
 - b) scale 2: medium resolution
 - c) scale 3: low resolution

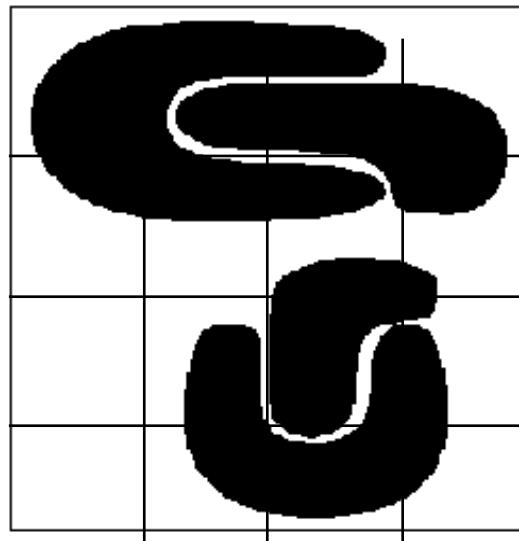
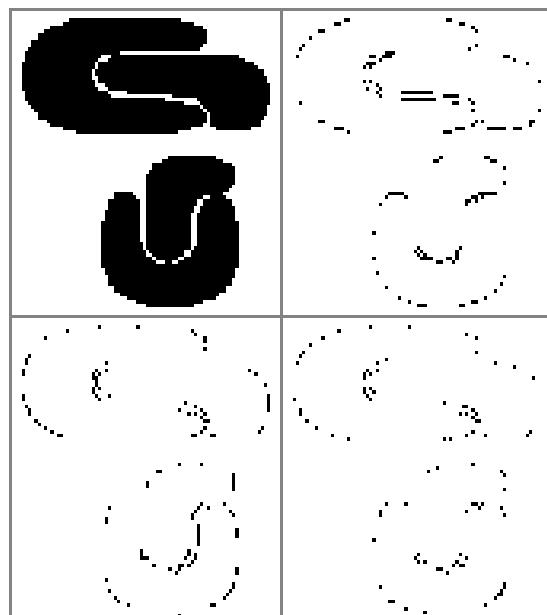


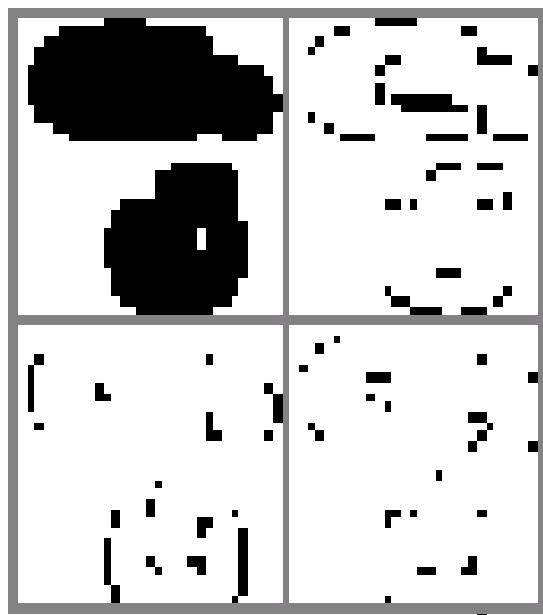
Figure 1: A sample 2-dimensional feature space.



a)



b)



c)

Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary

Model-Based Clustering

- What is model-based clustering?
 - Attempt to optimize the fit between the given data and some mathematical model
 - Based on the assumption: Data are generated by a mixture of underlying probability distribution
- Typical methods
 - Statistical approach
 - EM (Expectation maximization), AutoClass
 - Machine learning approach
 - COBWEB, CLASSIT
 - Neural network approach
 - SOM (Self-Organizing Feature Map)

EM — Expectation Maximization

- EM — A popular iterative refinement algorithm
- An extension to k-means
 - Assign each object to a cluster according to a weight (prob. distribution)
 - New means are computed based on weighted measures
- General idea
 - Starts with an initial estimate of the parameter vector
 - Iteratively rescores the patterns against the mixture density produced by the parameter vector
 - The rescored patterns are used to update the parameter updates
 - Patterns belonging to the same cluster, if they are placed by their scores in a particular component
- Algorithm converges fast but may not be in global optima

The EM (Expectation Maximization) Algorithm

- Initially, randomly assign k cluster centers
- Iteratively refine the clusters based on two steps
 - Expectation step: assign each data point X_i to cluster C_i with the following probability

$$P(X_i \in C_k) = p(C_k | X_i) = \frac{p(C_k)p(X_i | C_k)}{p(X_i)},$$

- Maximization step:
 - Estimation of model parameters

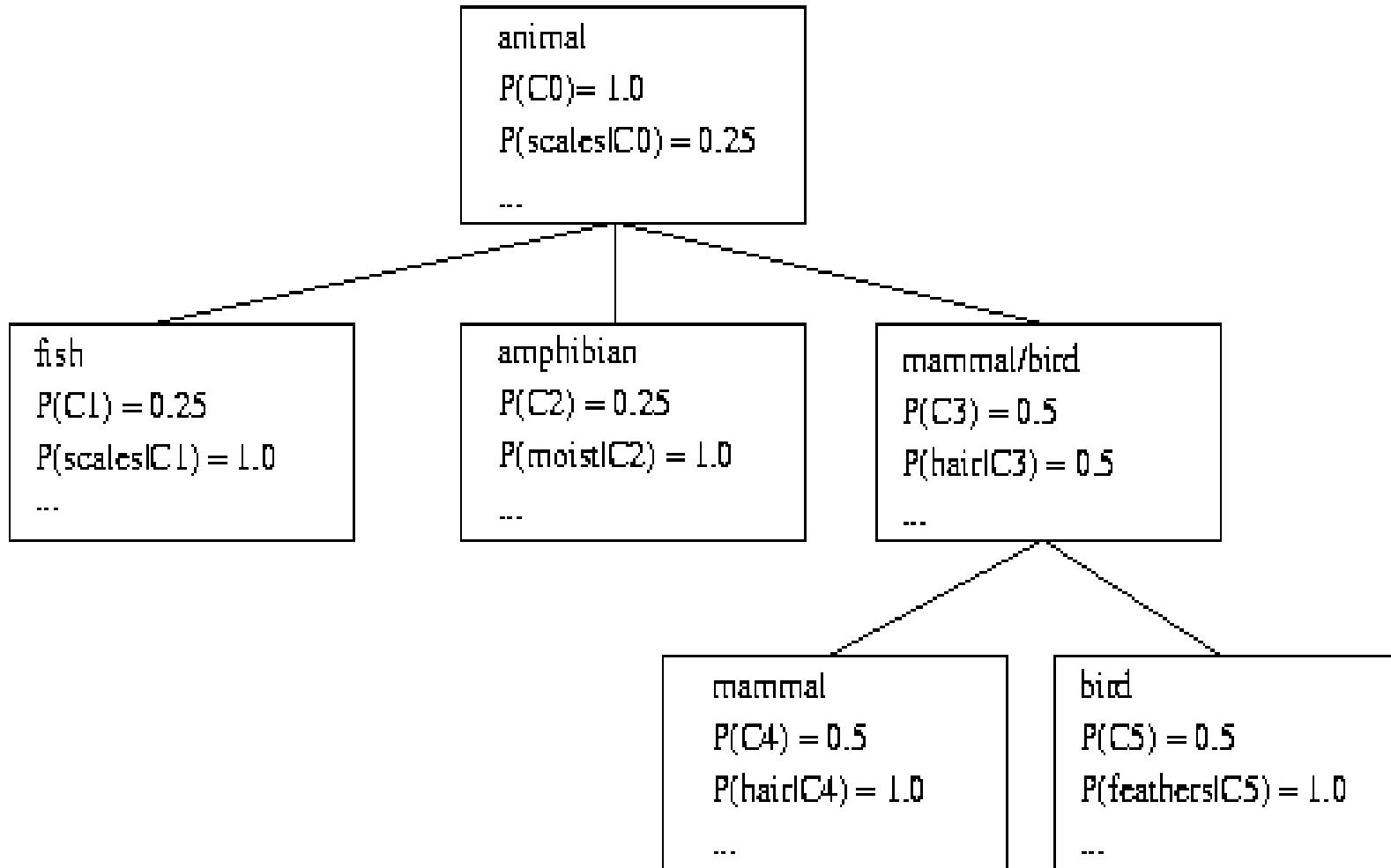
$$m_k = \frac{1}{N} \sum_{i=1}^N \frac{X_i P(X_i \in C_k)}{\sum_j P(X_i \in C_j)}.$$

Conceptual Clustering

- Conceptual clustering
 - A form of clustering in machine learning
 - Produces a classification scheme for a set of unlabeled objects
 - Finds characteristic description for each concept (class)
- COBWEB (Fisher'87)
 - A popular a simple method of incremental conceptual learning
 - Creates a hierarchical clustering in the form of a **classification tree**
 - Each node refers to a concept and contains a probabilistic description of that concept

COBWEB Clustering Method

A classification tree



More on Conceptual Clustering

- Limitations of COBWEB
 - The assumption that the attributes are independent of each other is often too strong because correlation may exist
 - Not suitable for clustering large database data – skewed tree and expensive probability distributions
- CLASSIT
 - an extension of COBWEB for incremental clustering of continuous data
 - suffers similar problems as COBWEB
- AutoClass (Cheeseman and Stutz, 1996)
 - Uses Bayesian statistical analysis to estimate the number of clusters
 - Popular in industry

Neural Network Approach

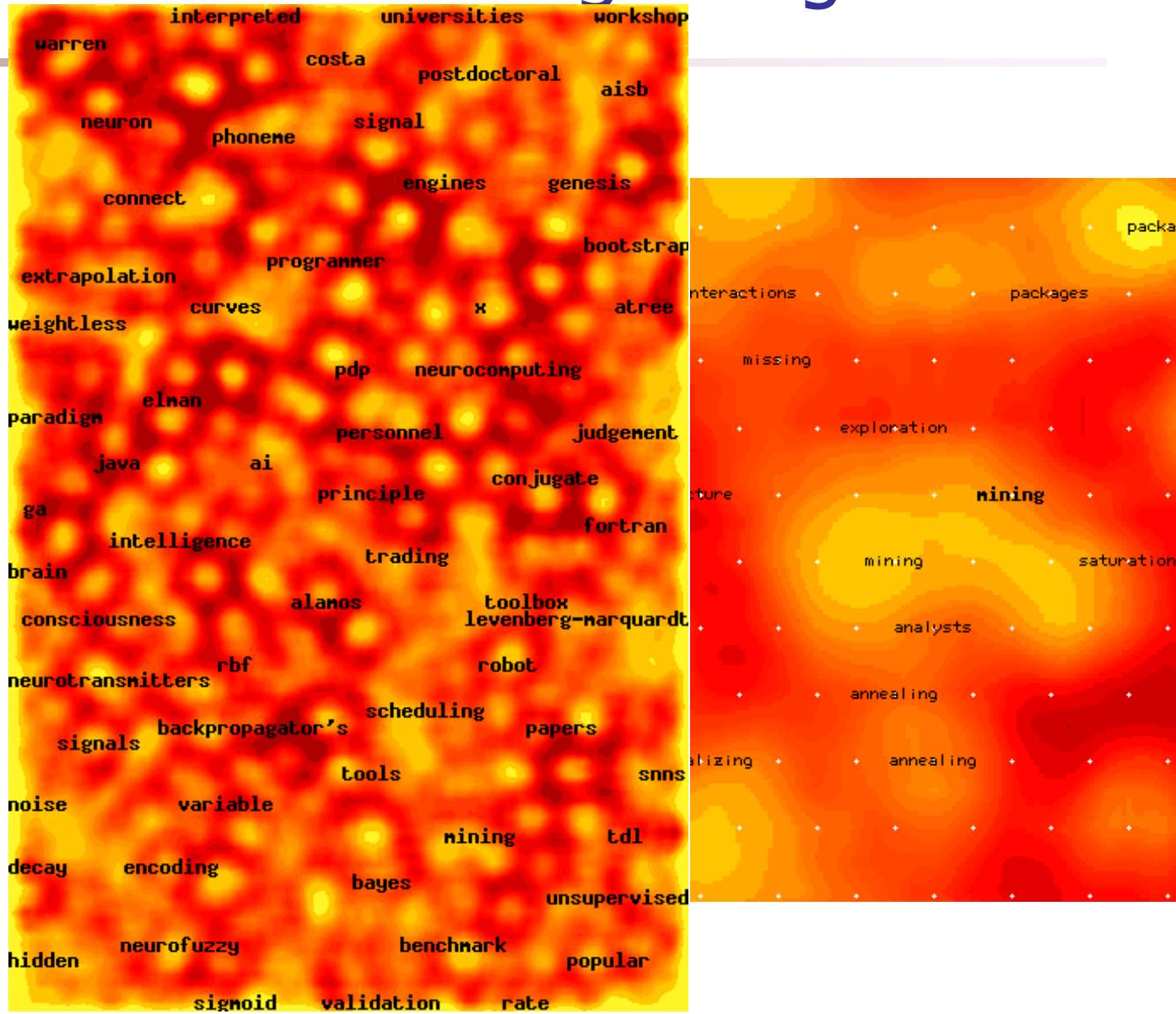
- Neural network approaches
 - Represent each cluster as an exemplar, acting as a “prototype” of the cluster
 - New objects are distributed to the cluster whose exemplar is the most similar according to some distance measure
- Typical methods
 - SOM (Soft-Organizing feature Map)
 - Competitive learning
 - Involves a hierarchical architecture of several units (neurons)
 - Neurons compete in a “winner-takes-all” fashion for the object currently being presented

Self-Organizing Feature Map (SOM)

- SOMs, also called topological ordered maps, or Kohonen Self-Organizing Feature Map (KSOMs)
- It maps all the points in a high-dimensional source space into a 2 to 3-d target space, s.t., the distance and proximity relationship (i.e., topology) are preserved as much as possible
- Similar to k-means: cluster centers tend to lie in a low-dimensional manifold in the feature space
- Clustering is performed by having several units competing for the current object
 - The unit whose weight vector is closest to the current object wins
 - The winner and its neighbors learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space

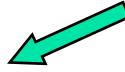
Web Document Clustering Using SOM

- The result of SOM clustering of 12088 Web articles
- The picture on the right: drilling down on the keyword “mining”
- Based on websom.hut.fi Web page



Chapter 6. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary



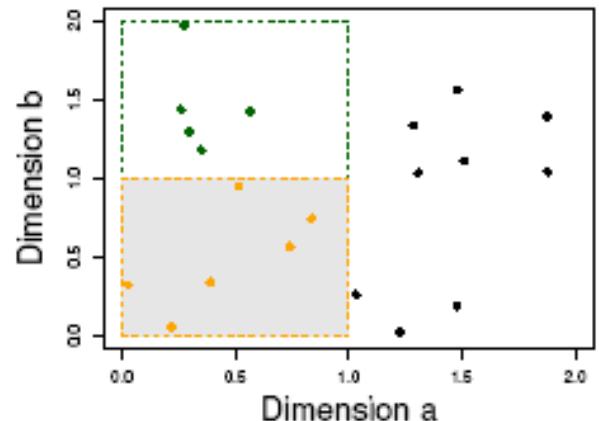
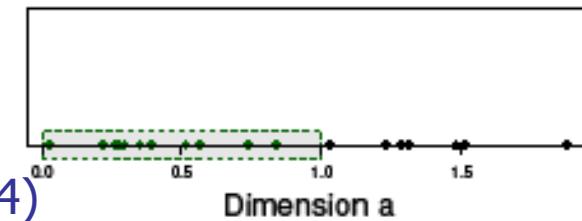
Clustering High-Dimensional Data

- Clustering high-dimensional data
 - Many applications: text documents, DNA micro-array data
 - Major challenges:
 - Many irrelevant dimensions may mask clusters
 - Distance measure becomes meaningless—due to equi-distance
 - Clusters may exist only in some subspaces
- Methods
 - Feature transformation: only effective if most dimensions are relevant
 - PCA & SVD useful only when features are highly correlated/redundant
 - Feature selection: wrapper or filter approaches
 - useful to find a subspace where the data have nice clusters
 - Subspace-clustering: find clusters in all the possible subspaces
 - CLIQUE, ProClus, and frequent pattern-based clustering

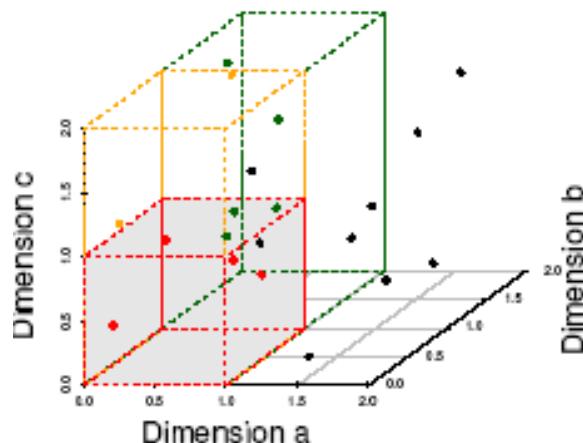
The Curse of Dimensionality

(graphs adapted from Parsons et al. KDD Explorations 2004)

- Data in only one dimension is relatively packed
- Adding a dimension “stretch” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless—due to equi-distance



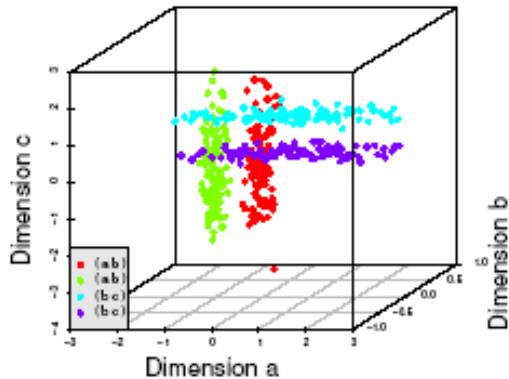
(b) 6 Objects in One Unit Bin



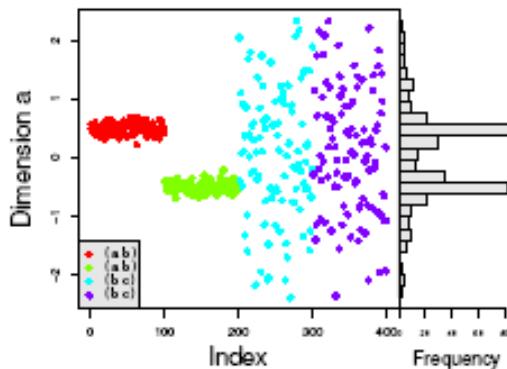
(c) 4 Objects in One Unit Bin

Why Subspace Clustering?

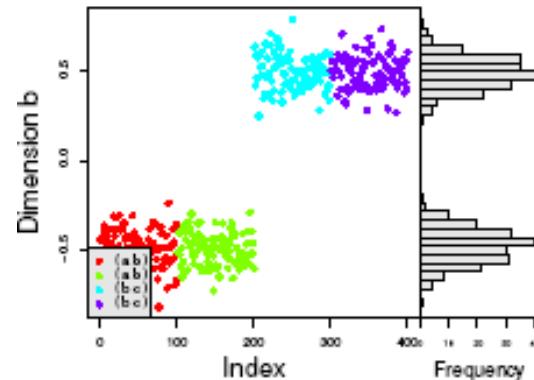
(adapted from Parsons et al. SIGKDD Explorations 2004)



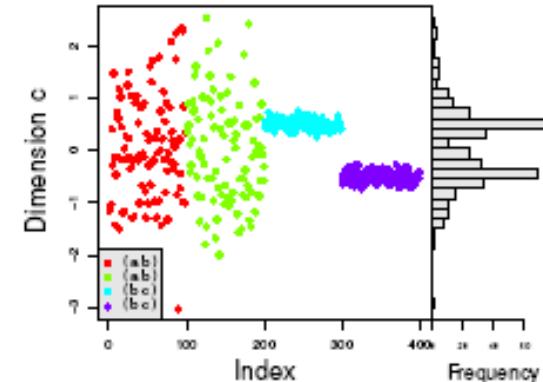
- Clusters may exist only in some subspaces
- Subspace-clustering: find clusters in all the subspaces



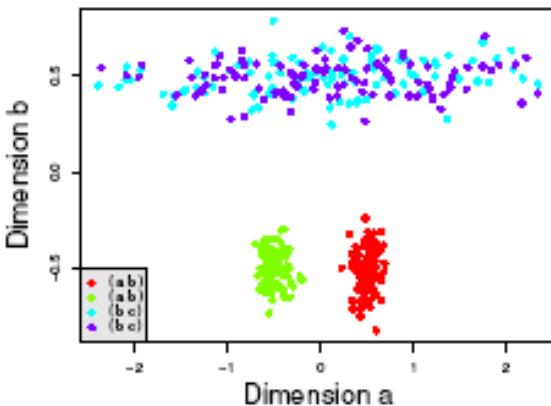
(a) Dimension a



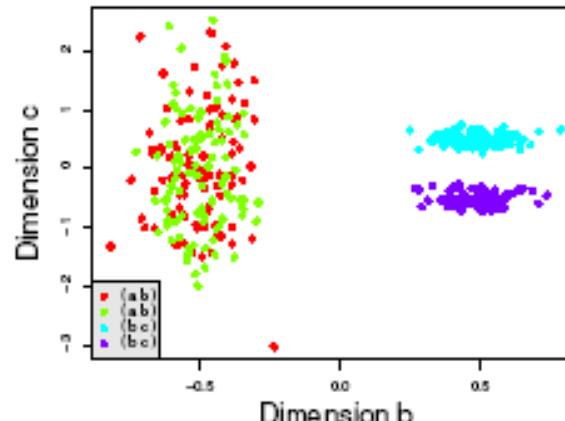
(b) Dimension b



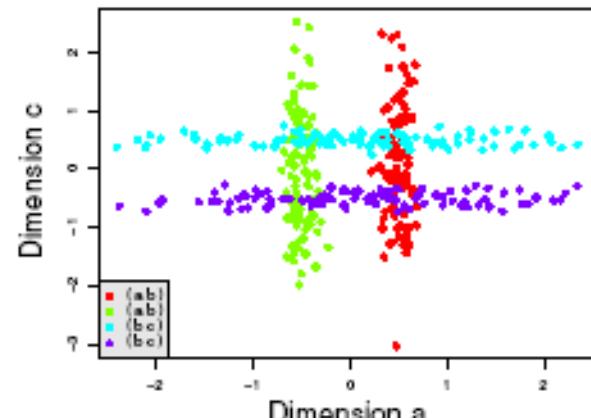
(c) Dimension c



(a) Dims a & b



(b) Dims b & c



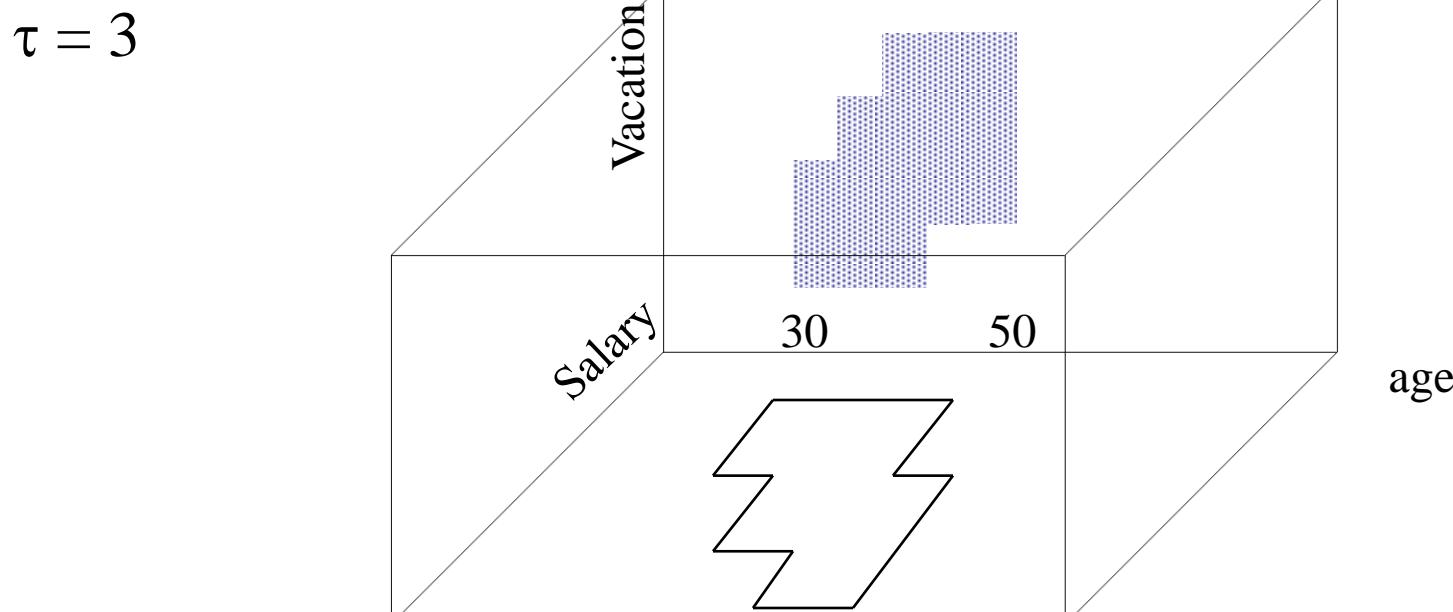
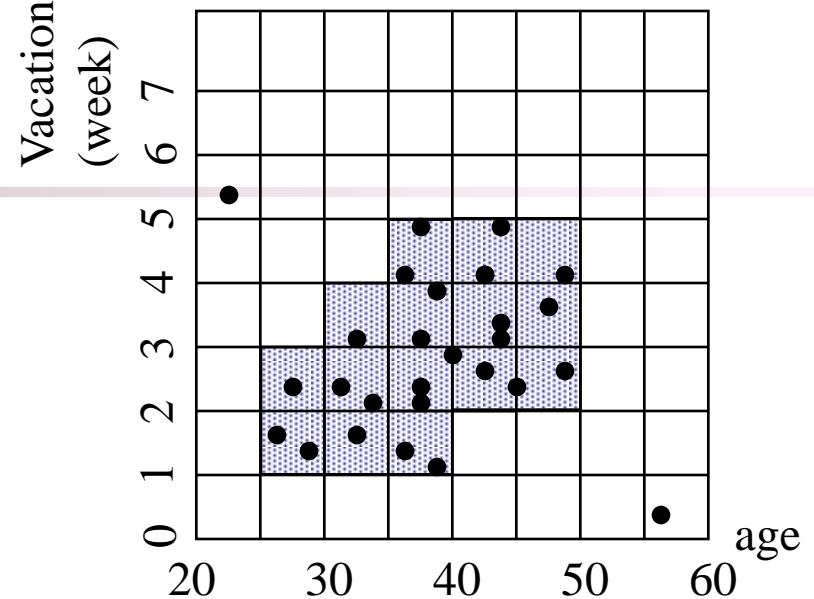
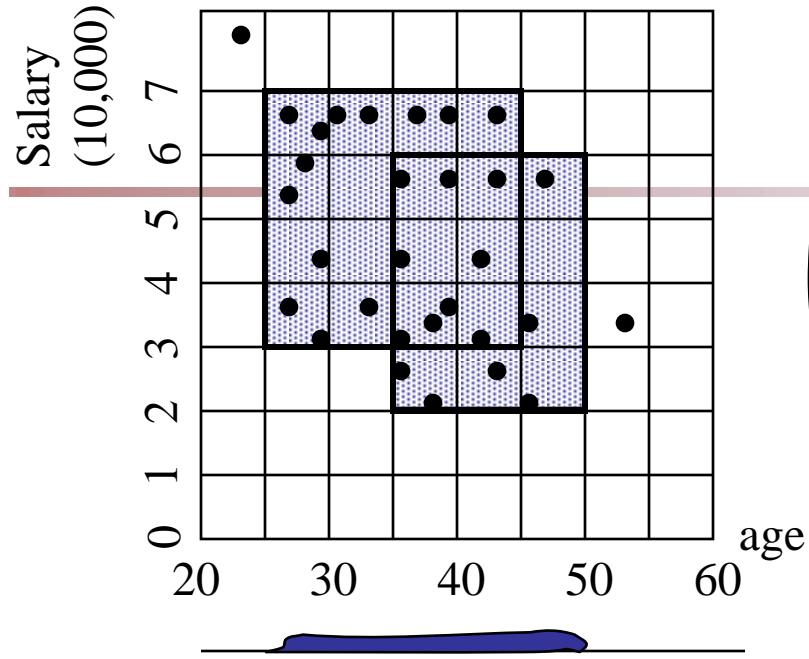
(c) Dims a & c

CLIQUE (Clustering In QUEst)

- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98)
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
 - It partitions each dimension into the same number of equal length interval
 - It partitions an m-dimensional data space into non-overlapping rectangular units
 - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
 - A cluster is a maximal set of connected dense units within a subspace

CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters
 - Determine dense units in all subspaces of interests
 - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
 - Determine maximal regions that cover a cluster of connected dense units for each cluster
 - Determination of minimal cover for each cluster



Strength and Weakness of *CLIQUE*

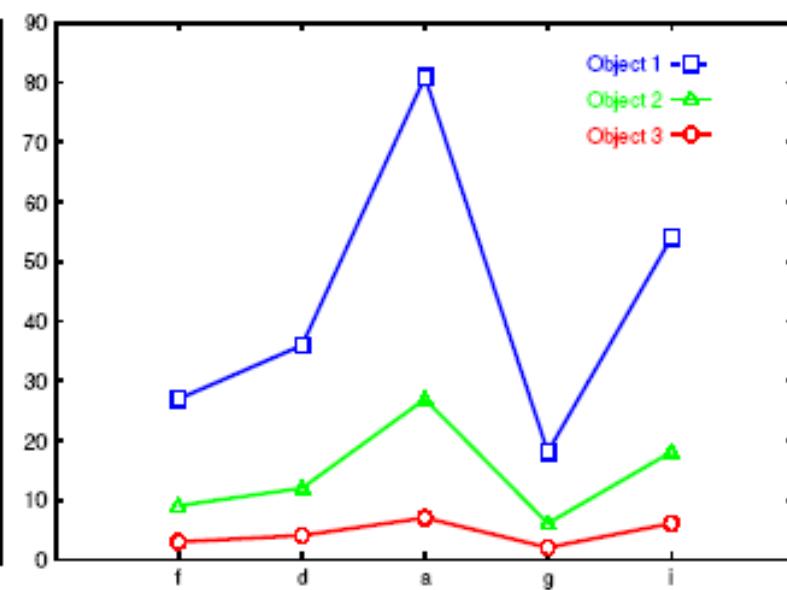
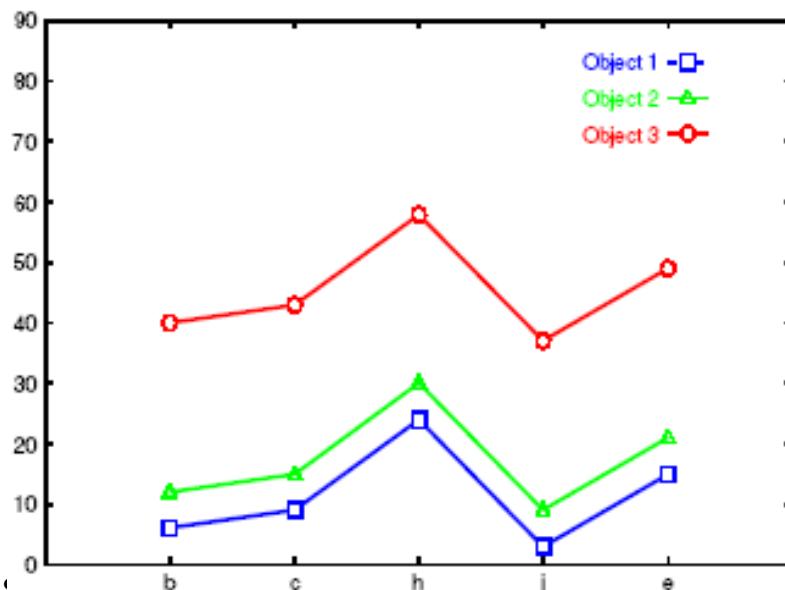
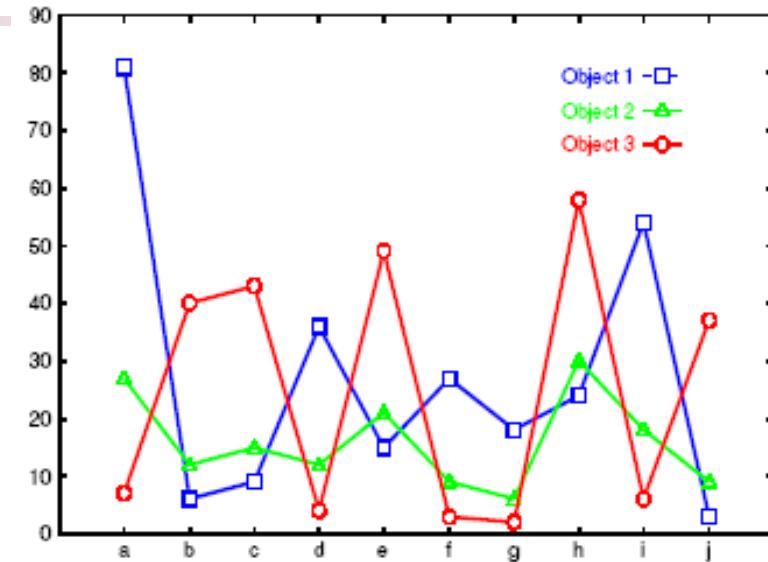
- Strength
 - automatically finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
 - *insensitive* to the order of records in input and does not presume some canonical data distribution
 - scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases
- Weakness
 - The accuracy of the clustering result may be degraded at the expense of simplicity of the method

Frequent Pattern-Based Approach

- Clustering high-dimensional space (e.g., clustering text documents, microarray data)
 - Projected subspace-clustering: which dimensions to be projected on?
 - CLIQUE, ProClus
 - Feature extraction: costly and may not be effective?
 - Using frequent patterns as “features”
 - “Frequent” are inherent features
 - Mining freq. patterns may not be so expensive
- Typical methods
 - Frequent-term-based document clustering
 - Clustering by pattern similarity in micro-array data (pClustering)

Clustering by Pattern Similarity (p -Clustering)

- Right: The micro-array “raw” data shows 3 genes and their values in a multi-dimensional space
 - Difficult to find their patterns
- Bottom: Some subsets of dimensions form nice **shift** and **scaling** patterns



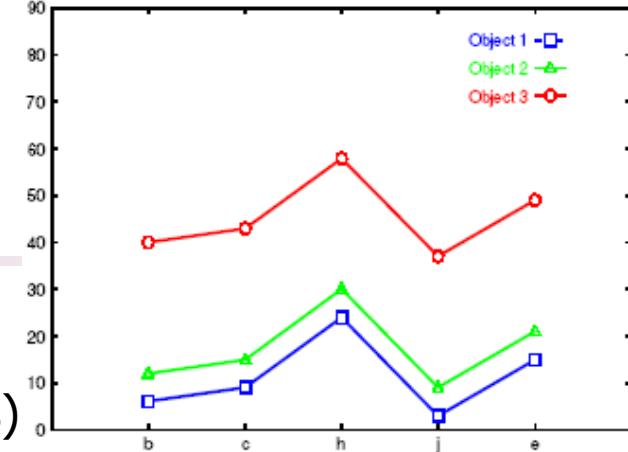
Why p -Clustering?

- Microarray data analysis may need to
 - Clustering on thousands of dimensions (attributes)
 - Discovery of both **shift** and **scaling** patterns
- Clustering with Euclidean distance measure? — cannot find shift patterns
- Clustering on derived attribute $A_{ij} = a_i - a_j$? — introduces **N(N-1)** dimensions
- Bi-cluster using transformed mean-squared residue score matrix (I, J)

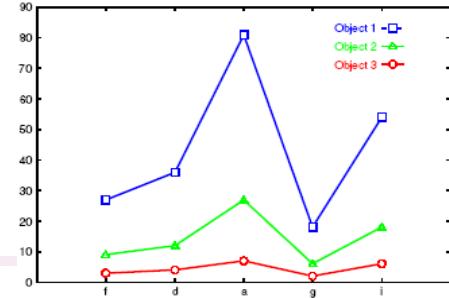
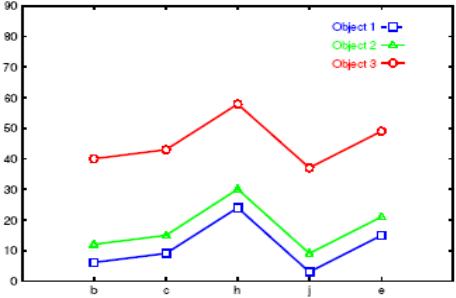
$$H(IJ) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (d_{ij} - d_{iJ} - d_{IJ} + d_{IJ})^2$$

- Where $d_{ij} = \frac{1}{|J|} \sum_{j \in J} d_{ij}$ $d_{iJ} = \frac{1}{|I|} \sum_{i \in I} d_{ij}$ $d_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} d_{ij}$
- A submatrix is a δ -cluster if $H(I, J) \leq \delta$ for some $\delta > 0$

- Problems with bi-cluster
 - No downward closure property,
 - Due to averaging, it may contain outliers but still within δ -threshold



p -Clustering: Clustering by Pattern Similarity



- Given object x, y in O and features a, b in T , p Cluster is a 2 by 2 matrix

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = |(d_{xa} - d_{xb}) - (d_{ya} - d_{yb})|$$

- A pair (O, T) is in δ - p Cluster if for any 2 by 2 matrix X in (O, T) , $pScore(X) \leq \delta$ for some $\delta > 0$
- Properties of δ - p Cluster
 - Downward closure
 - Clusters are more homogeneous than bi-cluster (thus the name: pair-wise Cluster)
- Pattern-growth algorithm has been developed for efficient mining
- For scaling patterns, one can observe, taking logarithmic on $\frac{d_{xa}}{d_{ya}} / \frac{d_{xb}}{d_{yb}} < \delta$ will lead to the p Score form

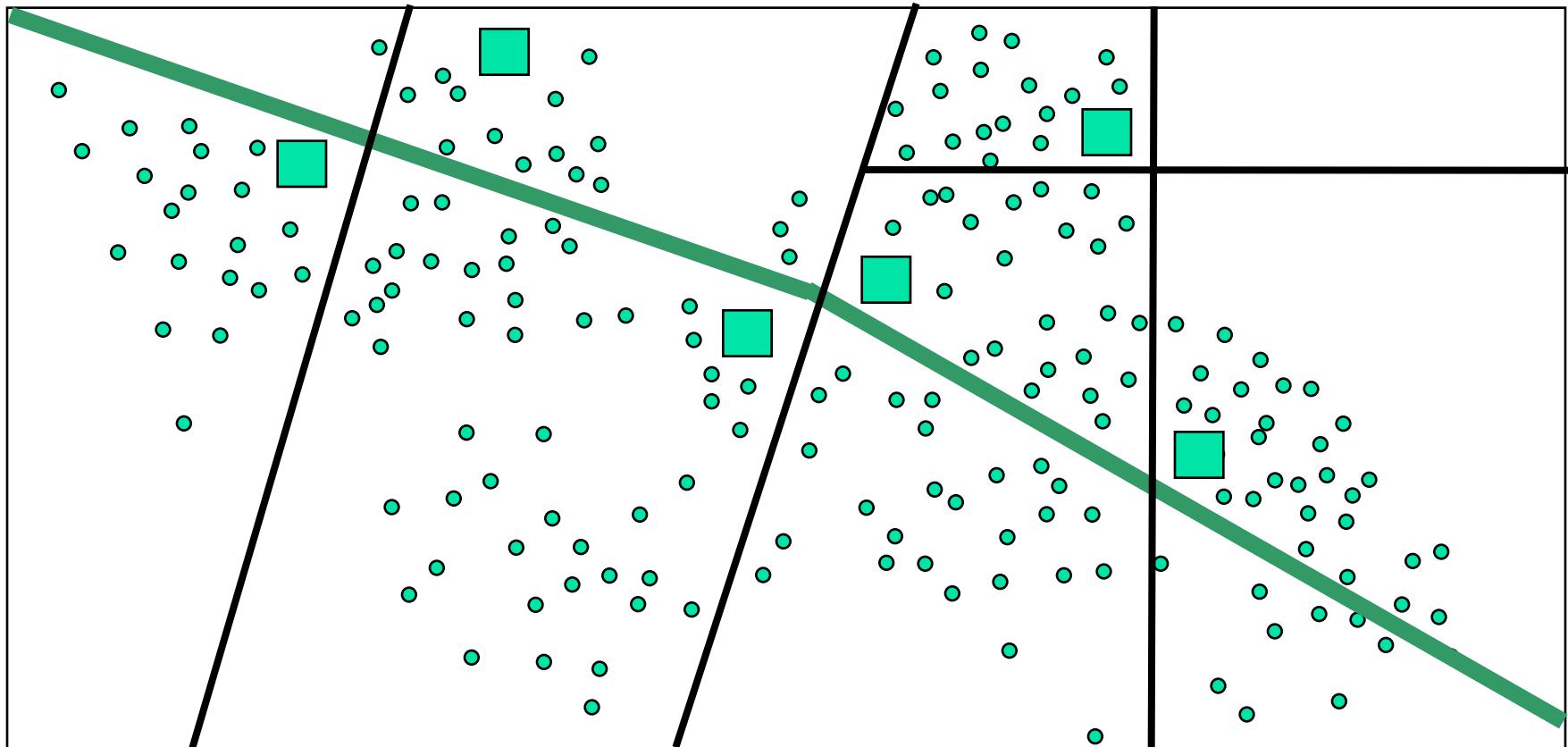
Chapter 6. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary



Why Constraint-Based Cluster Analysis?

- Need user feedback: Users know their applications the best
- Less parameters but more user-desired constraints, e.g., an ATM allocation problem: obstacle & desired clusters

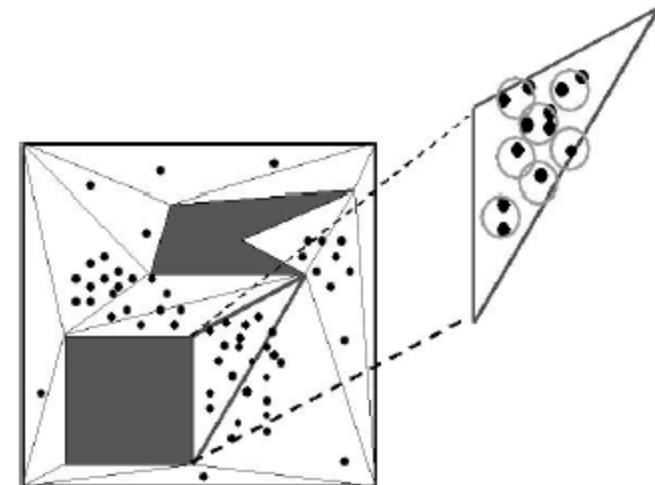
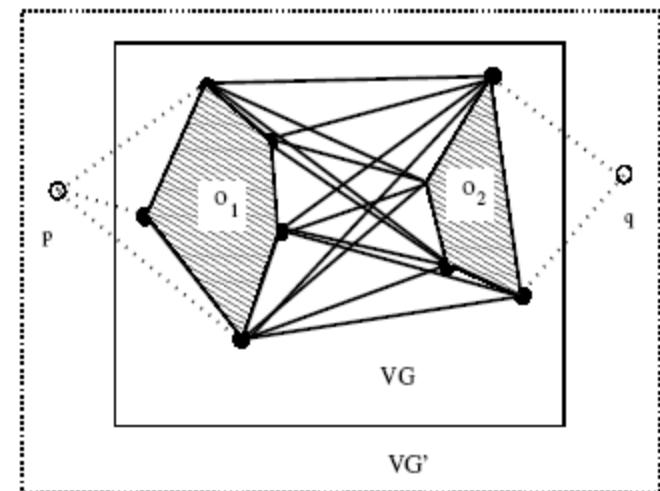


A Classification of Constraints in Cluster Analysis

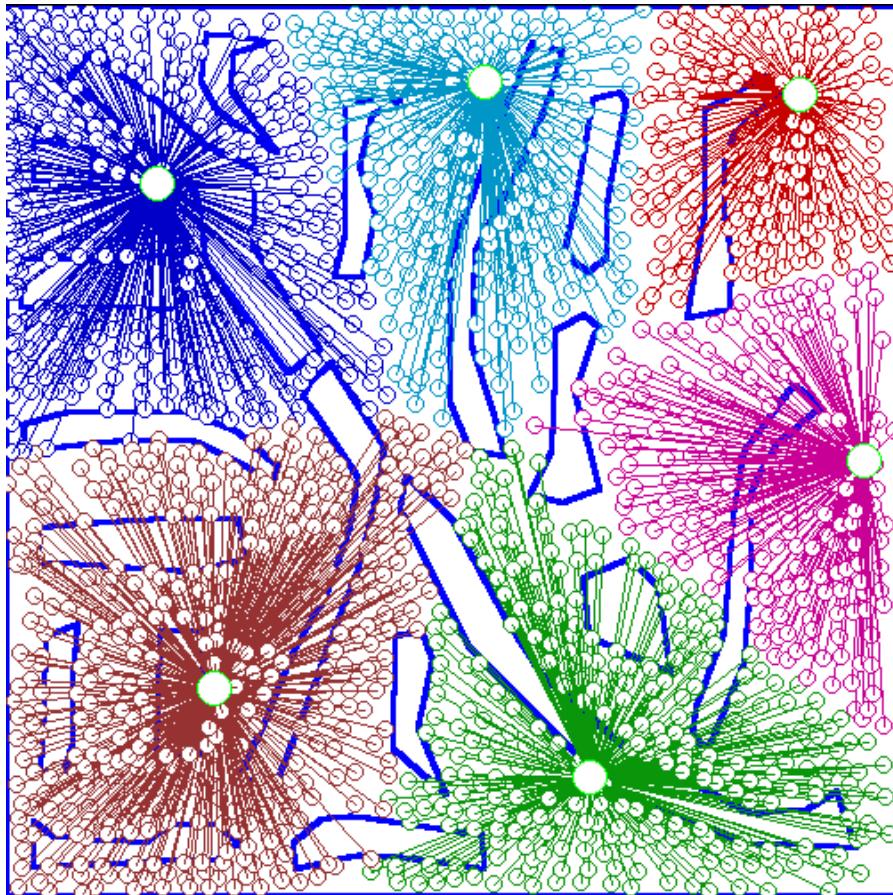
- Clustering in applications: desirable to have user-guided (i.e., constrained) cluster analysis
- Different constraints in cluster analysis:
 - Constraints on individual objects (do selection first)
 - Cluster on houses worth over \$300K
 - Constraints on distance or similarity functions
 - Weighted functions, obstacles (e.g., rivers, lakes)
 - Constraints on the selection of clustering parameters
 - # of clusters, MinPts, etc.
 - User-specified constraints
 - Contain at least 500 valued customers and 5000 ordinary ones
 - Semi-supervised: giving small training sets as “constraints” or hints

Clustering With Obstacle Objects

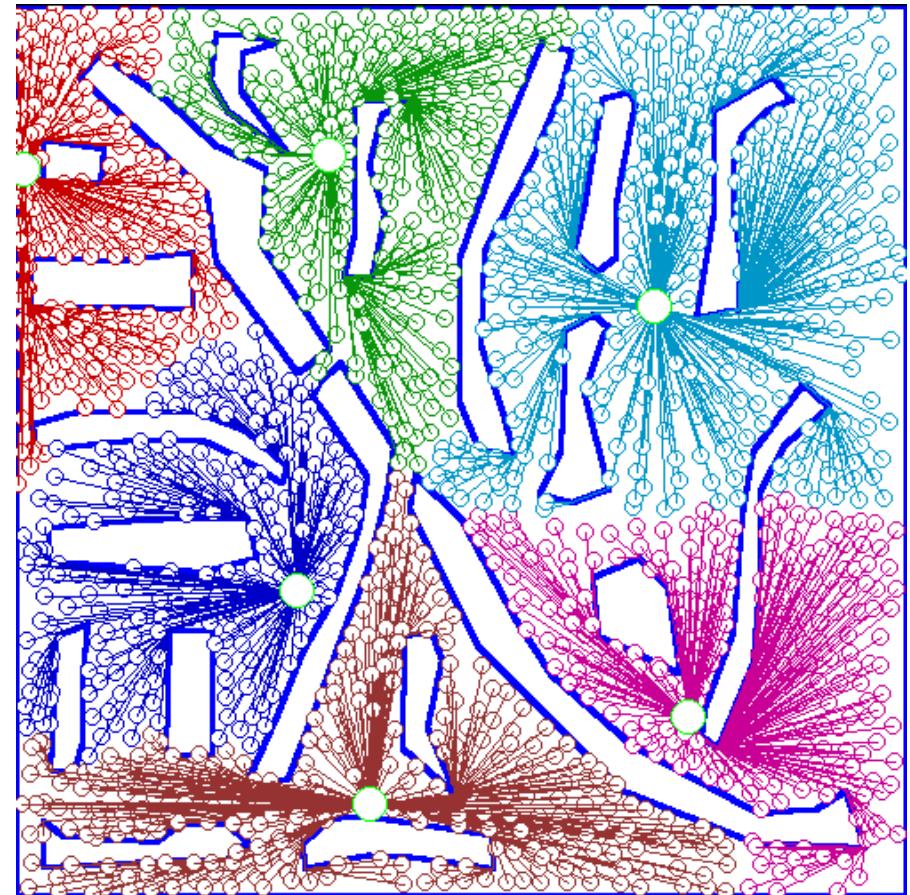
- K-medoids is more preferable since k-means may locate the ATM center in the middle of a lake
- Visibility graph and shortest path
- Triangulation and micro-clustering
- Two kinds of join indices (shortest-paths) worth pre-computation
 - VV index: indices for any pair of obstacle vertices
 - MV index: indices for any pair of micro-cluster and obstacle indices



An Example: Clustering With Obstacle Objects



Not Taking obstacles into account



Taking obstacles into account

Clustering with User-Specified Constraints

- Example: Locating k delivery centers, each serving at least m valued customers and n ordinary ones
- Proposed approach
 - Find an initial “solution” by partitioning the data set into k groups and satisfying user-constraints
 - Iteratively refine the solution by micro-clustering relocation (e.g., moving $\delta \mu$ -clusters from cluster C_i to C_j) and “deadlock” handling (break the microclusters when necessary)
 - Efficiency is improved by micro-clustering
- How to handle more complicated constraints?
 - E.g., having approximately same number of valued customers in each cluster?! — Can you solve it?

Cluster Analysis

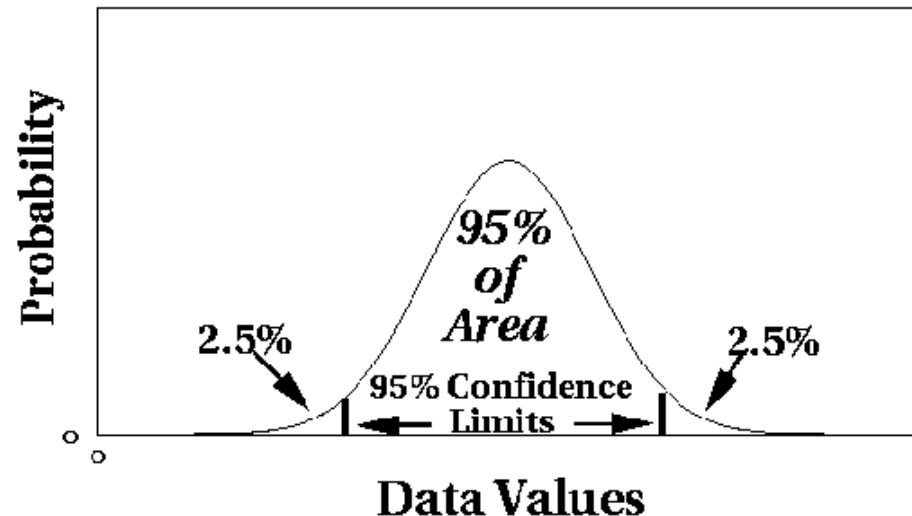
1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary



What Is Outlier Discovery?

- What are outliers?
 - The set of objects are considerably dissimilar from the remainder of the data
 - Example: Sports: Michael Jordon, Wayne Gretzky, ...
- Problem: Define and find outliers in large data sets
- Applications:
 - Credit card fraud detection
 - Telecom fraud detection
 - Customer segmentation
 - Medical analysis

Outlier Discovery: Statistical Approaches



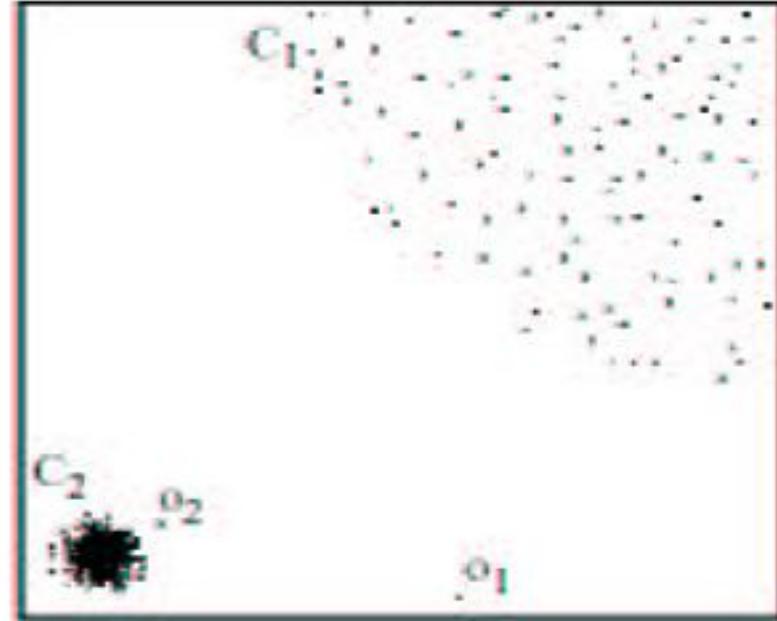
- ✖ Assume a model underlying distribution that generates data set (e.g. normal distribution)
- Use discordancy tests depending on
 - data distribution
 - distribution parameter (e.g., mean, variance)
 - number of expected outliers
- Drawbacks
 - most tests are for single attribute
 - In many cases, data distribution may not be known

Outlier Discovery: Distance-Based Approach

- Introduced to counter the main limitations imposed by statistical methods
 - We need multi-dimensional analysis without knowing data distribution
- Distance-based outlier: A DB(p , D)-outlier is an object O in a dataset T such that at least a fraction p of the objects in T lies at a distance greater than D from O
- Algorithms for mining distance-based outliers
 - Index-based algorithm
 - Nested-loop algorithm
 - Cell-based algorithm

Density-Based Local Outlier Detection

- Distance-based outlier detection is based on global distance distribution
- It encounters difficulties to identify outliers if data is not uniformly distributed
- Ex. C_1 contains 400 loosely distributed points, C_2 has 100 tightly condensed points, 2 outlier points o_1, o_2
- Distance-based method cannot identify o_2 as an outlier
- Need the concept of local outlier



- Local outlier factor (LOF)
 - Assume outlier is not crisp
 - Each point has a LOF

Outlier Discovery: Deviation-Based Approach

- Identifies outliers by examining the main characteristics of objects in a group
- Objects that “deviate” from this description are considered outliers
- Sequential exception technique
 - simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects
- OLAP data cube technique
 - uses data cubes to identify regions of anomalies in large multidimensional data

Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Grid-Based Methods
8. Model-Based Methods
9. Clustering High-Dimensional Data
10. Constraint-Based Clustering
11. Outlier Analysis
12. Summary



Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- There are still lots of research issues on cluster analysis

Problems and Challenges

- Considerable progress has been made in scalable clustering methods
 - Partitioning: k-means, k-medoids, CLARANS
 - Hierarchical: BIRCH, ROCK, CHAMELEON
 - Density-based: DBSCAN, OPTICS, DenClue
 - Grid-based: STING, WaveCluster, CLIQUE
 - Model-based: EM, Cobweb, SOM
 - Frequent pattern-based: pCluster
 - Constraint-based: COD, constrained-clustering
- Current clustering techniques do not address all the requirements adequately, still an active area of research

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- P. Arabie, L. J. Hubert, and G. De Soete. Clustering and Classification. World Scientific, 1996
- Beil F., Ester M., Xu X.: "[Frequent Term-Based Text Clustering](#)", KDD'02
- M. M. Breunig, H.-P. Kriegel, R. Ng, J. Sander. LOF: Identifying Density-Based Local Outliers. SIGMOD 2000.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. VLDB'98.

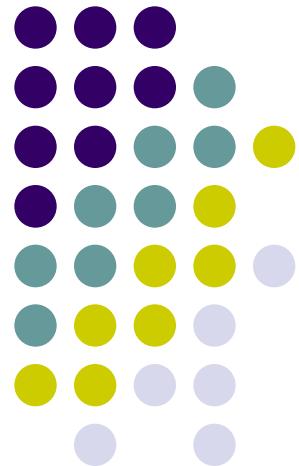
References (2)

- V. Ganti, J. Gehrke, R. Ramakrishnan. CACTUS Clustering Categorical Data Using Summaries. *KDD'99*.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. *SIGMOD'98*.
- S. Guha, R. Rastogi, and K. Shim. [ROCK: A robust clustering algorithm for categorical attributes](#). In *ICDE'99*, pp. 512-521, Sydney, Australia, March 1999.
- A. Hinneburg, D.I A. Keim: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. *KDD'98*.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.
- G. Karypis, E.-H. Han, and V. Kumar. [CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling](#). *COMPUTER*, 32(8): 68-75, 1999.
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. *VLDB'98*.
- G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- P. Michaud. Clustering techniques. *Future Generation Computer systems*, 13, 1997.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. *VLDB'94*.

References (3)

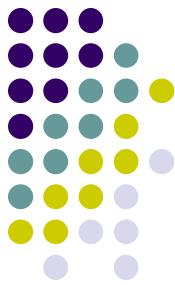
- L. Parsons, E. Haque and H. Liu, [Subspace Clustering for High Dimensional Data: A Review](#), SIGKDD Explorations, 6(1), June 2004
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition,,
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. [Constraint-Based Clustering in Large Databases](#), ICDT'01.
- A. K. H. Tung, J. Hou, and J. Han. [Spatial Clustering in the Presence of Obstacles](#) , ICDE'01
- H. Wang, W. Wang, J. Yang, and P.S. Yu. [Clustering by pattern similarity in large data sets](#), SIGMOD'02.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. SIGMOD'96.

K-MEANS CLUSTERING



INTRODUCTION-

What is clustering?



- **Clustering** is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

Types of clustering:



1. **Hierarchical algorithms**: these find successive clusters using previously established clusters.
 1. Agglomerative ("bottom-up"): Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.
 2. Divisive ("top-down"): Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.
2. **Partitional clustering**: Partitional algorithms determine all clusters at once. They include:
 - **K-means and derivatives**
 - Fuzzy c-means clustering
 - QT clustering algorithm

Common Distance measures:



- *Distance measure* will determine how the *similarity* of two elements is calculated and it will influence the shape of the clusters.

They include:

1. The Euclidean distance (also called 2-norm distance) is given by:

$$d(x, y) = \sqrt{\sum_{i=1}^p |x_i - y_i|^2}$$

2. The Manhattan distance (also called taxicab norm or 1-norm) is given by:

$$d(x, y) = \sqrt[3]{\sum_{i=1}^p |x_i - y_i|^3}$$



3. The maximum norm is given by:

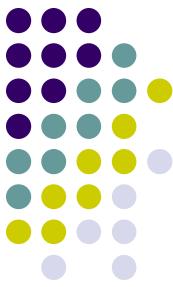
$$d(x, y) = \max_{1 \leq i \leq p} |x_i - y_i|$$

4. The Mahalanobis distance corrects data for different scales and correlations in the variables.
5. Inner product space: The angle between two vectors can be used as a distance measure when clustering high dimensional data
6. Hamming distance (sometimes edit distance) measures the minimum number of substitutions required to change one member into another.



K-MEANS CLUSTERING

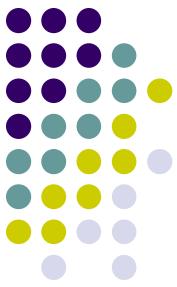
- The **k-means algorithm** is an algorithm to cluster n objects based on attributes into k partitions, where $k < n$.
- It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.
- It assumes that the object attributes form a vector space.



- An algorithm for partitioning (or clustering) N data points into K disjoint subsets S_j containing data points so as to minimize the sum-of-squares criterion

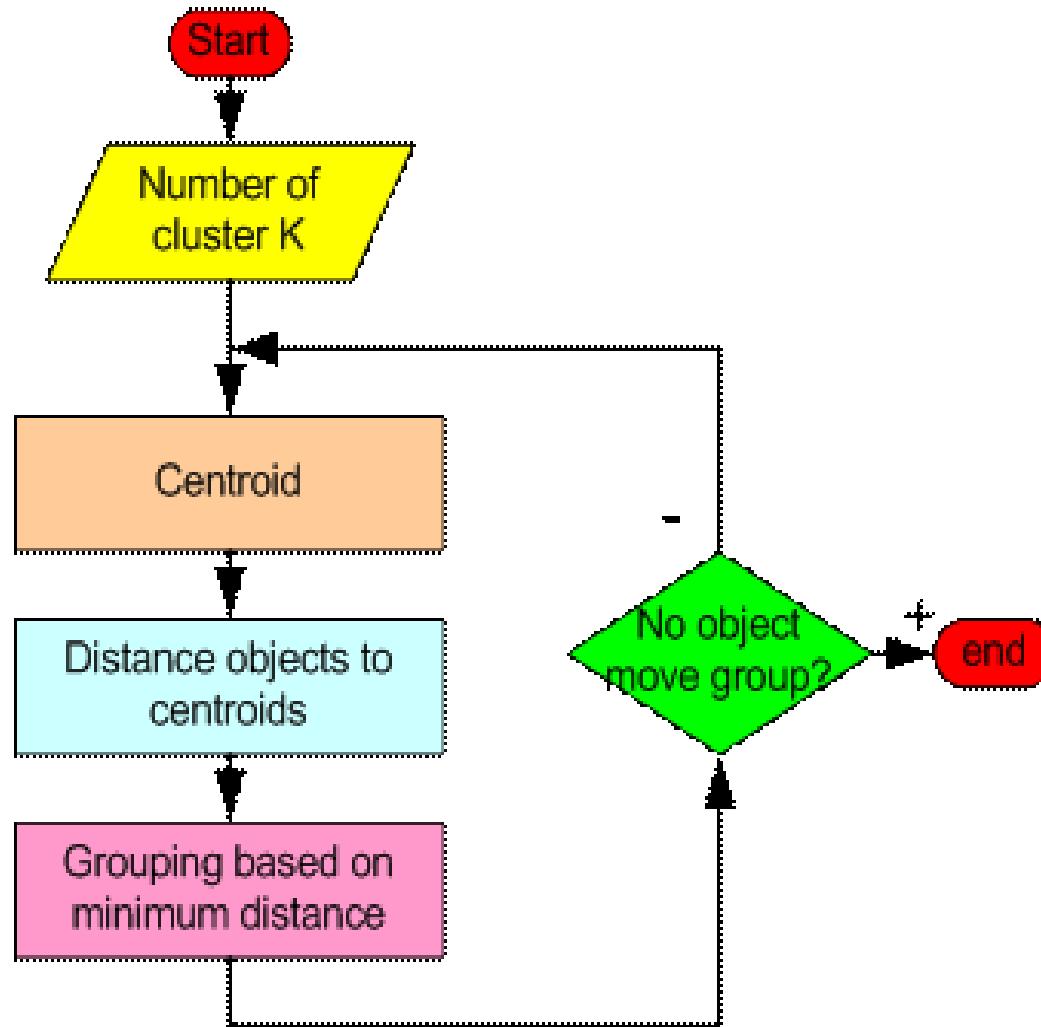
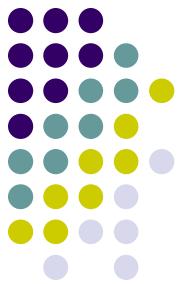
$$J = \sum_{j=1}^K \sum_{n \in S_j} \|x_n - \mu_j\|^2,$$

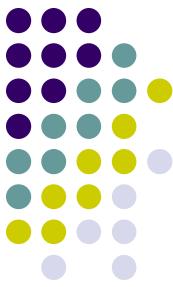
where x_n is a vector representing the the n^{th} data point and μ_j is the geometric centroid of the data points in S_j .



- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

How the K-Mean Clustering algorithm works?





- **Step 1:** Begin with a decision on the value of k = number of clusters .
- **Step 2:** Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following:
 1. Take the first k training sample as single-element clusters
 2. Assign each of the remaining $(N-k)$ training sample to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.



- **Step 3:** Take each sample in sequence and compute its [distance](#) from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.
- **Step 4 .** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

A Simple example showing the implementation of k-means algorithm

(using K=2)



Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5



Step 1:

Initialization: Randomly we choose following two centroids ($k=2$) for two clusters.

In this case the 2 centroid are: $m_1=(1.0,1.0)$ and $m_2=(5.0,7.0)$.

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Individual	Mean Vector
Group 1	(1.0, 1.0)
Group 2	(5.0, 7.0)

Step 2:

- Thus, we obtain two clusters containing:
 $\{1,2,3\}$ and $\{4,5,6,7\}$.
- Their new centroids are:

$$m_1 = \left(\frac{1}{3}(1.0 + 1.5 + 3.0), \frac{1}{3}(1.0 + 2.0 + 4.0) \right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0 + 3.5 + 4.5 + 3.5), \frac{1}{4}(7.0 + 5.0 + 5.0 + 4.5) \right) \\ = (4.12, 5.38)$$

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

$$d(m_1, 2) = \sqrt{|1.0 - 1.5|^2 + |1.0 - 2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0 - 1.5|^2 + |7.0 - 2.0|^2} = 6.10$$



Step 3:

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.
- Therefore, the new clusters are:
 $\{1,2\}$ and $\{3,4,5,6,7\}$
- Next centroids are:
 $m_1=(1.25,1.5)$ and $m_2 = (3.9,5.1)$

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08



- Step 4 :

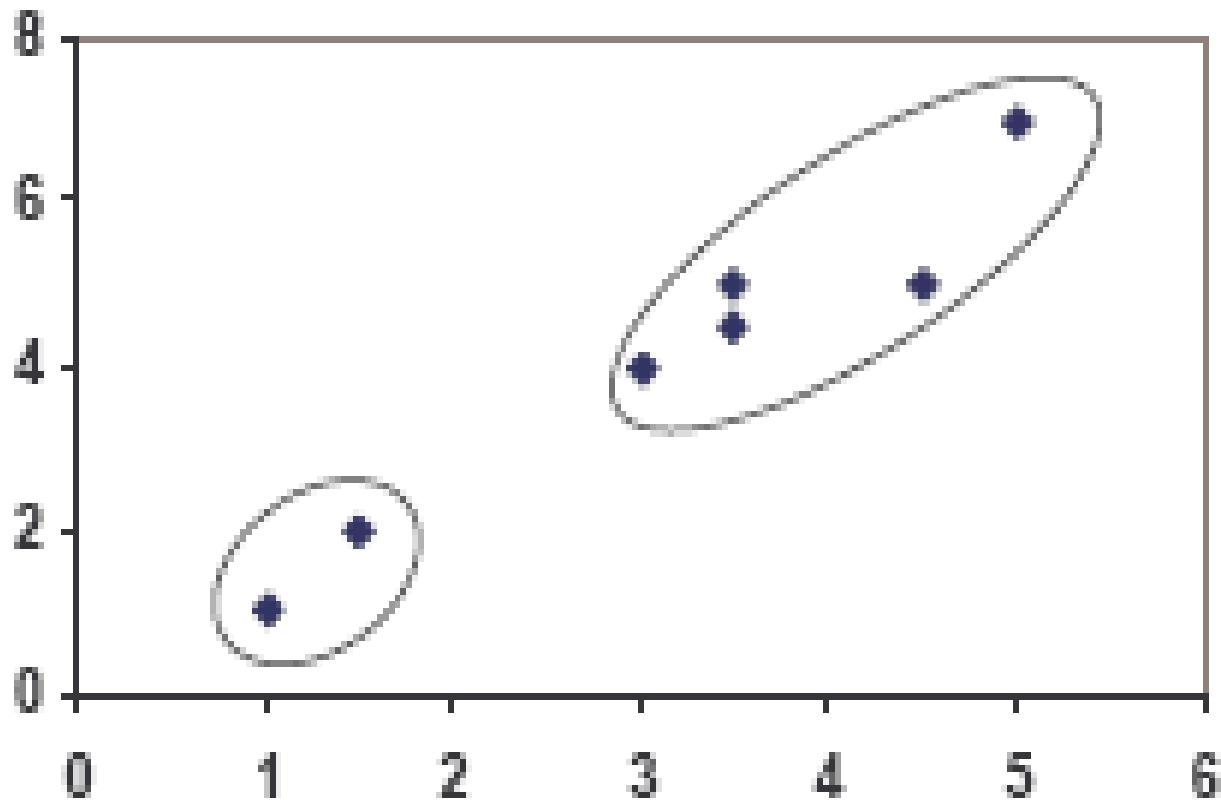
The clusters obtained are:

{1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.58	5.02
2	0.58	3.92
3	3.05	1.42
4	6.88	2.20
5	4.18	0.41
6	4.78	0.61
7	3.75	0.72

PLOT



(with K=3)



Individual	$m_1 = 1$	$m_2 = 2$	$m_3 = 3$	cluster
1	0	1.11	3.61	1
2	1.12	0	2.5	2
3	3.61	2.5	0	3
4	7.21	6.10	3.61	3
5	4.72	3.61	1.12	3
6	5.31	4.24	1.80	3
7	4.30	3.20	0.71	3

$\{ C_3 \}$

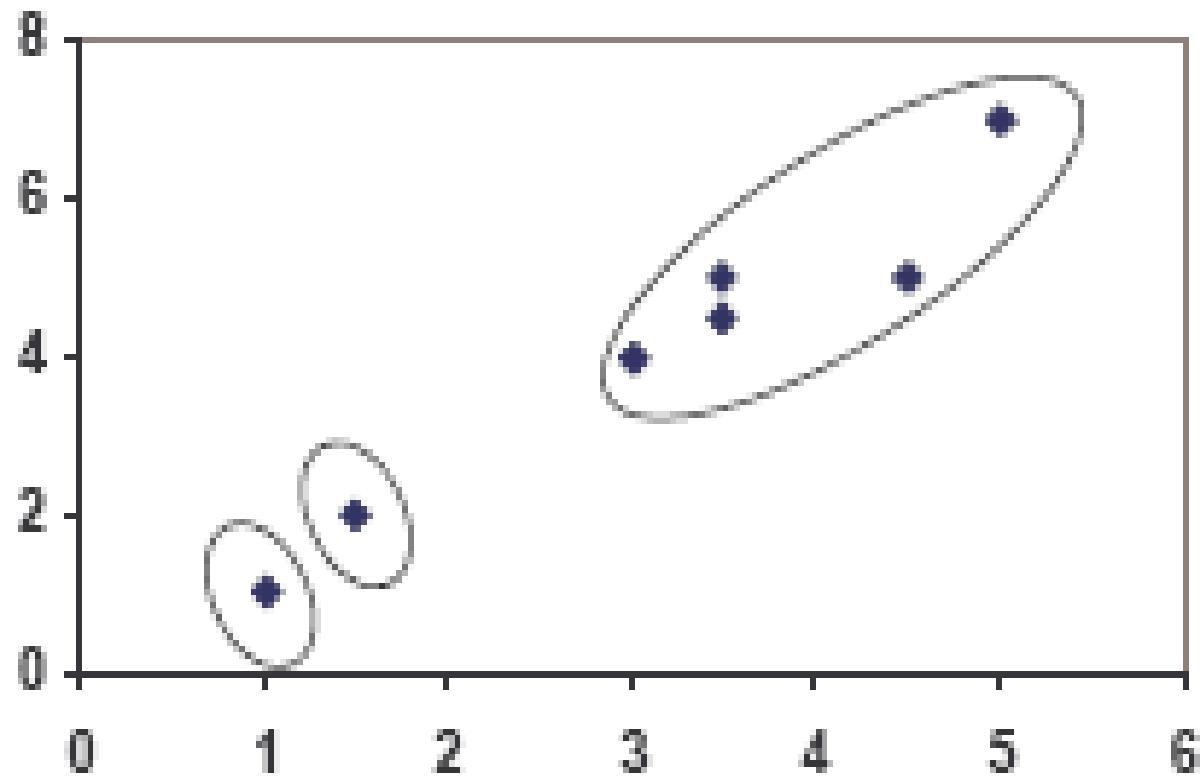
clustering with initial centroids (1, 2, 3)

Step 1

Individual	m_1 (1.0, 1.0)	m_2 (1.5, 2.0)	m_3 (3.9, 5.1)	cluster
1	0	1.11	5.02	1
2	1.12	0	3.92	2
3	3.61	2.5	1.42	3
4	7.21	6.10	2.20	3
5	4.72	3.61	0.41	3
6	5.31	4.24	0.81	3
7	4.30	3.20	0.72	3

Step 2

PLOT





Real-Life Numerical Example of K-Means Clustering

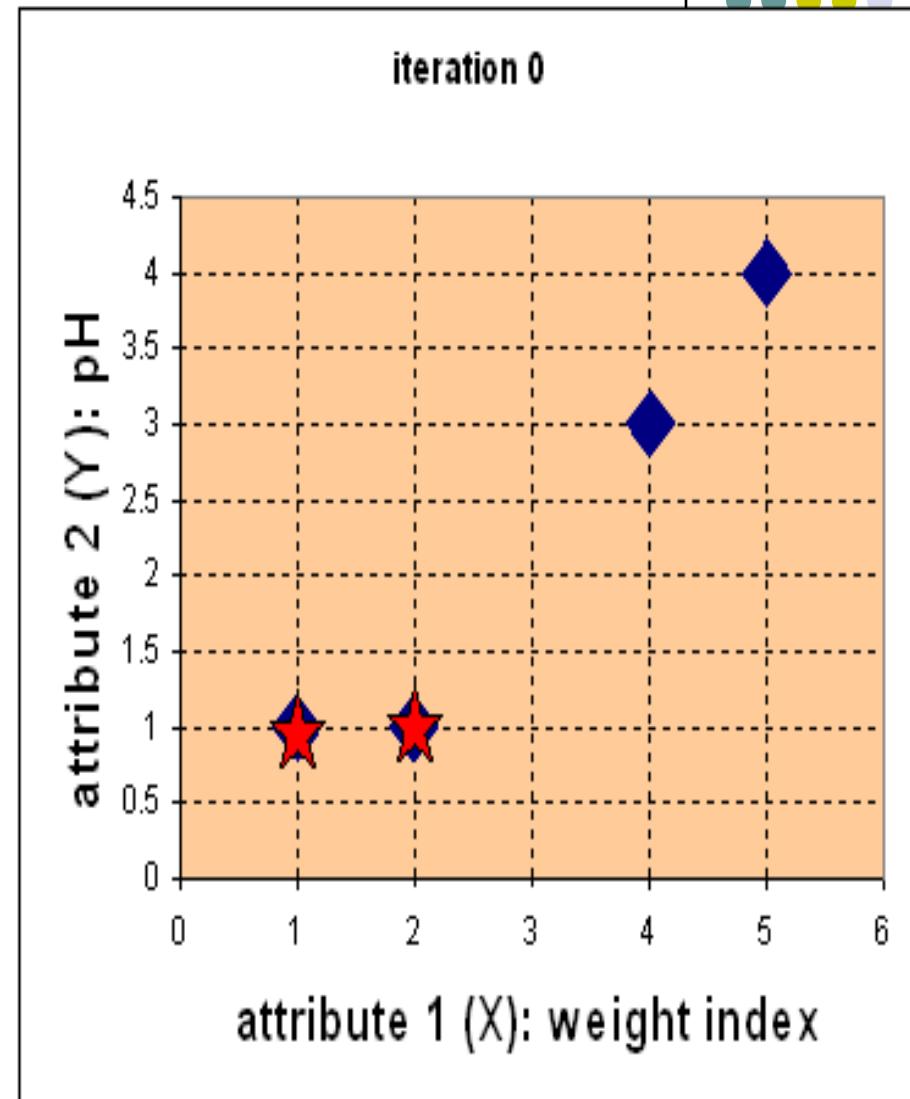
We have 4 medicines as our training data points object and each medicine has 2 attributes. Each attribute represents coordinate of the object. We have to determine which medicines belong to cluster 1 and which medicines belong to the other cluster.

Object	Attribute1 (X): weight index	Attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4



Step 1:

- Initial value of centroids : Suppose we use medicine A and medicine B as the first centroids.
- Let and c_1 and c_2 denote the coordinate of the centroids, then $c_1=(1,1)$ and $c_2=(2,1)$





- **Objects-Centroids distance** : we calculate the distance between cluster centroid to each object. Let us use Euclidean distance, then we have distance matrix at iteration 0 is

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{array}{ll} \mathbf{c}_1 = (1,1) & \text{group - 1} \\ \mathbf{c}_2 = (2,1) & \text{group - 2} \end{array}$$
$$\begin{array}{cccc} A & B & C & D \end{array}$$
$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{array}{ll} X \\ Y \end{array}$$

- Each column in the distance matrix symbolizes the object.
- The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid.
- For example, distance from medicine C = (4, 3) to the first centroid $\mathbf{c}_1 = (1,1)$ is , $\sqrt{(4-1)^2 + (3-1)^2} = 3.61$ and its distance to the second centroid is , $\mathbf{c}_2 = (2,1)$ is $\sqrt{(4-2)^2 + (3-1)^2} = 2.83$ etc.

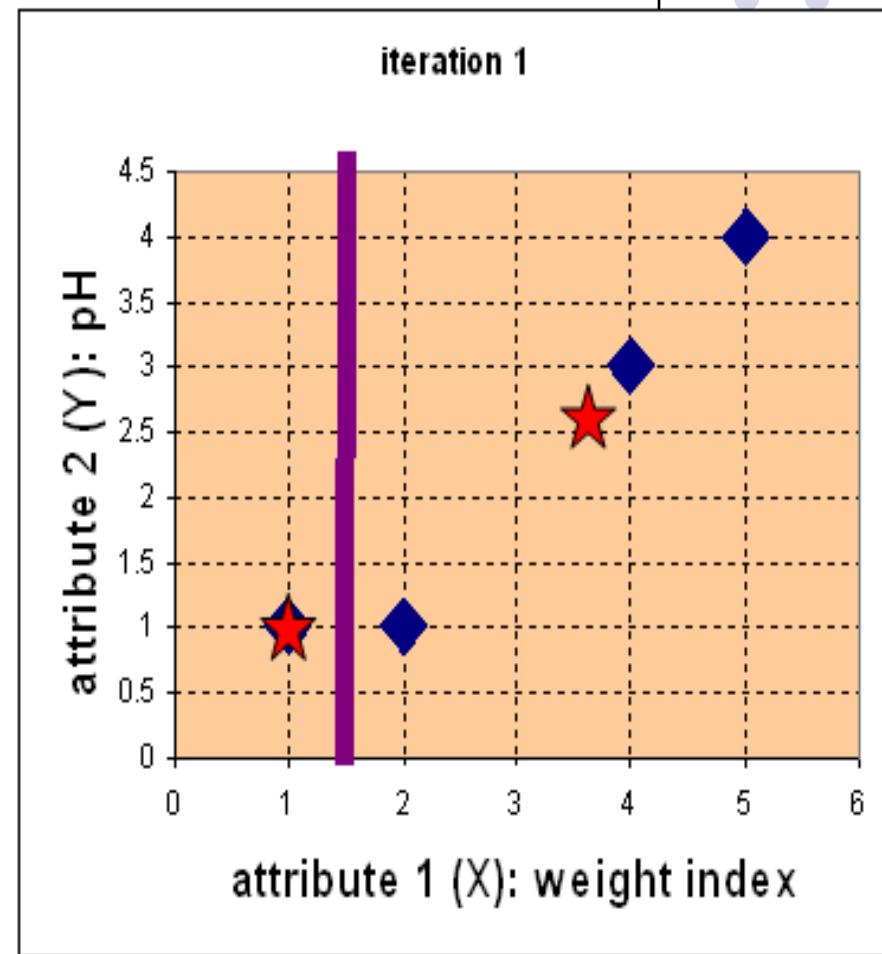
Step 2:



- **Objects clustering** : We assign each object based on the minimum distance.
- Medicine A is assigned to group 1, medicine B to group 2, medicine C to group 2 and medicine D to group 2.
- The elements of Group matrix below is 1 if and only if the object is assigned to that group.

$$\mathbf{G}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{array}{l} \text{group - 1} \\ \text{group - 2} \end{array}$$

A B C D





- **Iteration-1, Objects-Centroids distances :**
The next step is to compute the distance of all objects to the new centroids.
- Similar to step 2, we have distance matrix at iteration 1 is

$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{array}{ll} c_1 = (1,1) & group - 1 \\ c_2 = \left(\frac{11}{3}, \frac{8}{3}\right) & group - 2 \end{array}$$

$$\begin{array}{cccc} A & B & C & D \\ \begin{bmatrix} 1 & 2 & 4 & 5 \end{bmatrix} & X \\ \begin{bmatrix} 1 & 1 & 3 & 4 \end{bmatrix} & Y \end{array}$$



- **Iteration-1, Objects**

clustering: Based on the new distance matrix, we move the medicine B to Group 1 while all the other objects remain. The Group matrix is shown below

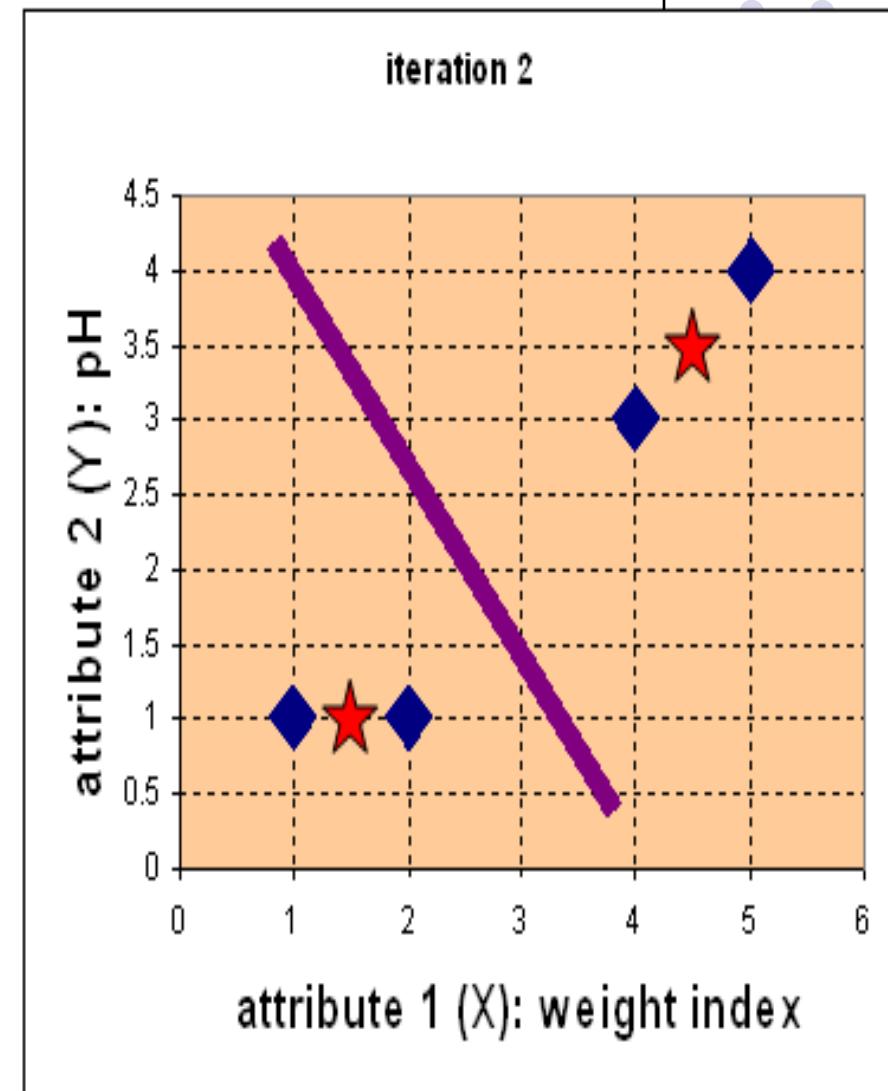
$$G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{array}{l} group - 1 \\ group - 2 \end{array}$$

A B C D

- **Iteration 2, determine**

centroids: Now we repeat step 4 to calculate the new centroids coordinate based on the clustering of previous iteration. Group 1 and group 2 both has two members, thus the new centroids are $c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1\frac{1}{2}, 1)$

and $c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4\frac{1}{2}, 3\frac{1}{2})$





- **Iteration-2, Objects-Centroids distances :**
Repeat step 2 again, we have new distance matrix at iteration 2 as

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{array}{ll} \mathbf{c}_1 = (1\frac{1}{2}, 1) & \text{group - 1} \\ \mathbf{c}_2 = (4\frac{1}{2}, 3\frac{1}{2}) & \text{group - 2} \end{array}$$
$$\begin{array}{cccc} A & B & C & D \end{array}$$
$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{array}{ll} X \\ Y \end{array}$$



- **Iteration-2, Objects clustering:** Again, we assign each object based on the minimum distance.

$$\mathbf{G}^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{array}{l} group - 1 \\ group - 2 \end{array}$$

A B C D

- We obtain result that $\mathbf{G}^2 = \mathbf{G}^1$. Comparing the grouping of last iteration and this iteration reveals that the objects does not move group anymore.
- Thus, the computation of the k-mean clustering has reached its stability and no more iteration is needed..



We get the final grouping as the results as:

<u>Object</u>	<u>Feature1(X): weight index</u>	<u>Feature2 (Y): pH</u>	<u>Group (result)</u>
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2



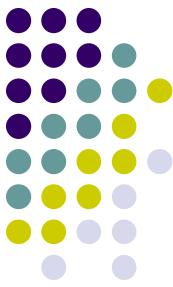
Weaknesses of K-Mean Clustering

1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The number of cluster, K , must be determined before hand. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments.
3. We never know the real cluster, using the same data, because if it is inputted in a different order it may produce different cluster if the number of data is few.
4. It is sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the *local optimum*.



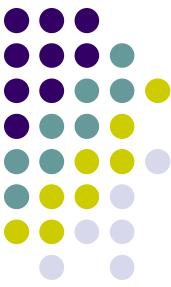
Applications of K-Mean Clustering

- It is relatively *efficient and fast*. It computes result at **O(tkn)**, where n is number of objects or points, k is number of clusters and t is number of iterations.
- k-means clustering can be applied to *machine learning or data mining*
- Used on acoustic data in speech understanding to convert waveforms into one of k categories (known as Vector Quantization or Image Segmentation).
- Also used for choosing color palettes on old fashioned graphical display devices and Image Quantization.



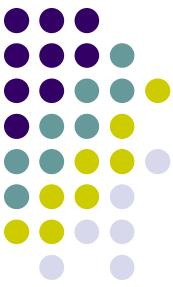
CONCLUSION

- *K-means algorithm* is useful for undirected knowledge discovery and is relatively simple. K-means has found wide spread usage in lot of fields, ranging from unsupervised learning of neural network, Pattern recognitions, Classification analysis, Artificial intelligence, image processing, machine vision, and many others.



References

- [Tutorial](#) - Tutorial with introduction of Clustering Algorithms (k-means, fuzzy-c-means, hierarchical, mixture of gaussians) + some interactive demos (java applets).
- Digital Image Processing and Analysis-by B.Chanda and D.Dutta Majumdar.
- H. Zha, C. Ding, M. Gu, X. He and H.D. Simon. "Spectral Relaxation for K-means Clustering", Neural Information Processing Systems vol.14 (NIPS 2001). pp. 1057-1064, Vancouver, Canada. Dec. 2001.
- J. A. Hartigan (1975) "Clustering Algorithms". Wiley.
- J. A. Hartigan and M. A. Wong (1979) "A K-Means Clustering Algorithm", Applied Statistics, Vol. 28, No. 1, p100-108.
- [D. Arthur, S. Vassilvitskii](#) (2006): "How Slow is the k-means Method?,"
- D. Arthur, S. Vassilvitskii: "[k-means++ The Advantages of Careful Seeding](#)" 2007 Symposium on Discrete Algorithms (SODA).
- www.wikipedia.com



Thank You

Expectation Maximization for Clustering

Dr. S. Suresh
Assistant Professor
Department of Computer Science
Banaras Hindu University

Outline

- Problems with K-means
- Mixture Models
- Expectation Maximization algorithm for clustering

K-means

Start with some initial cluster centers

Iterate:

- Assign/cluster each example to closest center
- Recalculate centers as the mean of the points in a cluster

Problems with K-means

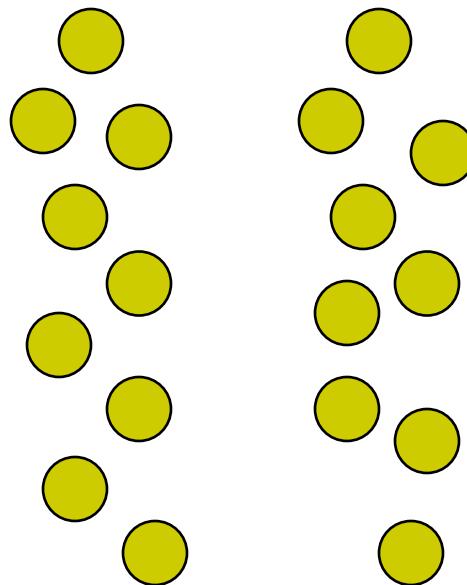
Determining K is challenging

Spherical assumption about the data (distance to cluster center)

Hard clustering isn't always right

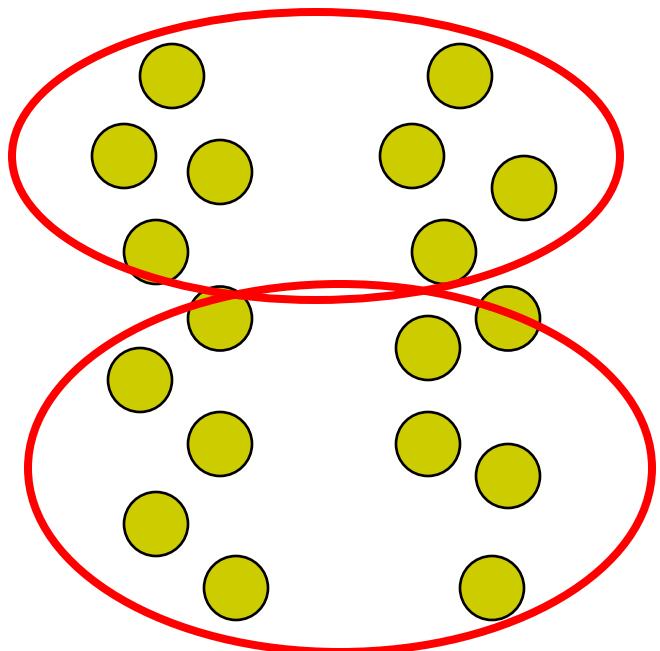
Greedy approach

Problems with K-means



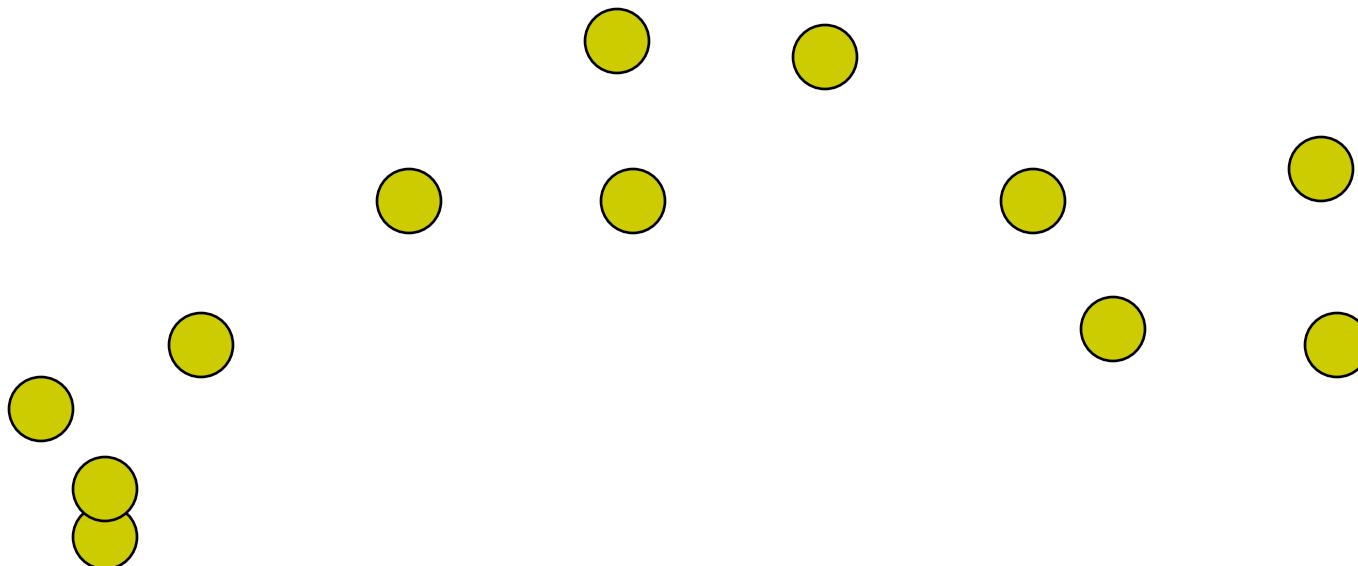
What would K-means give us here?

Assumes spherical clusters

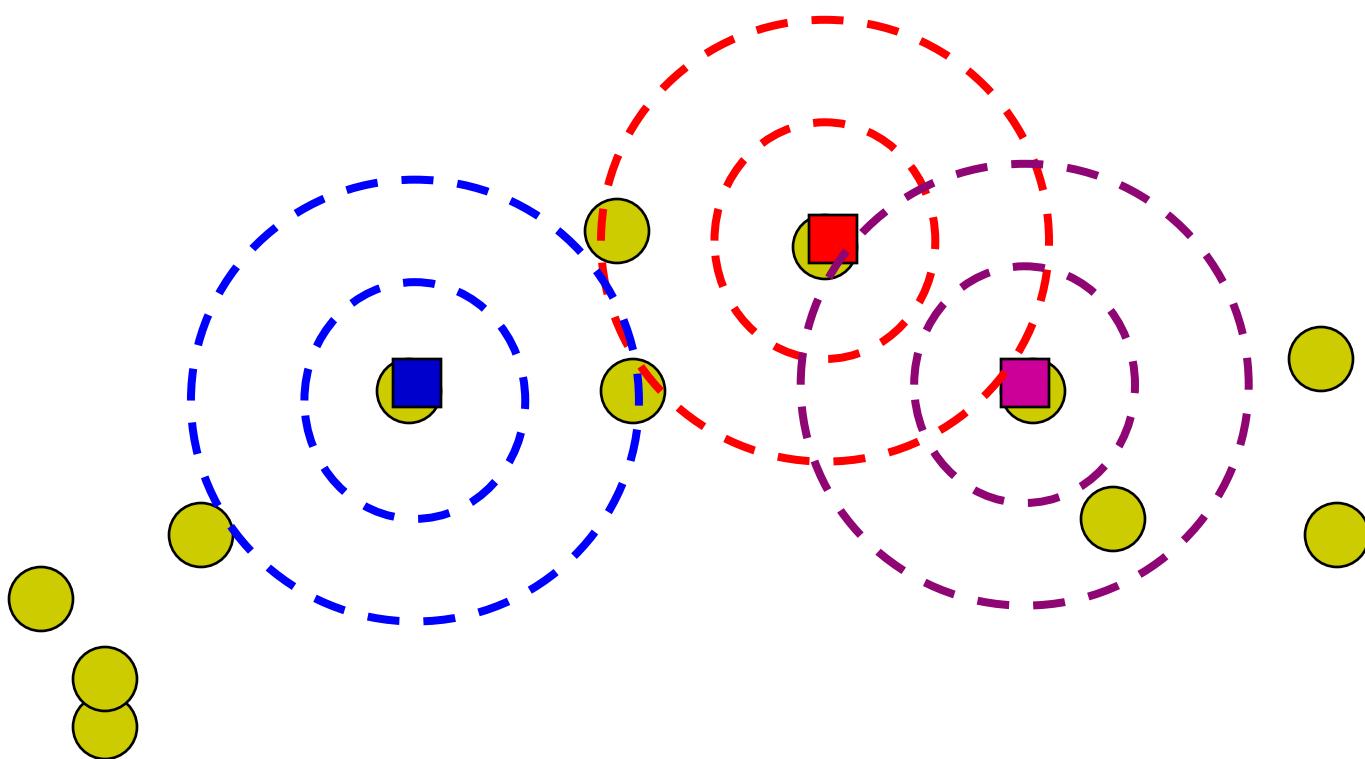


k-means assumes spherical clusters!

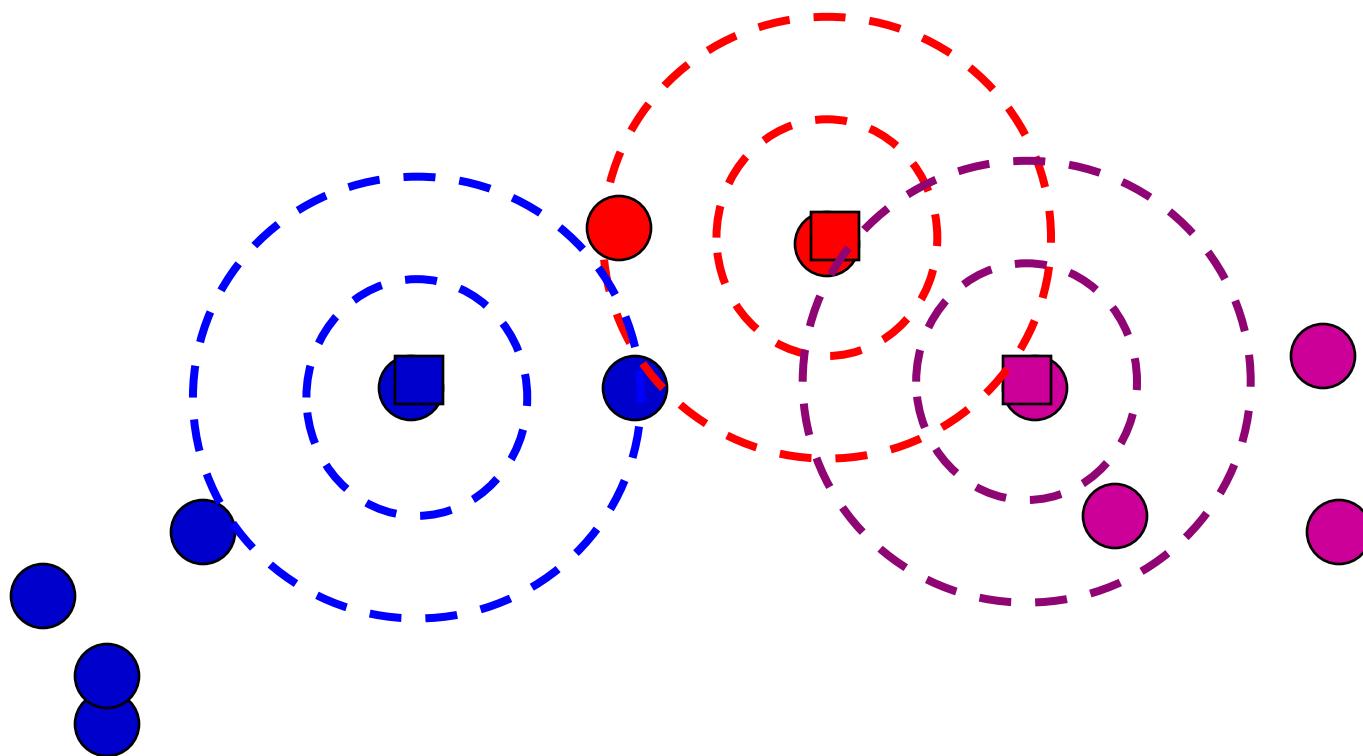
K-means: another view



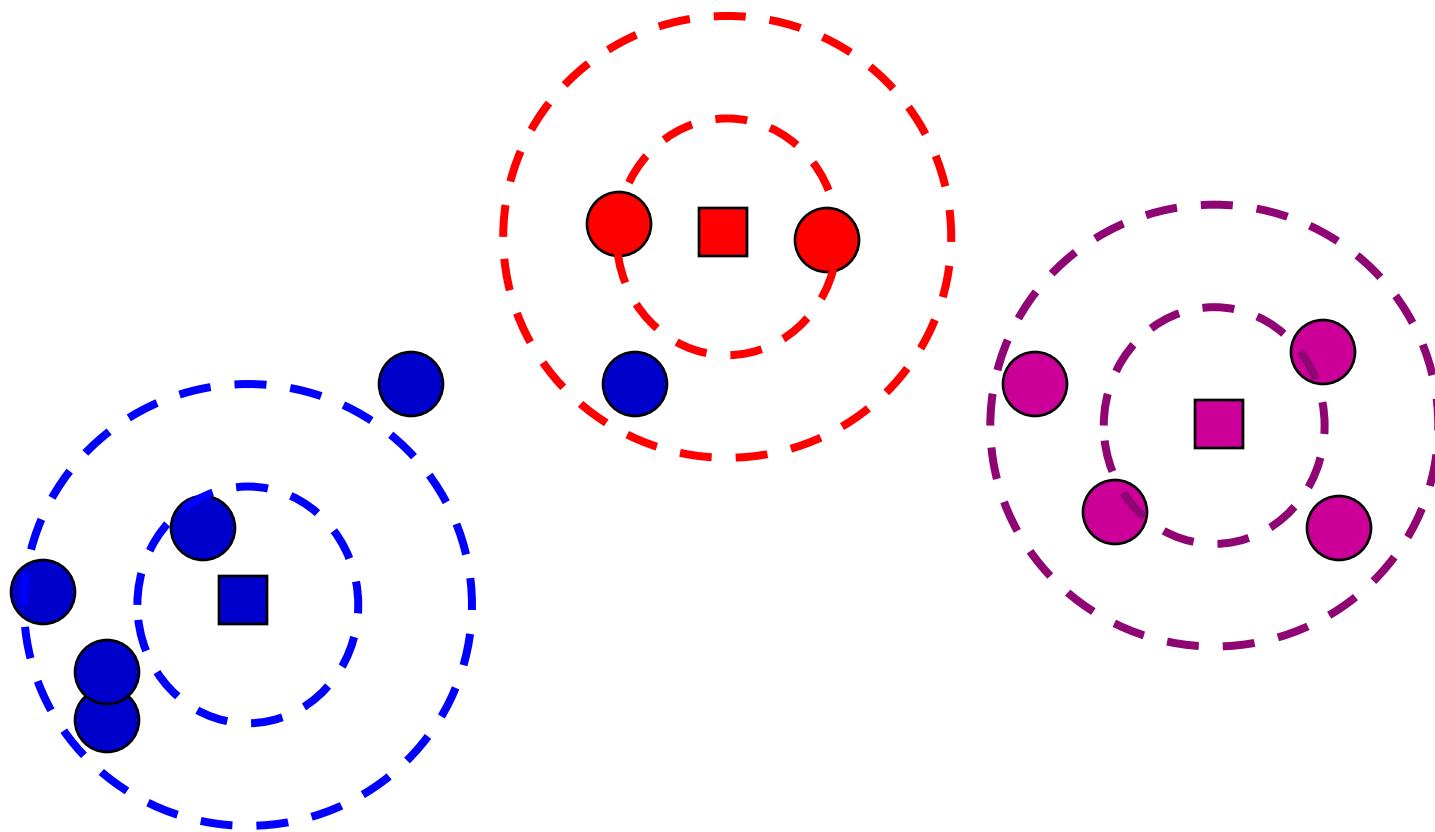
K-means: another view



K-means: assign points to nearest center



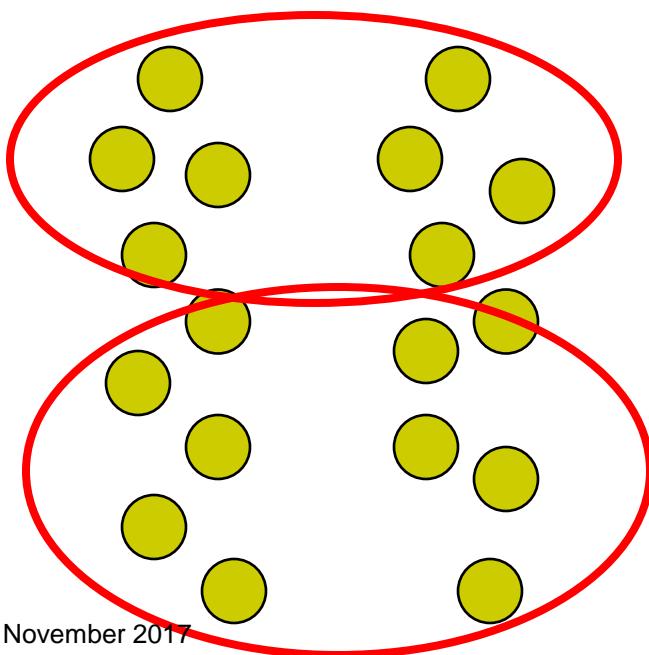
K-means: readjust centers



EM clustering: mixtures of Gaussians

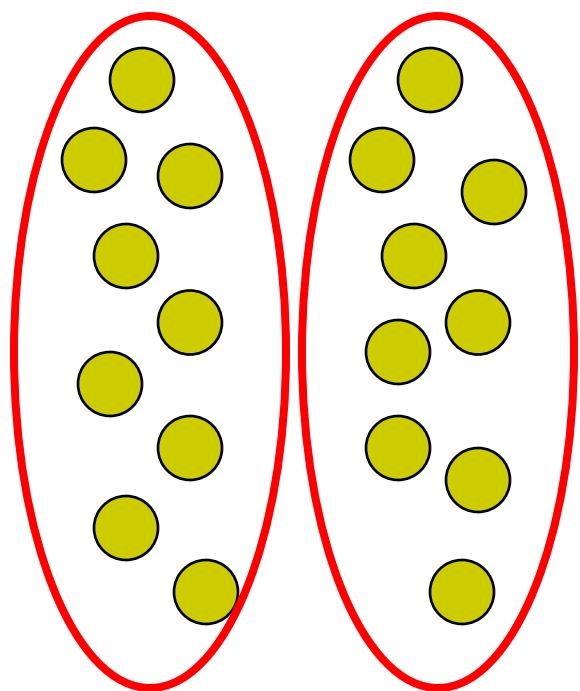
Assume data came from a mixture of Gaussians (**elliptical data**),
assign data to cluster with a certain *probability*

k-means



8 November 2017

EM



CS304 - Machine Learning

11

The Problem

- You have data that you believe is drawn from **n** populations
- You want to identify parameters for each population
- You don't know anything about the populations *a priori*
 - Except you believe that they're Gaussian...

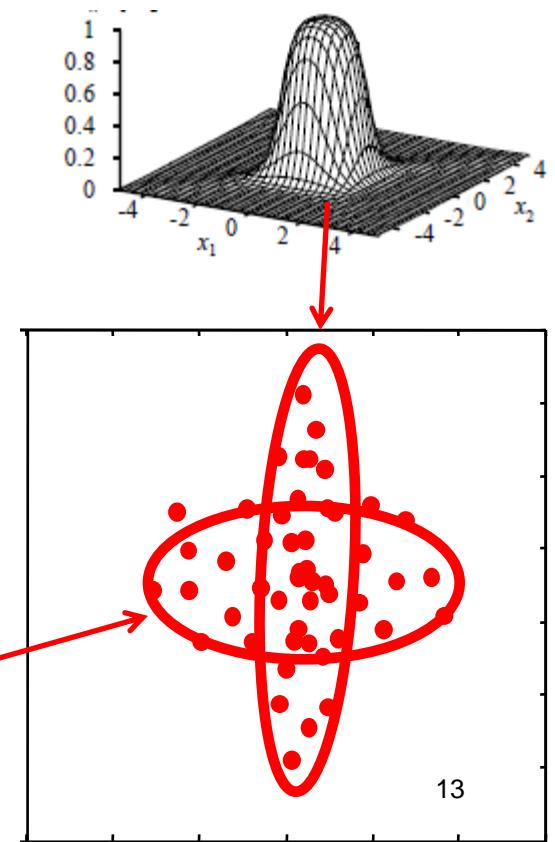
Mixtures of Gaussians

□ K-means algorithm

- Assigned each example to exactly one cluster
- What if clusters are overlapping?
 - Hard to tell which cluster is right
 - Maybe we should try to remain uncertain
- Used Euclidean distance
- What if cluster has a non-circular shape?

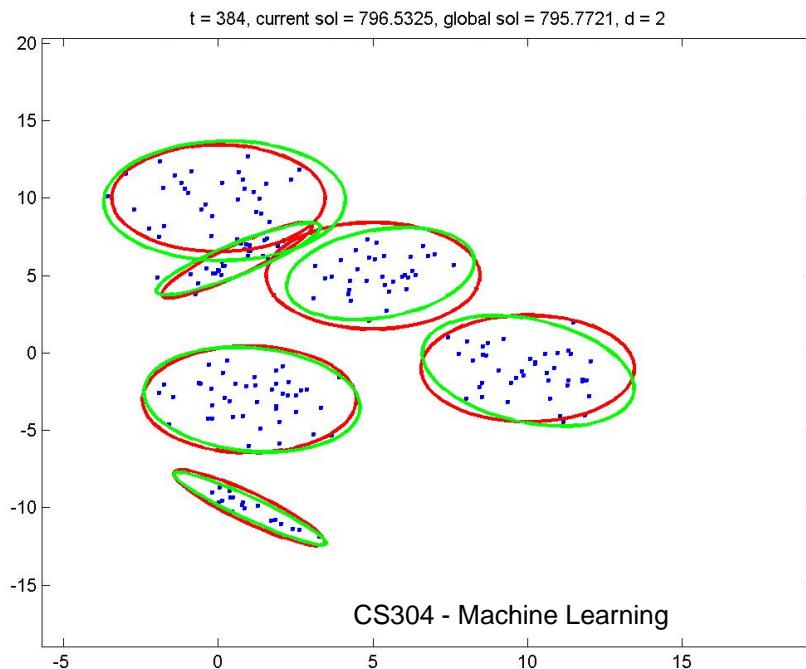
□ Gaussian mixture models

- Clusters modeled as multivariate Gaussians
 - Not just by their mean
- EM algorithm: assign data to cluster with some *probability*



Gaussian Mixture Models

- Rather than identifying clusters by “nearest” centroids
- Fit a Set of k Gaussians to the data
- Maximum Likelihood over a mixture model



Mixture Models

-
- A **mixture model** is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs.
 - Formally a mixture model corresponds to the mixture distribution that represents the probability distribution of observations in the overall population.

Mixture Models

- Formally a Mixture Model is the weighted sum of a number of pdfs where the weights are determined by a distribution, π

$$p(x) = \pi_0 f_0(x) + \pi_1 f_1(x) + \pi_2 f_2(x) + \dots + \pi_k f_k(x)$$

where $\sum_{i=0}^k \pi_i = 1$

$$p(x) = \sum_{i=0}^k \pi_i f_i(x)$$

Gaussian Mixture Models

- GMM: the weighted sum of a number of Gaussians where the weights are determined by a distribution, π

$$p(x) = \pi_0 N(x|\mu_0, \Sigma_0) + \pi_1 N(x|\mu_1, \Sigma_1) + \dots + \pi_k N(x|\mu_k, \Sigma_k)$$

$$\text{where } \sum_{i=0}^k \pi_i = 1$$

$$p(x) = \sum_{i=0}^k \pi_i N(x|\mu_k, \Sigma_k)$$

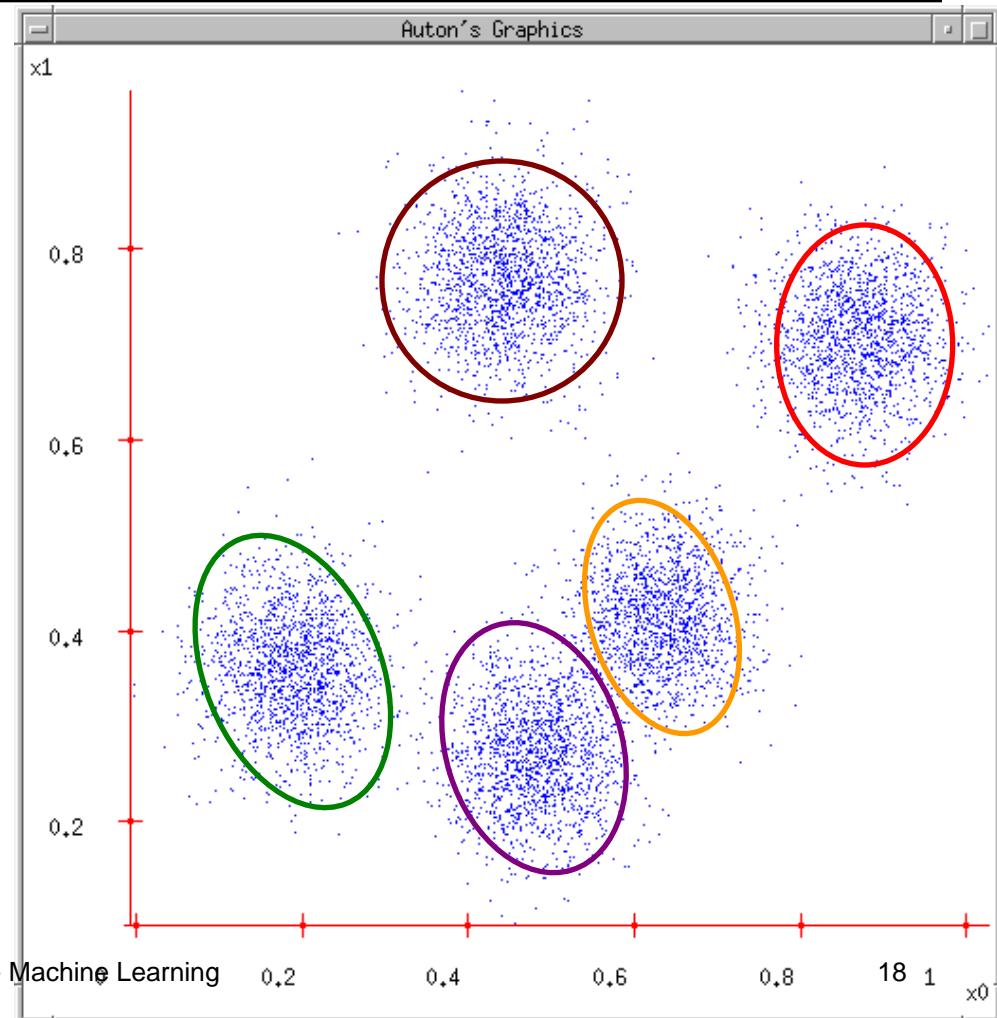
A Gaussian Mixture Model for Clustering

- Assume that data are generated from a mixture of Gaussian distributions
- For each Gaussian distribution
 - Center: μ_i
 - Variance: Σ_i (ignore)
- For each data point
 - Determine membership

z_{ij} : if x_i belongs to j-th cluster

8 November 2017

CS304 - Machine Learning



Learning a Gaussian Mixture (with known covariance)

□ Probability $p(x = x_i)$

$$p(x = x_i) = \sum_{\mu_j} p(x = x_i, \mu = \mu_j) = \sum_{\mu_j} p(\mu = \mu_j) p(x = x_i | \mu = \mu_j)$$

The probability density of the normal distribution is:

$$f(x | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where:

- μ is the mean or expectation of the distribution (and also its median and mode).
- σ is the standard deviation
- σ^2 is the variance

Learning a Gaussian Mixture (with known covariance)

□ Probability $p(x = x_i)$

$$\begin{aligned} p(x = x_i) &= \sum_{\mu_j} p(x = x_i, \mu = \mu_j) = \sum_{\mu_j} p(\mu = \mu_j) p(x = x_i | \mu = \mu_j) \\ &= \sum_{\mu_j} p(\mu = \mu_j) \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|x_i - \mu_j\|_2}{2\sigma^2}\right) \end{aligned}$$

□ Log-likelihood of data

$$\sum_i \log p(x = x_i) = \sum_i \log \left[\sum_{\mu_j} p(\mu = \mu_j) \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|x_i - \mu_j\|_2}{2\sigma^2}\right) \right]$$

8 November 2017 □ Apply MLE to find optimal parameters $\{p(\mu = \mu_j), \mu_j\}_j$

Learning a Gaussian Mixture

(with known covariance)

E-Step

$$E[z_{ij}] = p(\mu = \mu_j \mid x = x_i)$$

$$\begin{aligned} &= \frac{p(x = x_i \mid \mu = \mu_j) p(\mu = \mu_j)}{\sum_{n=1}^k p(x = x_i \mid \mu = \mu_n) p(\mu = \mu_j)} \\ &= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2} p(\mu = \mu_j)}{\sum_{n=1}^k e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2} p(\mu = \mu_n)} \end{aligned}$$

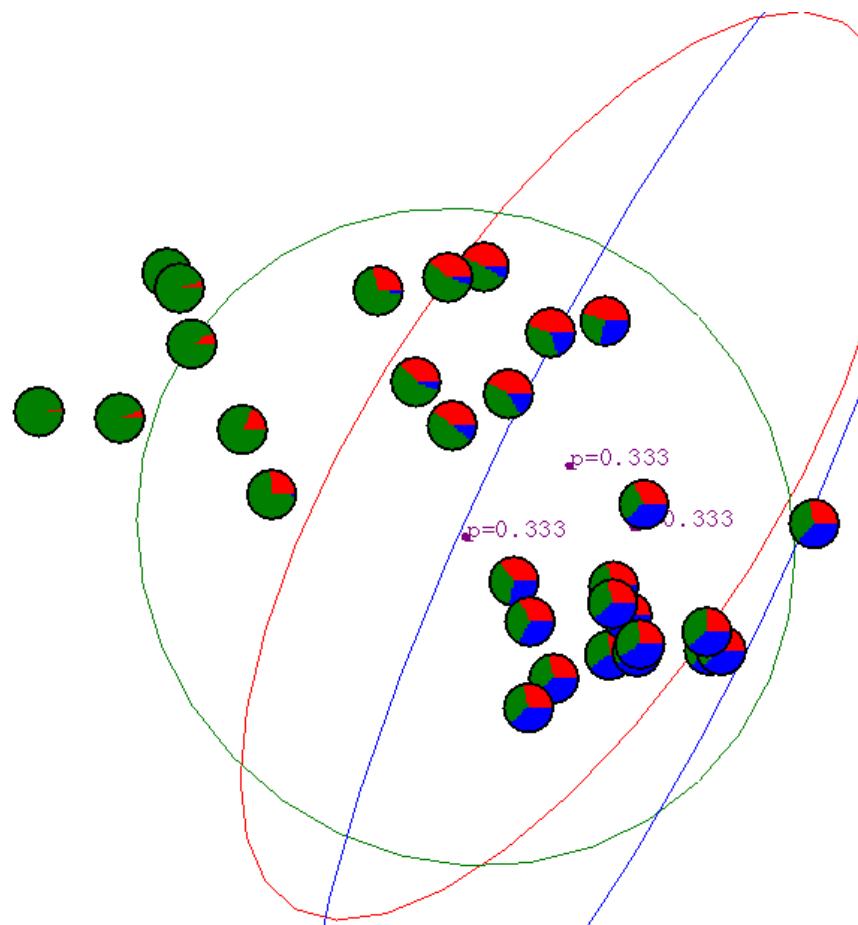
Learning a Gaussian Mixture (with known covariance)

M-Step

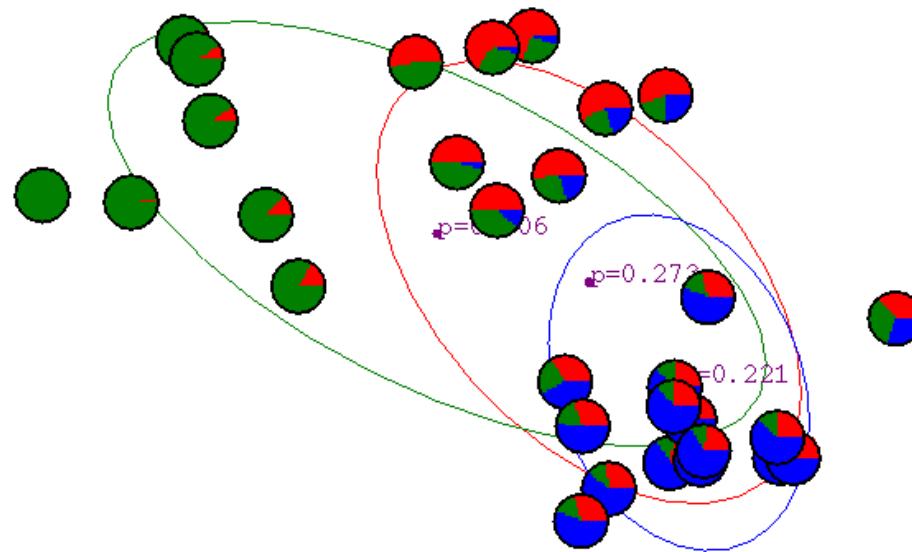
$$\mu_j \leftarrow \frac{1}{\sum_{i=1}^m E[z_{ij}]} \sum_{i=1}^m E[z_{ij}]x_i$$

$$p(\mu = \mu_j) \leftarrow \frac{1}{m} \sum_{i=1}^m E[z_{ij}]$$

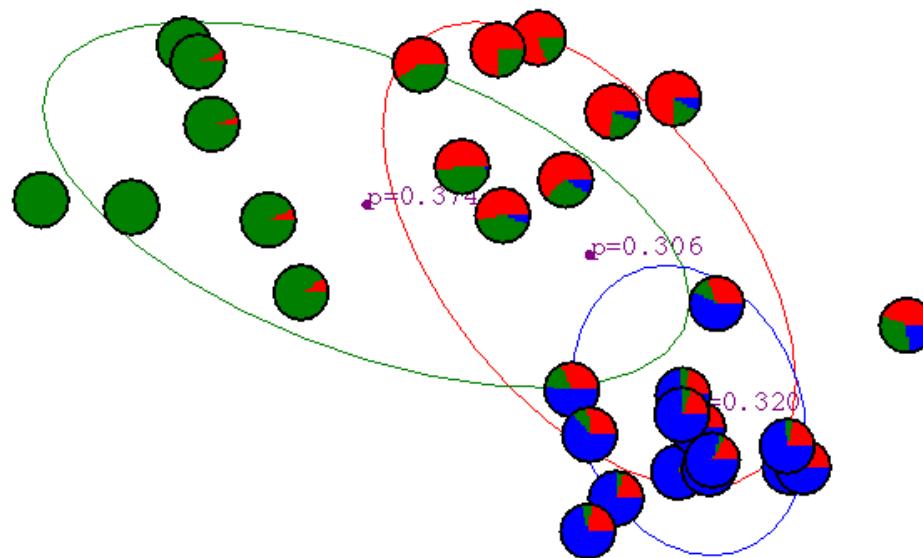
Gaussian Mixture Example: Start



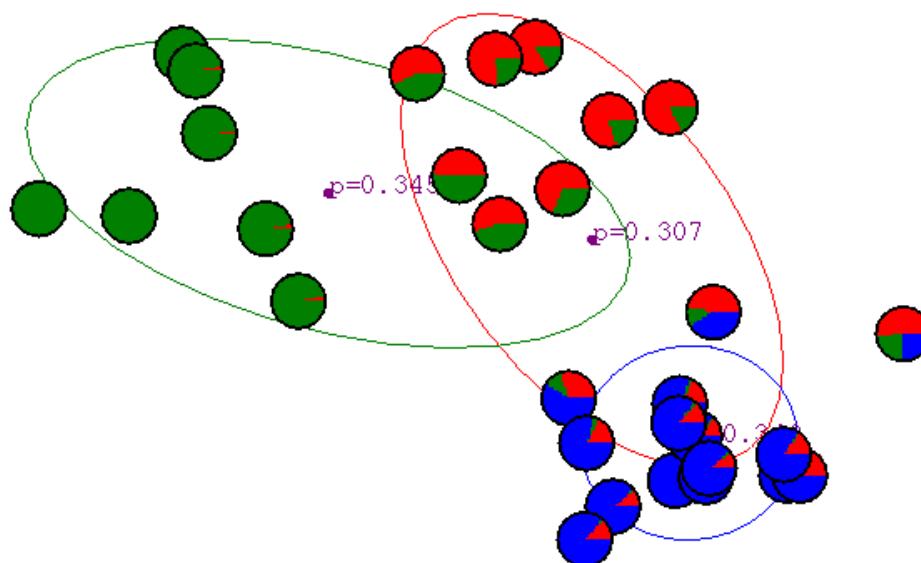
After First Iteration



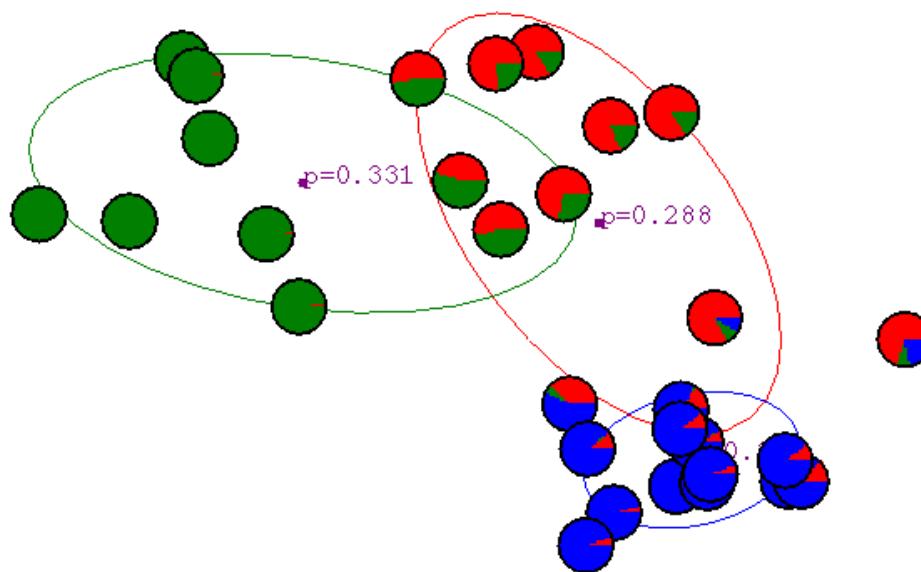
After 2nd Iteration



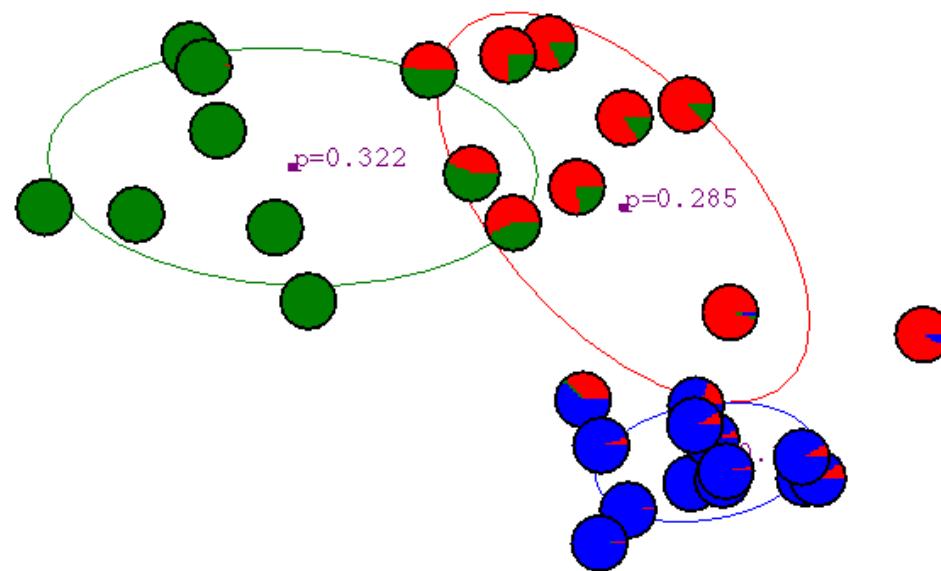
After 3rd Iteration



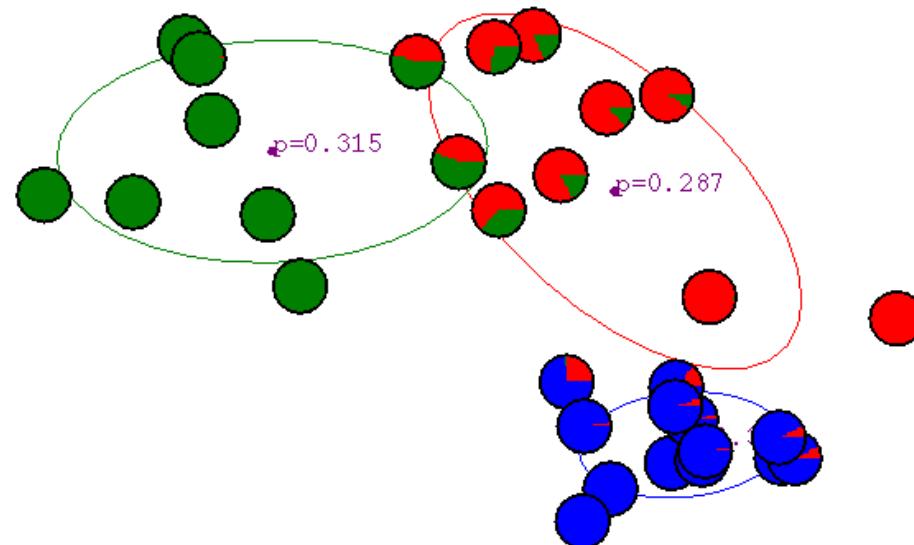
After 4th Iteration



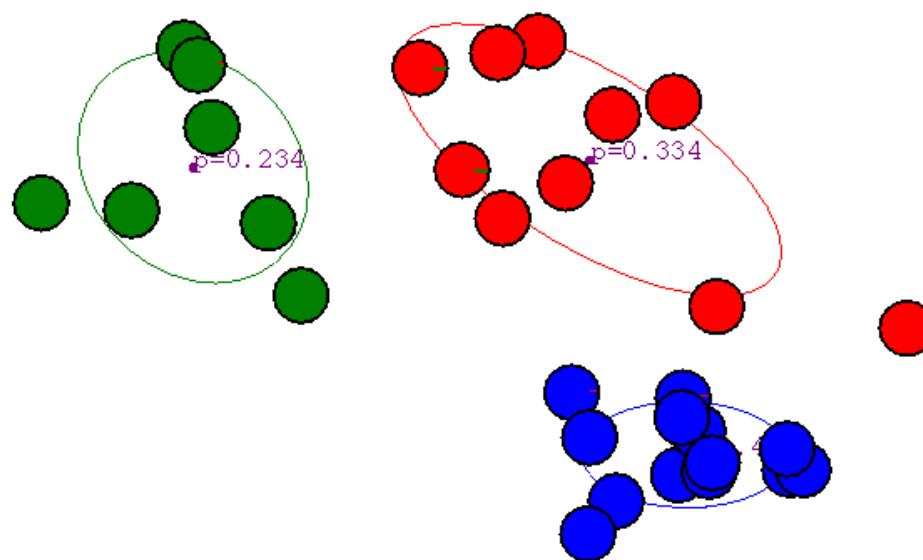
After 5th Iteration



After 6th Iteration



After 20th Iteration



Mixture Model for Doc Clustering

- A set of language models $\Theta = \{\theta_1, \theta_2, \dots, \theta_K\}$
 - $\theta_i = \{p(w_1 | \theta_i), p(w_2 | \theta_i), \dots, p(w_V | \theta_i)\}$

Mixture Model for Doc Clustering

- A set of language models $\Theta = \{\theta_1, \theta_2, \dots, \theta_K\}$
 - $\theta_i = \{p(w_1 | \theta_i), p(w_2 | \theta_i), \dots, p(w_V | \theta_i)\}$
- Probability $p(d = d_i)$

$$p(d = d_i) = \sum_{\theta_j} p(d = d_i, \theta = \theta_j)$$

Mixture Model for Doc Clustering

- A set of language models $\Theta = \{\theta_1, \theta_2, \dots, \theta_K\}$
 - $\theta_i = \{p(w_1 | \theta_i), p(w_2 | \theta_i), \dots, p(w_V | \theta_i)\}$
- Probability $p(d = d_i)$

$$\begin{aligned} p(d = d_i) &= \sum_{\theta_j} p(d = d_i, \theta = \theta_j) \\ &= \sum_{\theta_j} p(\theta = \theta_j) p(d = d_i | \theta = \theta_j) \end{aligned}$$

Mixture Model for Doc Clustering

- A set of language models

- $\theta_i = \{p(w_1 | \theta_i), p(w_2 | \theta_i), \dots\}$

Introduce hidden variable z_{ij}

z_{ij} : document d_i is generated by the j -th language model θ_j .

- Probability $p(d = d_i)$

$$p(d = d_i) = \sum_{\theta_j} p(d = d_i, \theta = \theta_j)$$

$$= \sum_{\theta_j} p(\theta = \theta_j) p(d = d_i | \theta = \theta_j)$$

$$\propto \sum_{\theta_j} p(\theta = \theta_j) \prod_{k=1}^V \left[p(w_k | \theta_j) \right]^{tf(w_k, d_i)}$$

Learning a Mixture Model

E-Step

$$\begin{aligned} E[z_{ij}] &= p(\theta = \theta_j \mid d = d_i) \\ &= \frac{p(d = d_i \mid \theta = \theta_j)p(\theta = \theta_j)}{\sum_{n=1}^K p(d = d_i \mid \theta = \theta_n)p(\theta = \theta_n)} \\ &= \frac{\prod_{m=1}^V [p(w_m \mid \theta_j)]^{tf(w_k, d_i)} p(\theta = \theta_j)}{\sum_{n=1}^K \prod_{m=1}^V [p(w_m \mid \theta_n)]^{tf(w_k, d_i)} p(\theta = \theta_n)} \end{aligned}$$

K: number of language models

Learning a Mixture Model

M-Step

$$p(w_i | \theta_j) \leftarrow \frac{\sum_{k=1}^N E[z_{ij}] \ tf(w_i, d_k)}{\sum_{k=1}^N E[z_{ij}] |d_k|}$$

$$p(\theta = \theta_j) \leftarrow \frac{1}{N} \sum_{i=1}^N E[z_{ij}]$$

N: number of documents

Examples of Mixture Models

“segment 1”	“segment 2”	“matrix 1”	“matrix 2”	“line 1”	“line 2”
imag SEGMENT texture color tissue brain slice cluster mri volume	speaker speech recogni signal train hmm source speakerind. SEGMENT sound	robust MATRIX eigenvalu uncertaini plane linear condition perturb root suffici	manufactur cell part MATRIX cellular famili design machinepart format group	constraint LINE match locat imag geometr impos segment fundament recogn	alpha redshift LINE galaxi quasar absorp high ssup densiti veloc



Application (I): Search Result Clustering

company | products | solutions | customers | demos | partners | press

LYCOS

jaguar

Search Clustering by Vivisimo

► Other demos ► Help! ► Tell us what you think!

Clustered Results

Top 185 results retrieved for the query **jaguar** ([Details](#))

1. [Jaguar Cars](#) [new window] [frame] [preview]
Official worldwide web site of **Jaguar** Cars. Gama actual, concesionarios, historia, noticias, anuncios y servicios fina
URL: www.jaguar.com - show in clusters
Sources: Lycos 1

2. [Jaguar Cars](#) [new window] [frame] [preview]
URL: www.jaguarcars.com - show in clusters
Sources: Lycos 2, Lycos 59, Lycos 90, Lycos 97, Lycos 99

3. [www.jaguar-racing.com](#) [new window] [frame] [preview]
URL: www.jaguar-racing.com - show in clusters
Sources: Lycos 3, Lycos 93, Lycos 115

4. [Jaguar Cars](#) [new window] [frame] [preview]
United States United Kingdom Germany Japan France Italy Spain...
URL: www.jaguarvehicles.com - show in clusters
Sources: Lycos 4, Lycos 8, Lycos 41, Lycos 102, Lycos 188

5. [Apple - Mac OS X](#) [new window] [frame] [preview]
... queries to find your stuff, refining the list as you narrow options. Sure you could quantify that as up to six times far
Jaguar, but youll probably think Panthers done almost before you...
URL: www.apple.com/macosx - Machine Learning
Sources: Lycos 5

Find in clusters:

Enter Keywords

Application (II): Navigation

Entertainment in the Yahoo! Directory - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://dir.yahoo.com/Entertainment/ Google

Getting Started Latest Headlines

Yahoo! My Yahoo! Mail Welcome, Guest [Sign In]

YAHOO! DIRECTORY

Search: the Web | the Directory | this category

Entertainment Email this page Suggest a Site Advanced Search

Directory > Entertainment

SPONSOR RESULTS

 [Value City Furniture](#)
www.vcf.com Quality Home Entertainment Packages Browse Today and Find a Store.

CATEGORIES ([What's This?](#))

Top Categories

- ◆ [Music](#) (76772) NEW!
- ◆ [Actors](#) (19211) NEW!
- ◆ [Movies and Film](#) (40031) NEW!
- ◆ [Television Shows](#) (17085) NEW!
- ◆ [Humor](#) (3927)
- ◆ [Comics and Animation](#) (5778) NEW!

Additional Categories

- ◆ [Amusement and Theme Parks](#) (449)
- ◆ [Awards](#) (698)
- ◆ [Blogs@](#)
- ◆ [Books and Literature@](#)
- ◆ [Chats and Forums](#) (47)
- ◆ [Comedy](#) (1730)
- ◆ [Consumer Electronics](#) (1355) NEW!
- ◆ [Magic](#) (353)
- ◆ [News and Media](#) (443)
- ◆ [Organizations](#) (33)
- ◆ [Performing Arts@](#)
- ◆ [Radio@](#)
- ◆ [Randomized Things](#) (57)
- ◆ [Reviews](#) (32)
- ◆ [Software](#) (100)

SPONSOR RESULTS

 [Entertainment Center Furniture](#)
Save 30-60% On A Variety Of Furniture For Any Room Thru 11/13.
JCPenney.com

 [Studiotech Official Site](#)
StudioTech Entertainment Furniture. Factory Direct...
www.StudioTech.com

 [Bush Entertainment Furniture](#)
Save up to 50% factory-direct.
www.bushfurniturecollection.com

8 November 2017 CS304 - Machine Learning

Done

start Firefox jie... W... S... ar... Mi... 1... II... E... EN 11:46 AM

Application (III): Google News

Google News - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Getting Started Latest Headlines

Web Images Maps News Shopping Gmail more ▾ Sign in

http://news.google.com/nwshp?hl=en&tab=wn

Google News Search News Search the Web

Search and browse 4,500 news sources updated continuously.

News archive search | Advanced news search | Blog search

Top Stories U.S. Go

Fed to Purchase US Commercial Paper to Ease Crunch (Update3)
Bloomberg - 1 hour ago
By Craig Torres Oct. 7 (Bloomberg) -- The Federal Reserve will create a special fund to purchase US commercial paper after the credit crunch threatened to cut off a key source of funding for corporations.
[Treasurys off as Fed plans to buy commercial paper](#) MarketWatch
[Bernanke Goes Commercial](#) Forbes
[CNNMoney.com - Wall Street Journal - Washington Post - Reut](#)
[all 1,291 news articles »](#)

Palin Adds Fannie Mae Execs to List of Objectionable Obama Associates
Washington Post - 1 hour ago
By Perry Bacon Jr. JACKSONVILLE, Fla. -- William Ayers isn't the only one of Barack Obama's supporters Alaska Gov. Sarah Palin finds objectionable.
[Video: AP Campaign Minute](#) AssociatedPress
[Forget Palin, McCain needs Peyton](#) Christian Science Monitor
[CNN International - Atlantic Online - The Weekly Standard - Newsweek](#)
[all 5,708 news articles »](#)

Thai Army Sends Troops to Help Police Keep Peace
Washington Post - 1 hour ago
Thailand's military agreed Tuesday to deploy hundreds of uniformed soldiers to the streets of Bangkok to help police restore order after

AMD finds oil money to finance its split into two companies
New York Times - [all 521 news articles »](#)

Debate Preview: Town Hall Format May Not Help John McCain This Time
U.S. News & World Report - [all 655 news articles »](#)

Physicists Share Nobel Prize For Work On Subatomic Checks out of Rehab for Sex
[watch \[简明英汉词典\]](#)
[\[wɔtʃ\]](#)
n.注视,注意,手表,看守,守护,监视,值班人
vt.看,注视,照顾,监视,警戒,守护,看守
vi.观看,注视,守候
adj.手表的,挂表的

Red Sox Are on Display in Postseason
New York Times - [all 3,429 news articles »](#)

Kenya deports anti-Obama author
Christian Science Monitor - [all 423 news articles »](#)

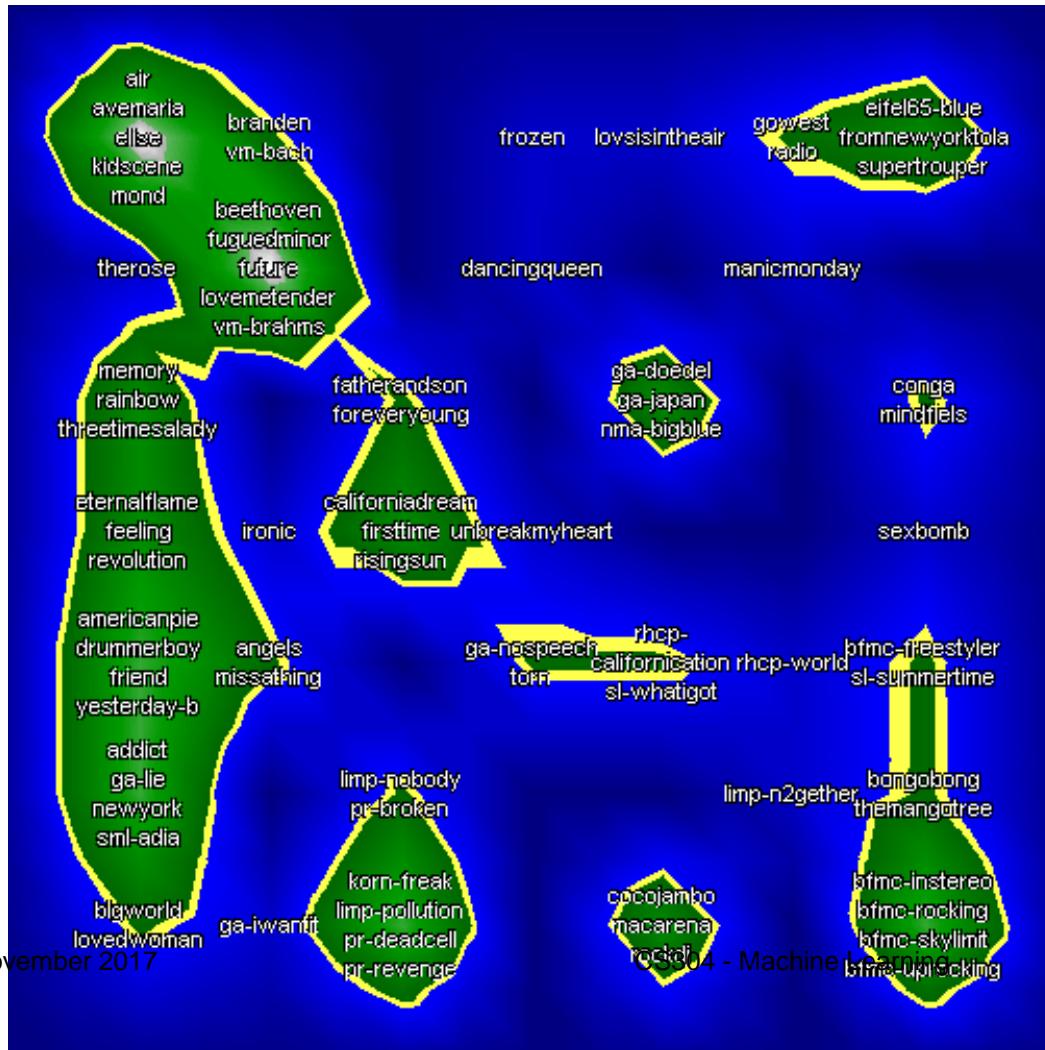
40 hurt when Qantas plane forced to land
United Press International - [all 297 news articles »](#)

In The News

Sarah Palin American League

8 November 2017 CS304 Machine Learning 40

Application (III): Visualization



Islands of music
(Pampalk et al., KDD' 03)

Application (IV): Image Compression

