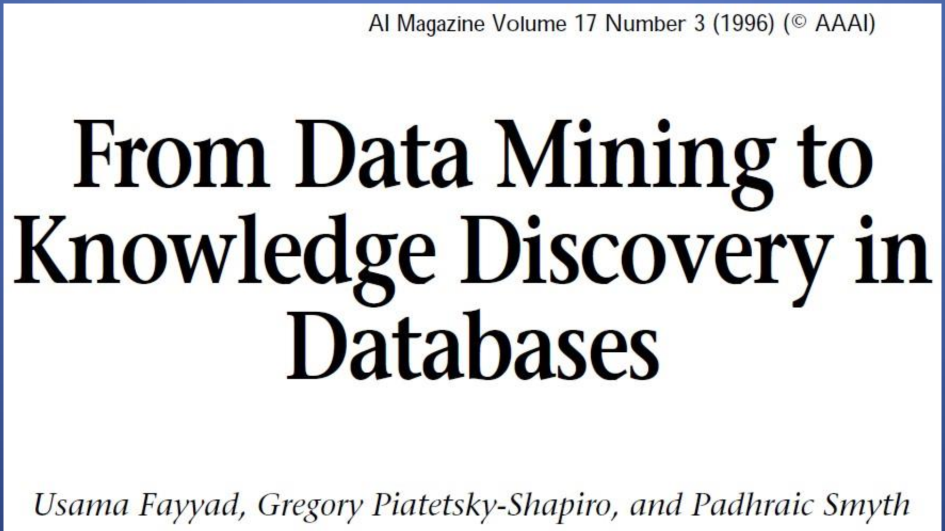
Data Mining, Business Intelligence, *and* Data Science

What is “*Data Mining*”?

Definition

Data mining is the application of specific algorithms for extracting patterns from data. The distinction between the KDD process and the data-mining step (within the process) is a central point…



AI Magazine Volume 17 Number 3 (1996) (© AAAI)

From Data Mining to Knowledge Discovery in Databases

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth

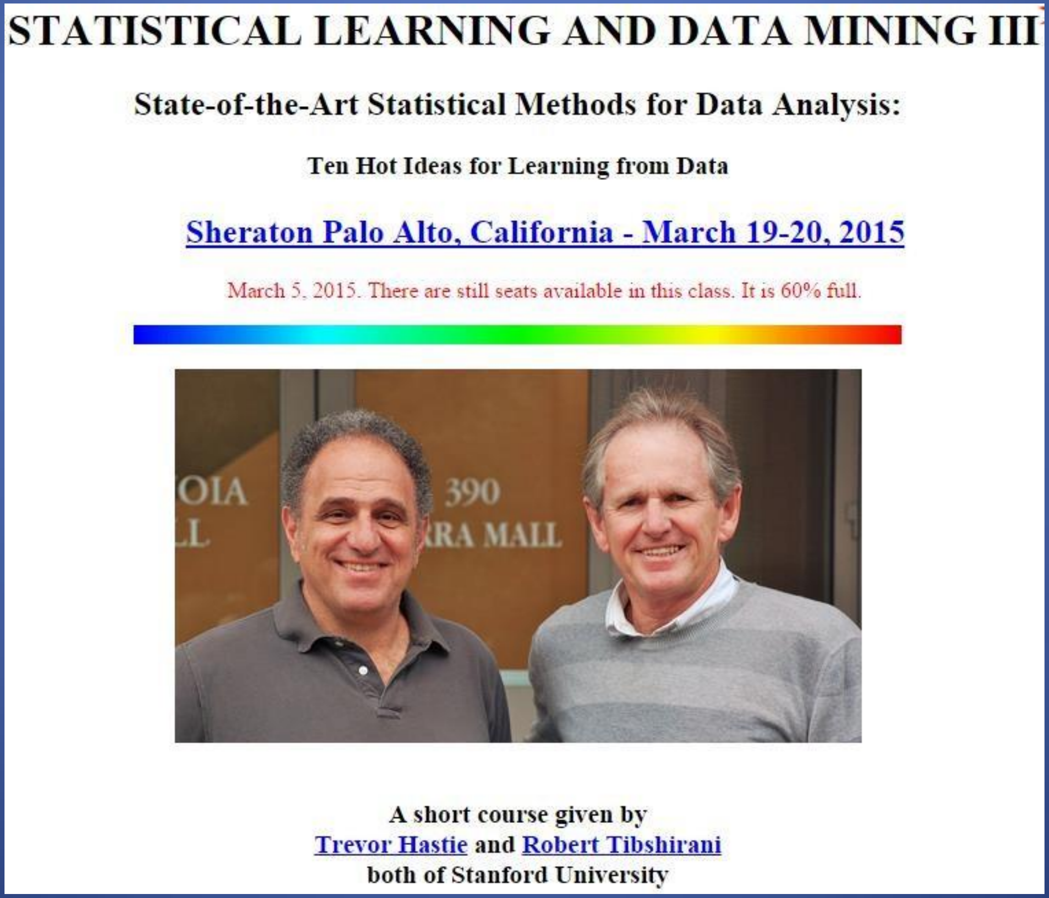
History

"**Data mining**" was introduced in the **1990s**, but data mining is the **evolution of a field** with a long history.

Data mining roots are traced back along *three* family lines:

- classical **statistics**,
- **artificial intelligence**,
- and **machine learning**.

Data Mining & Stats?



STATISTICAL LEARNING AND DATA MINING III

State-of-the-Art Statistical Methods for Data Analysis:

Ten Hot Ideas for Learning from Data

Sheraton Palo Alto, California - March 19-20, 2015

March 5, 2015. There are still seats available in this class. It is 60% full.

A short course given by
Trevor Hastie and Robert Tibshirani
both of Stanford University

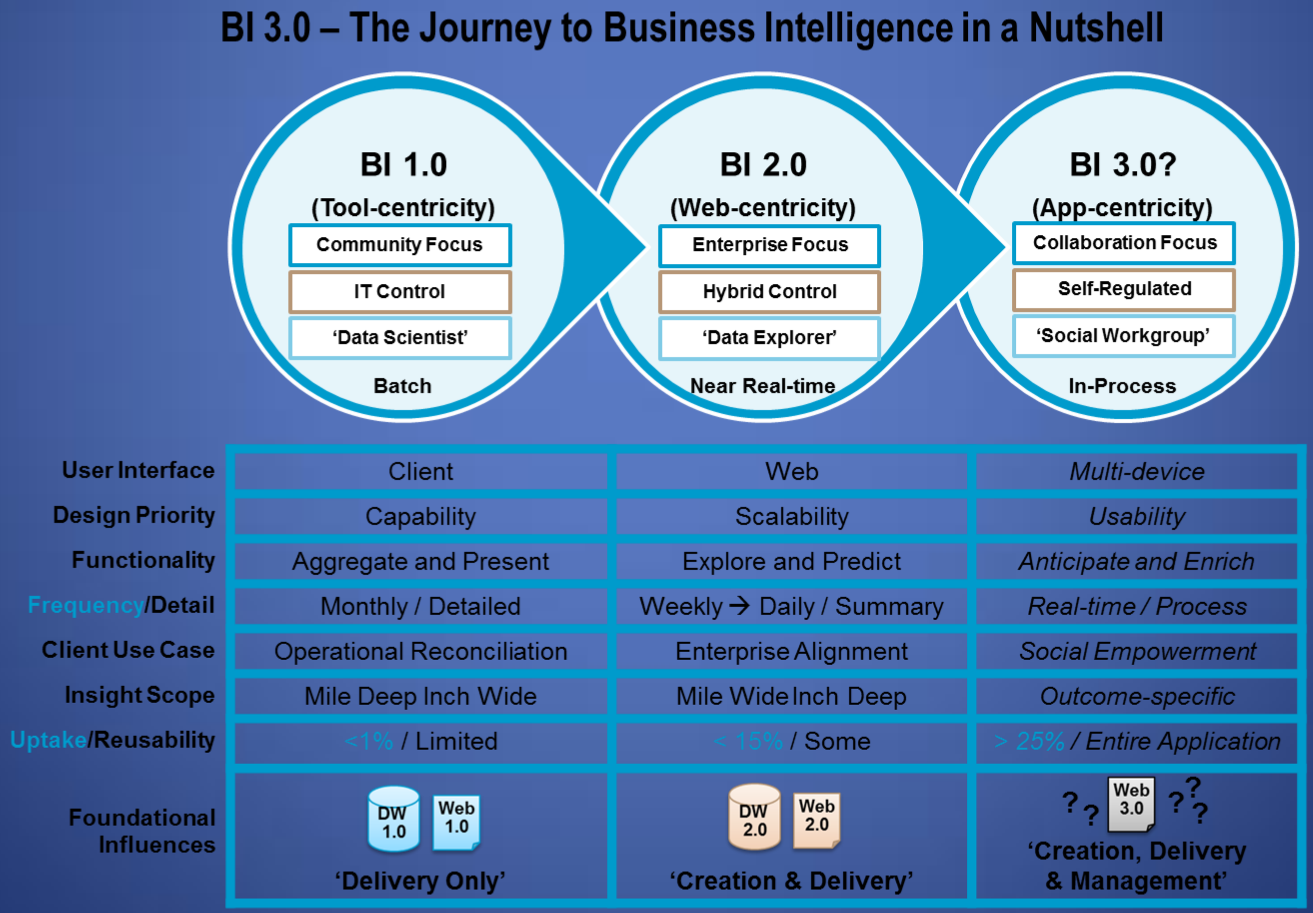
What is *“Business Intelligence”*?

DefinitionS

Gartner.

Business intelligence (BI) is an umbrella term that includes the **applications**, **infrastructure** and **tools**, and best practices that enable access to and analysis of information to improve and optimize decisions and performance.

BI 1.0 - 2.0 - 3.0



BI 3.0 – The Journey to Business Intelligence in a Nutshell

BI 1.0
(Tool-centricity)
- Community Focus
- IT Control
- 'Data Scientist'

Batch

BI 2.0
(Web-centricity)
- Enterprise Focus
- Hybrid Control
- 'Data Explorer'

Near Real-time

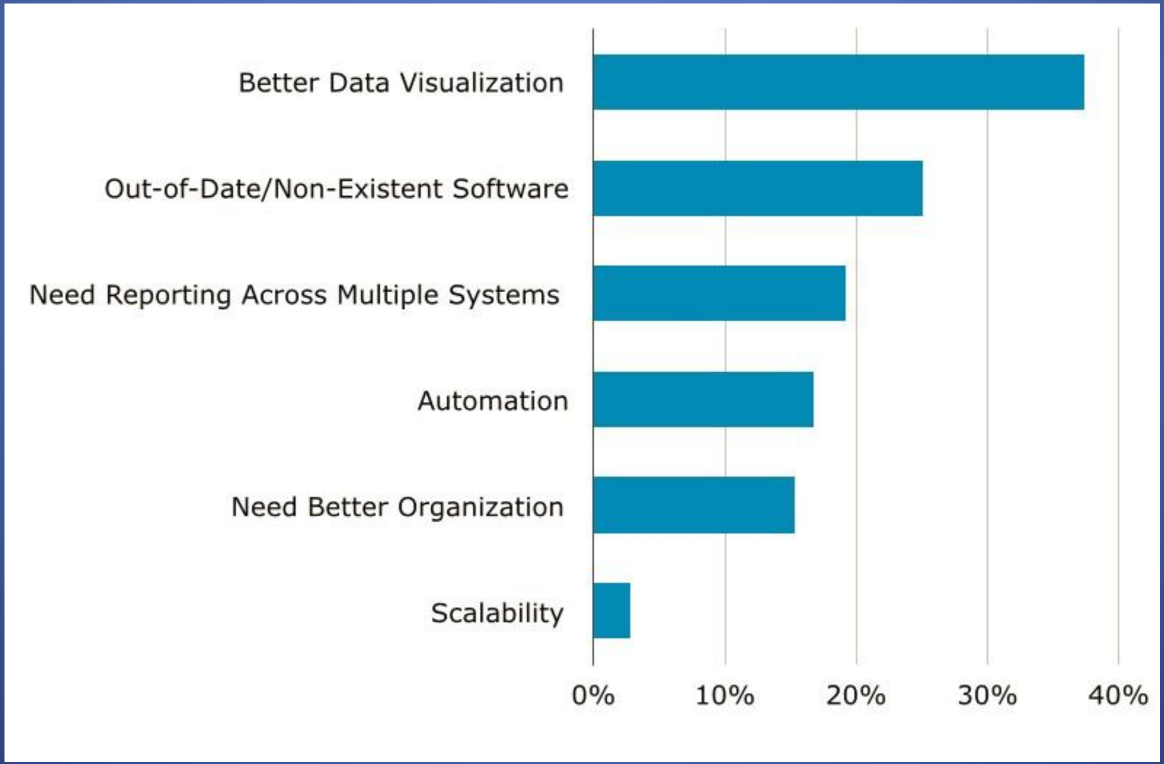
BI 3.0?
(App-centricity)
- Collaboration Focus
- Self-Regulated
- 'Social Workgroup'

In-Process

User Interface	Client	Web	*Multi-device*
Design Priority	Capability	Scalability	*Usability*
Functionality	Aggregate and Present	Explore and Predict	*Anticipate and Enrich*
Frequency/**Detail**	Monthly / Detailed	Weekly → Daily / Summary	*Real-time / Process*
Client Use Case	Operational Reconciliation	Enterprise Alignment	*Social Empowerment*
Insight Scope	Mile Deep Inch Wide	Mile Wide Inch Deep	*Outcome-specific*
Uptake/**Reusability**	<1% / Limited	< 15% / Some	*> 25% / Entire Application*
Foundational Influences	DW 1.0 Web 1.0 **'Delivery Only'**	DW 2.0 Web 2.0 **'Creation & Delivery'**	? ? Web 3.0 ? ? ? **'Creation, Delivery & Management'**

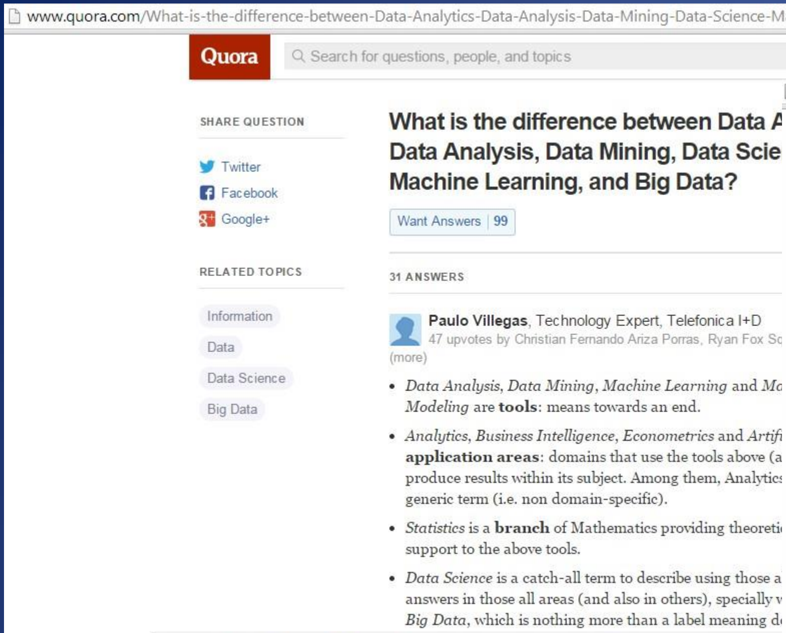
What Business want from BI?

Buyers Overwhelmingly Want Better Data Visualization



What is “*Data Science*”?

http://www.quora.com/What-is-the-difference-between-Data-Analytics-Data-Analysis-Data-Mining-Data-Science-Machine-Learning-and-Big-Data-1



www.quora.com/What-is-the-difference-between-Data-Analytics-Data-Analysis-Data-Mining-Data-Science-Machine-Learning-and-

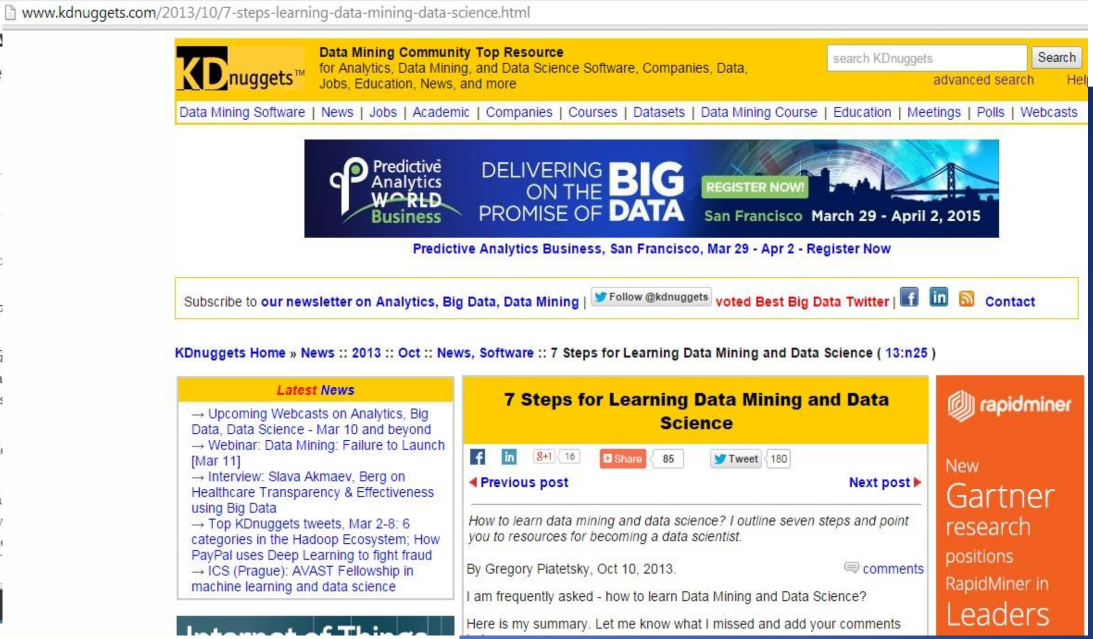
What is the difference between Data A Data Analysis, Data Mining, Data Scie Machine Learning, and Big Data?

Want Answers | 99

31 ANSWERS

Paulo Villegas, Technology Expert, Telefonica I+D
47 upvotes by Christian Fernando Ariza Porras, Ryan Fox Sc
(more)

- *Data Analysis*, *Data Mining*, *Machine Learning* and *Mo* *Modeling* are **tools**: means towards an end.
- *Analytics*, *Business Intelligence*, *Econometrics* and *Artifi* **application areas**: domains that use the tools above (a produce results within its subject. Among them, Analytics generic term (i.e. non domain-specific).
- *Statistics* is a **branch** of Mathematics providing theoreti support to the above tools.
- *Data Science* is a catch-all term to describe using those a answers in those all areas (and also in others), specially v *Big Data*, which is nothing more than a label meaning d



www.kdnuggets.com/2013/10/7-steps-learning-data-mining-data-science.html

KDnuggets™
search KDnuggets
Search
advanced search
Hel

Data Mining Community Top Resource
for Analytics, Data Mining, and Data Science Software, Companies, Data, Jobs, Education, News, and more

Data Mining Software | News | Jobs | Academic | Companies | Courses | Datasets | Data Mining Course | Education | Meetings | Polls | Webcasts
Predictive Analytics World Business
DELIVERING ON THE PROMISE OF BIG DATA
REGISTER NOW!
San Francisco March 29 - April 2, 2015

Predictive Analytics Business, San Francisco, Mar 29 - Apr 2 - Register Now

Subscribe to **our newsletter on Analytics, Big Data, Data Mining** | Follow @kdnuggets **voted Best Big Data Twitter** | **Contact**

KDnuggets Home » News :: 2013 :: Oct :: News, Software :: 7 Steps for Learning Data Mining and Data Science (13:n25)

Latest News

→ Upcoming Webcasts on Analytics, Big Data, Data Science - Mar 10 and beyond
→ Webinar: Data Mining: Failure to Launch [Mar 11]
→ Interview: Slava Akmaev, Berg on Healthcare Transparency & Effectiveness using Big Data
→ Top KDnuggets tweets, Mar 2-8: 6 categories in the Hadoop Ecosystem; How PayPal uses Deep Learning to fight fraud
→ ICS (Prague): AVAST Fellowship in machine learning and data science

7 Steps for Learning Data Mining and Data Science

◀ Previous post Next post ▶

How to learn data mining and data science? I outline seven steps and point you to resources for becoming a data scientist.

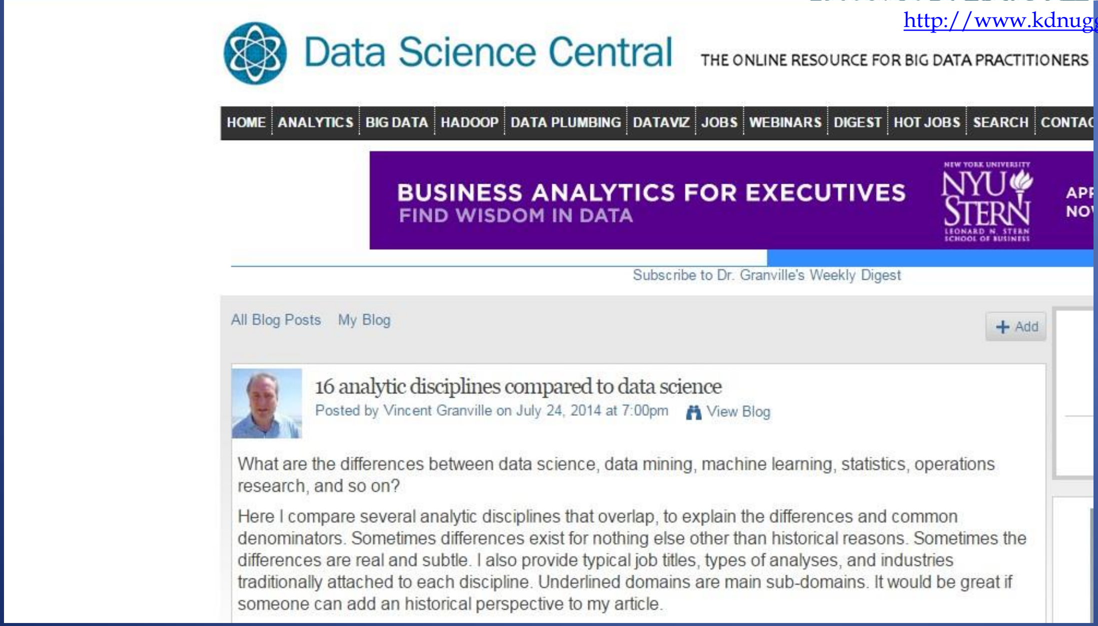
By Gregory Piatetsky, Oct 10, 2013. comments

I am frequently asked - how to learn Data Mining and Data Science?

Here is my summary. Let me know what I missed and add your comments

http://www.kdnuggets.com/2013/10/7-steps-learning-data-mining-data-science.html

www.datasciencecentral.com/profiles/blogs/17-analytic-disciplines-compared



Data Science Central
THE ONLINE RESOURCE FOR BIG DATA PRACTITIONERS
HOME ANALYTICS BIG DATA HADOOP DATA PLUMBING DATAVIZ JOBS WEBINARS DIGEST HOT JOBS SEARCH CONTAC
BUSINESS ANALYTICS FOR EXECUTIVES
FIND WISDOM IN DATA
NEW YORK UNIVERSITY NYU STERN LEONARD N. STERN SCHOOL OF BUSINESS
APP NOV

Subscribe to Dr. Granville's Weekly Digest

All Blog Posts My Blog
+ Add

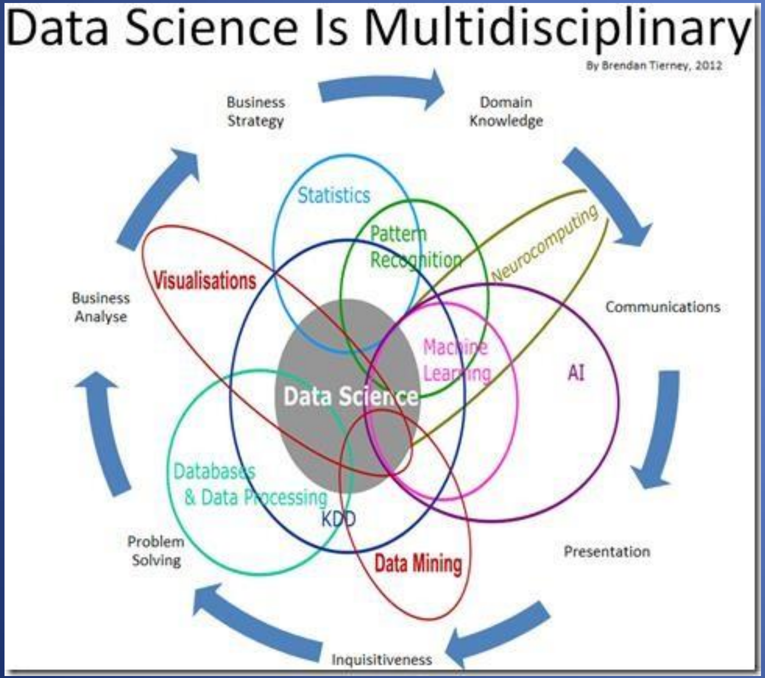
16 analytic disciplines compared to data science
Posted by Vincent Granville on July 24, 2014 at 7:00pm View Blog

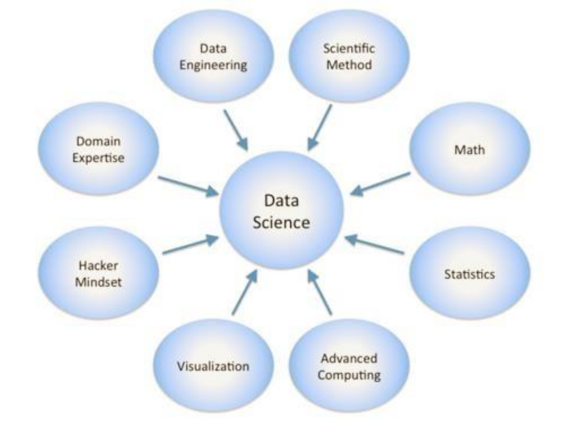
What are the differences between data science, data mining, machine learning, statistics, operations research, and so on?

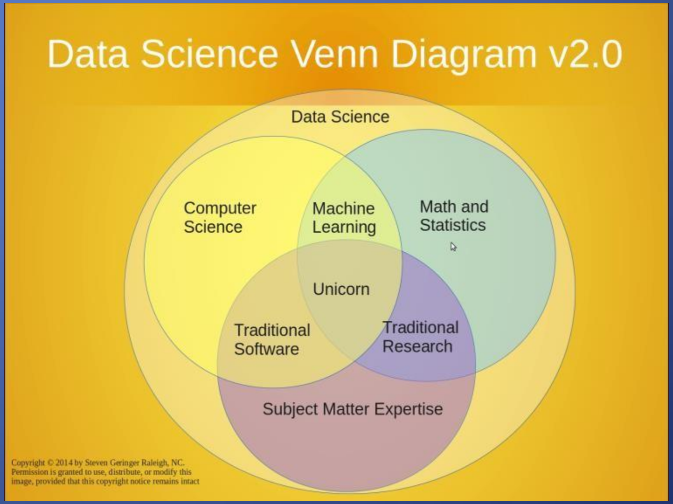
Here I compare several analytic disciplines that overlap, to explain the differences and common denominators. Sometimes differences exist for nothing else other than historical reasons. Sometimes the differences are real and subtle. I also provide typical job titles, types of analyses, and industries traditionally attached to each discipline. Underlined domains are main sub-domains. It would be great if someone can add an historical perspective to my article.

http://www.datasciencecentral.com/profiles/blogs/17-analytic-disciplines-compared

Definition?


Data Science Is Multidisciplinary
By Brendan Tierney, 2012
Business Strategy
Domain Knowledge
Communications
Presentation
Inquisitiveness
Problem Solving
Business Analyse
Statistics
Pattern Recognition
Neurocomputing
Visualisations
Machine Learning
AI
Data Science
Databases & Data Processing
KDD
Data Mining


Data Engineering
Scientific Method
Domain Expertise
Math
Data Science
Hacker Mindset
Statistics
Visualization
Advanced Computing


Data Science Venn Diagram v2.0
Data Science
Computer Science
Machine Learning
Math and Statistics
Unicorn
Traditional Software
Traditional Research
Subject Matter Expertise
Copyright © 2014 by Steven Geringer Raleigh, NC.
Permission is granted to use, distribute, or modify this image, provided that this copyright notice remains intact

Related Qualification?

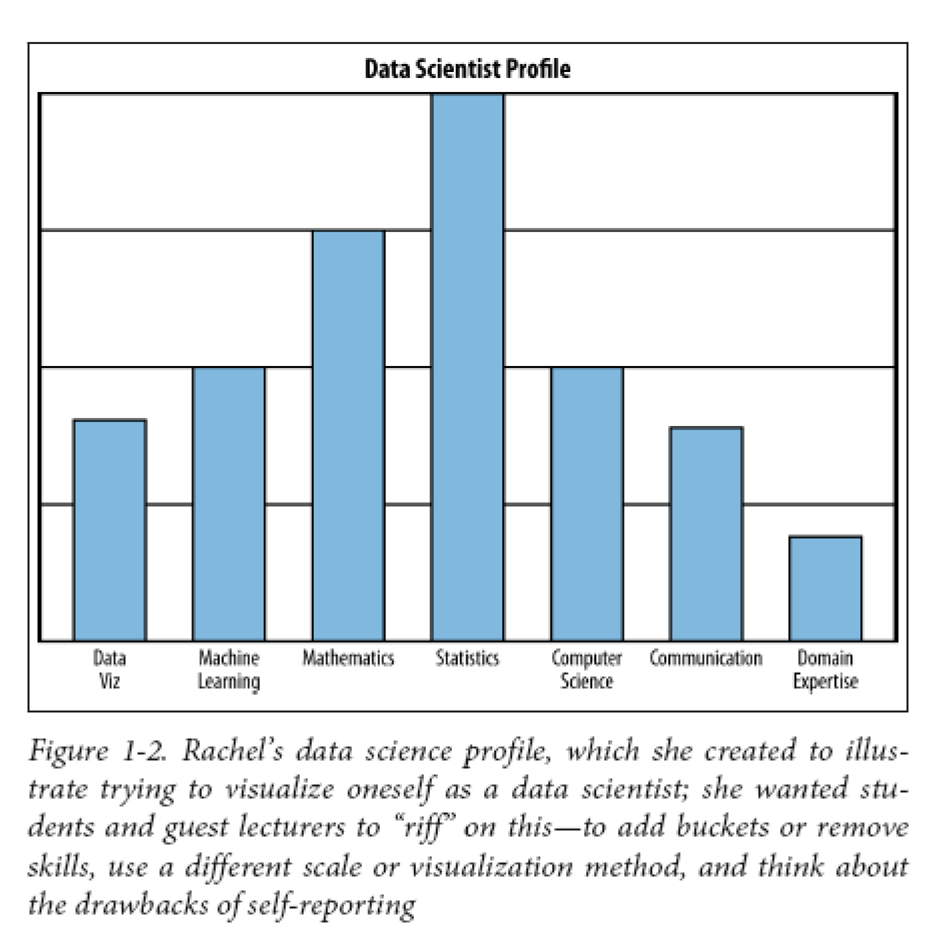
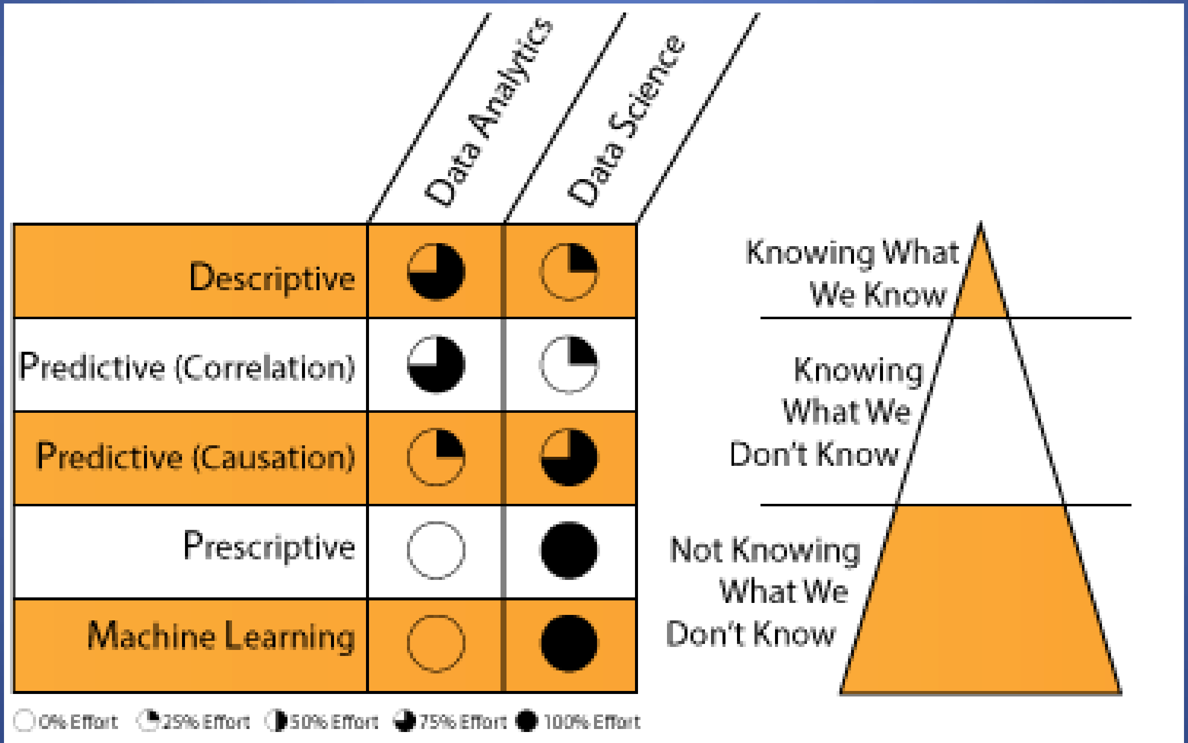


Figure 1-2. Rachel's data science profile, which she created to illustrate trying to visualize oneself as a data scientist; she wanted students and guest lecturers to "riff" on this—to add buckets or remove skills, use a different scale or visualization method, and think about the drawbacks of self-reporting

Data Science vs. Data Analytics



	Data Analytics	Data Science
Descriptive	◕	◔
Predictive (Correlation)	◕	◔
Predictive (Causation)	◔	◕
Prescriptive	○	●
Machine Learning	○	●

○ 0% Effort ◔ 25% Effort ◑ 50% Effort ◕ 75% Effort ● 100% Effort

Knowing What We Know

Knowing What We Don't Know

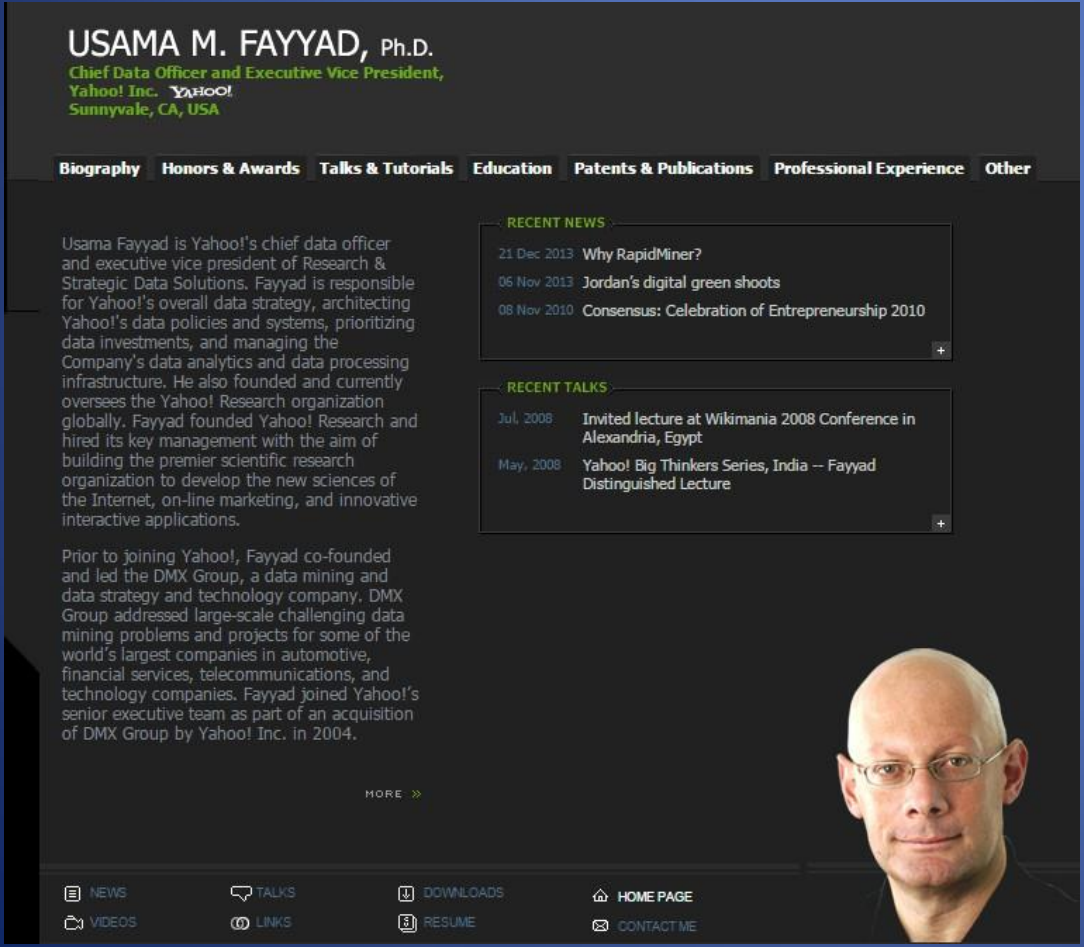
Not Knowing What We Don't Know

Relationship between them?

What do you think?

Real-World Cases

Real-World Cases



USAMA M. FAYYAD, Ph.D.

Chief Data Officer and Executive Vice President,
Yahoo! Inc. YAHOO!
Sunnyvale, CA, USA

Usama Fayyad is Yahoo!'s chief data officer and executive vice president of Research & Strategic Data Solutions. Fayyad is responsible for Yahoo!'s overall data strategy, architecting Yahoo!'s data policies and systems, prioritizing data investments, and managing the Company's data analytics and data processing infrastructure. He also founded and currently oversees the Yahoo! Research organization globally. Fayyad founded Yahoo! Research and hired its key management with the aim of building the premier scientific research organization to develop the new sciences of the Internet, on-line marketing, and innovative interactive applications.

Prior to joining Yahoo!, Fayyad co-founded and led the DMX Group, a data mining and data strategy and technology company. DMX Group addressed large-scale challenging data mining problems and projects for some of the world's largest companies in automotive, financial services, telecommunications, and technology companies. Fayyad joined Yahoo!'s senior executive team as part of an acquisition of DMX Group by Yahoo! Inc. in 2004.

RECENT NEWS

21 Dec 2013 Why RapidMiner?
06 Nov 2013 Jordan's digital green shoots
08 Nov 2010 Consensus: Celebration of Entrepreneurship 2010

RECENT TALKS

Jul. 2008 Invited lecture at Wikimania 2008 Conference in Alexandria, Egypt
May, 2008 Yahoo! Big Thinkers Series, India -- Fayyad Distinguished Lecture

2005….Yahoo!'s users, through their use of our network of products, generate over **10 terabytes** of data **per day**. This is the equivalent of the entire text contents of the library of Congress. This is data that describes product usage, and does not include content, email, or images, etc.

From Yahoo! To DigiMine

AMERICAN EXPRESS
GE
DAIMLERCHRYSLER
Ford
Microsoft
at&t
T-Mobile
THE WALL STREET JOURNAL.
BARNES & NOBLE BOOKSELLERS
CBS NEWS

The Awesome Ways Big Data Is Used Today To Change Our World

1. Understanding and Targeting Customers
2. Understanding and Optimizing Business Processes
3. Personal Quantification and Performance Optimization
4. Improving Healthcare and Public Health
5. Improving Sports Performance
6. Improving Science and Research
7. Optimizing Machine and Device Performance
8. Improving Security and Law Enforcement.
9. Improving and Optimizing Cities and Countries
10. Financial Trading

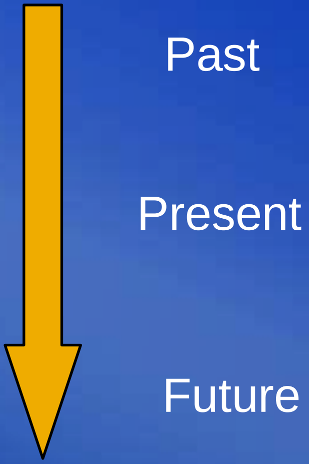
Q & A

What is Business Intelligence?

- Business Intelligence is the processes, technologies, and tools that help us change data into information, information into knowledge and knowledge into plans that guide organization
- Technologies for gathering, storing, analyzing and providing access to data to help enterprise users make better business Decisions

Why BI?

- What happened?
- What is happening?
- Why did it happen?
- What will happen?
- What do I want to happen?



Past

Present

Future

The characteristics of a Business intelligence solution

- Single point of access to information
- Timely answers to Business questions
- Using BI in all Departments of an organization

Key Stages of BI

Data Sourcing
Data Analysis
Situation Awareness
Risk Analysis
Decision Support

BI applications and technologies can help companies analyze:

- share
- changes in customer behavior and spending patterns
- customers' preferences

company capabilities

market conditions

Significance of BI…

- Companies need to have accurate, up-to-date information on customer preferences ,
  So that company can quickly adapt to their changing demands

- BI applications can also help managers to be better informed about actions that a company's competitors are taking

- It help analysts and managers to determine which adjustments are mostly likely to respond to changing trends

- IT can help companies develop a more consistent,data-based decision,which can produce better results than making business decisions by "guesswork"

MODULES

- Dashboards
- Key Performance Indicators
- Graphical OLAP
- Forecasting
- Graphical Reporting

MODULE DESCRIPTION

- Dashboards

- BI dashboards can provide a customized snapshot of daily operations, and assist the user in identifying problems and the source of those problems, as well as providing valuable, up-to-date information about financial results, sales and other critical information – all in one place

- **Key Performance Indicators**

- BI provides simplified KPI management and tracking with powerful features, formulae and expressions, and flexible frequency, and threshold levels. This module enables clear, concise definition and tracking of performance indicators for a period, and measures performance as compared to a previous period. Intuitive, color highlighters ensure that users can see these indicators in a clear manner and accurately present information to management and team members. Users can further analyse performance with easy-to-use features like drill down, drill through, slice and dice and graphical data mining

- **Graphical OLAP**

- Graphical Business Intelligence (BI) OLAP technology makes it easy for your users to find, filter and analyse data, going beyond numbers, and allowing users to visualize the information with eye-catching, stunning displays, and valuable indicators and gauges, charts, and a variety of graph types from which to choose

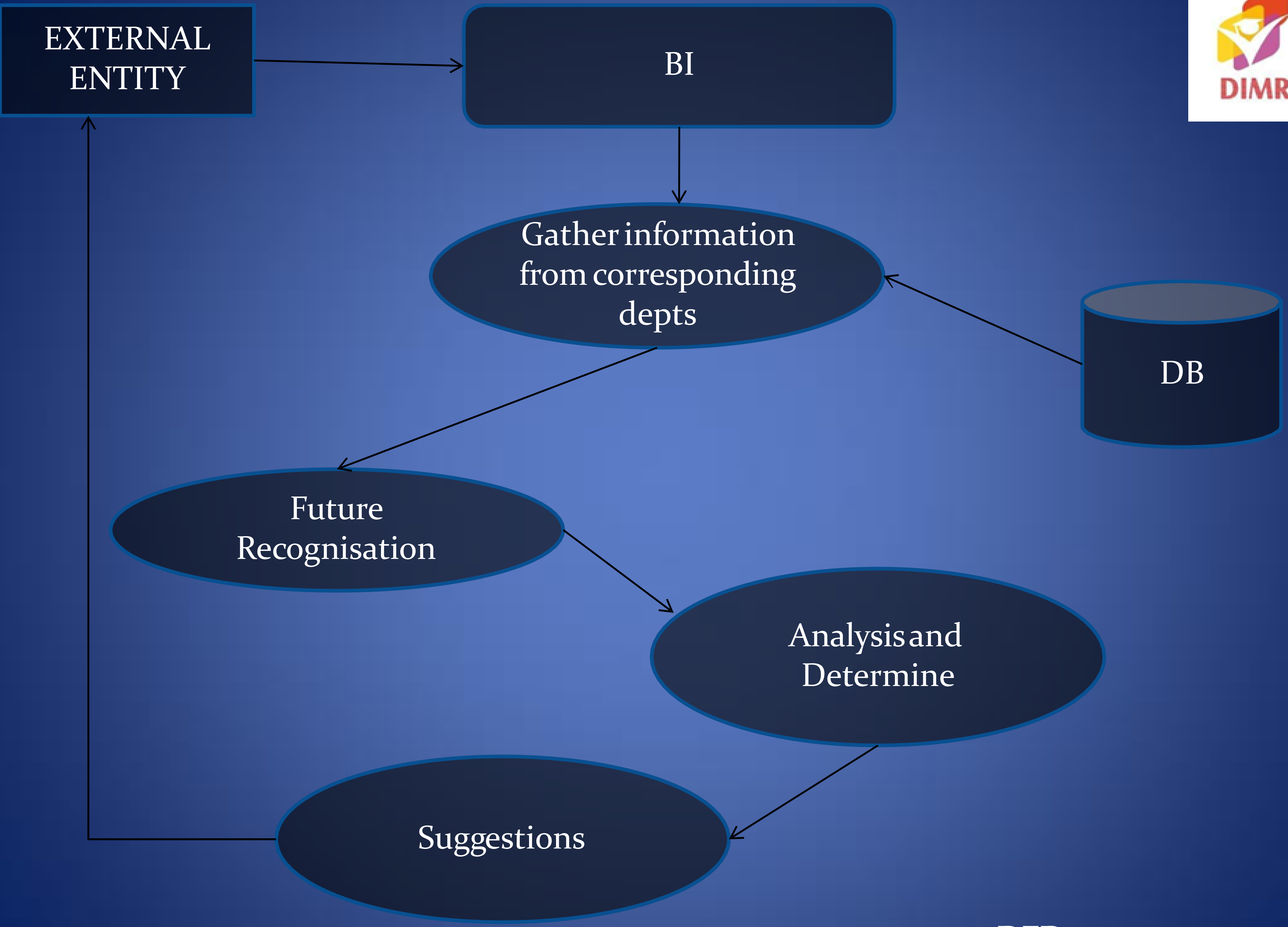
- **Forecasting and Predictive Analysis**

- Our predictive analysis uses historical product, sales, pricing, financial, budget and other data, and forecasts the measures with numerous time series options, e.g., year, quarter, month, week, day, hour or even second to improve your planning process

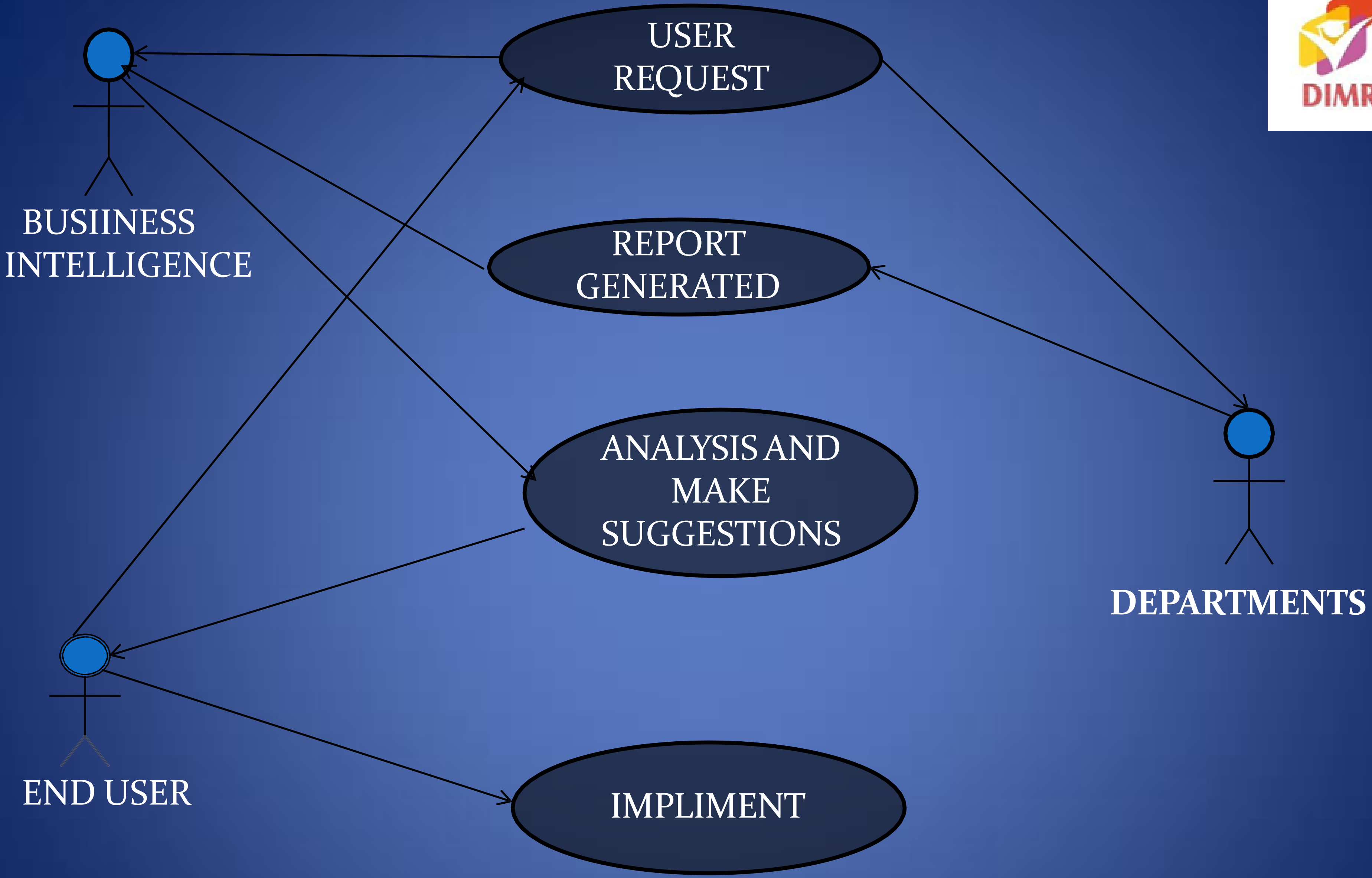
- **Reports**

- BI Reports delivers web-based BI reports to anyone (or everyone) in the organization within minutes! The BI suite is simple to use, practical to implement and affordable for every organization. With our BI reporting and performance reporting module, you just point-and-click and drag-and-drop and you can instantly create a report to summarize your performance metrics, or operational data

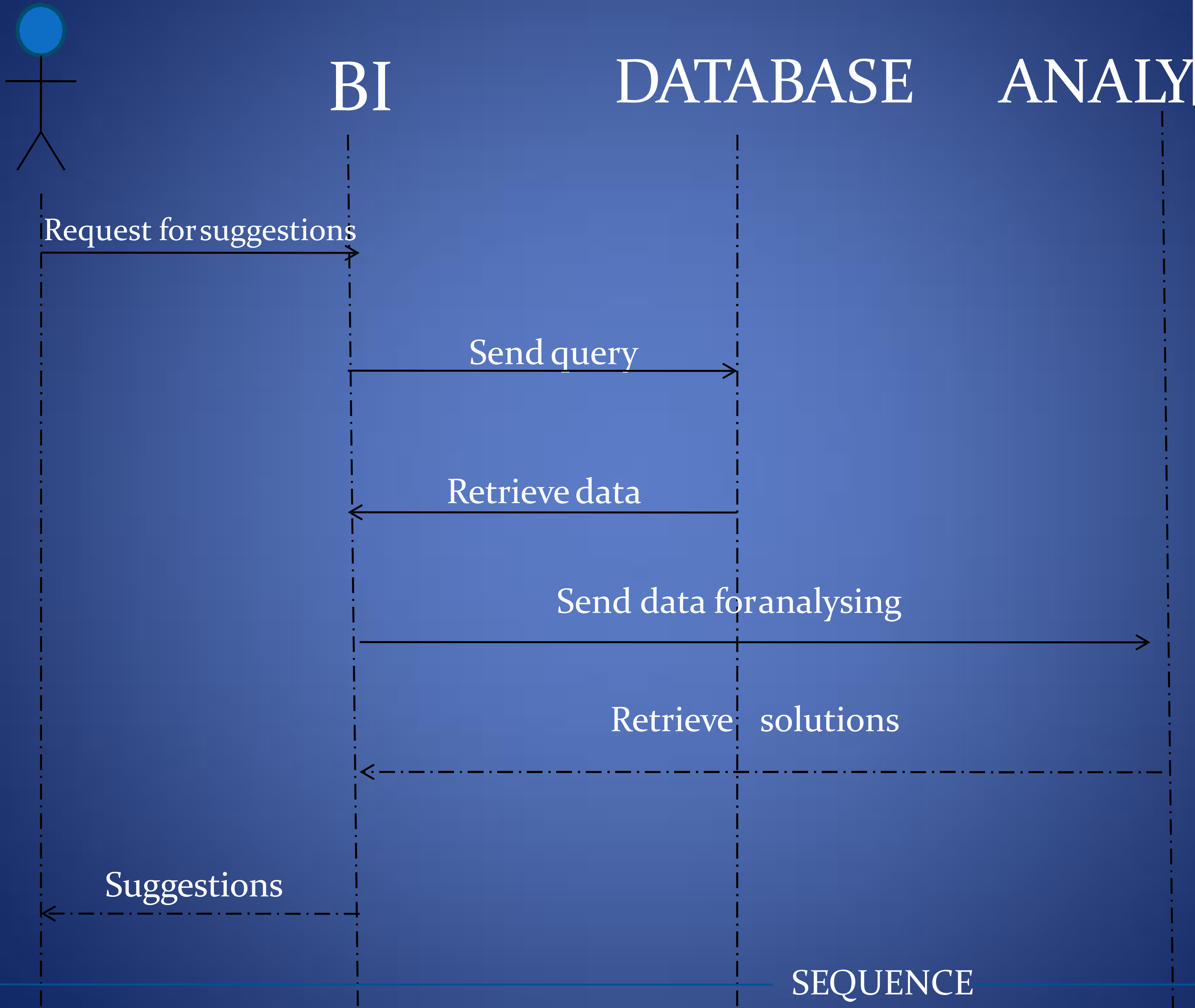
CLASS DIAGRAMS


EXTERNAL ENTITY
BI
Gather information from corresponding depts
DB
Future Recognisation
Analysis and Determine
Suggestions

DFD


USER REQUEST
BUSIINESS INTELLIGENCE
REPORT GENERATED
ANALYSIS AND MAKE SUGGESTIONS
DEPARTMENTS
END USER
IMPLIMENT

USECASE



BI

DATABASE

ANALY

Request for suggestions

Send query

Retrieve data

Send data for analysing

Retrieve solutions

Suggestions

SEQUENCE

Thank You!

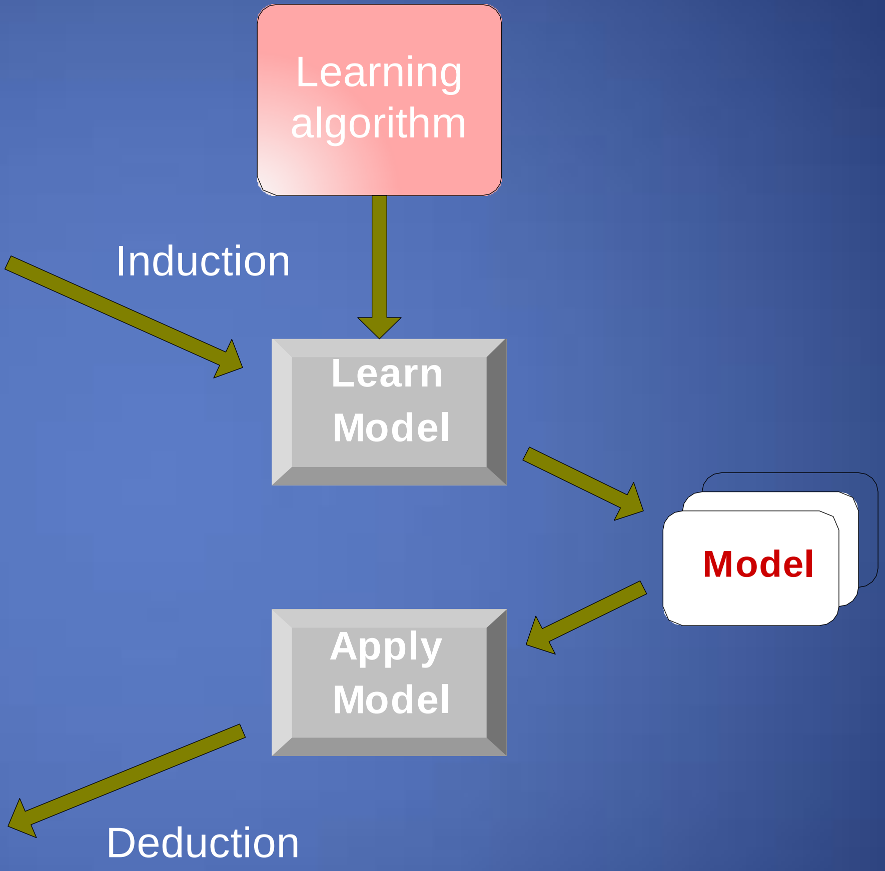
Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Learning algorithm

Induction

Learn Model

Model

Apply Model

Deduction

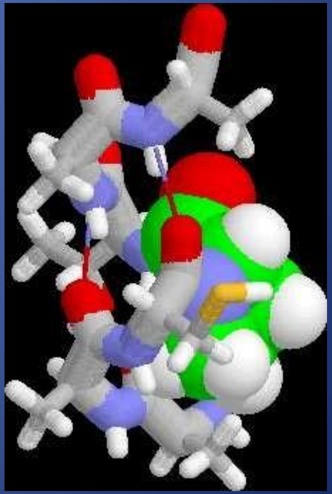
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Examples of Classification Task

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent

- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil



- Categorizing news stories as finance, weather, entertainment, sports, etc

Classification Techniques

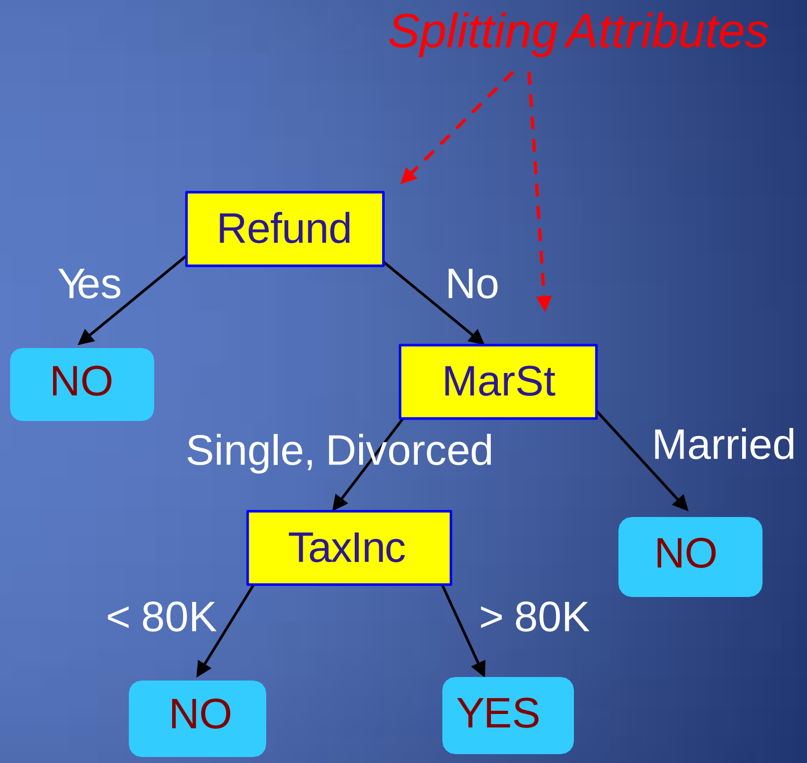
- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

Example of a Decision Tree

categorical categorical continuous class

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

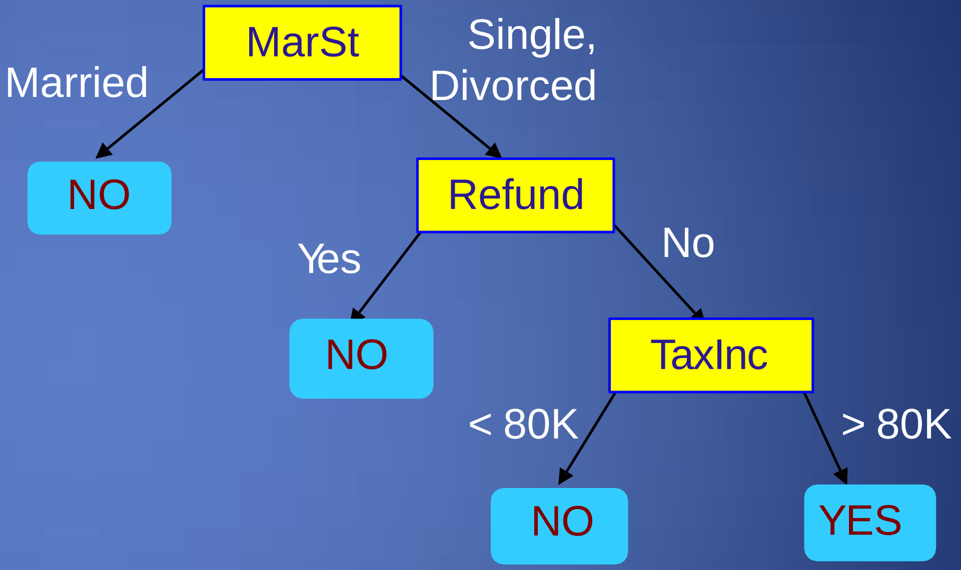
Training Data



Model: Decision Tree

Another Example of Decision Tree

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

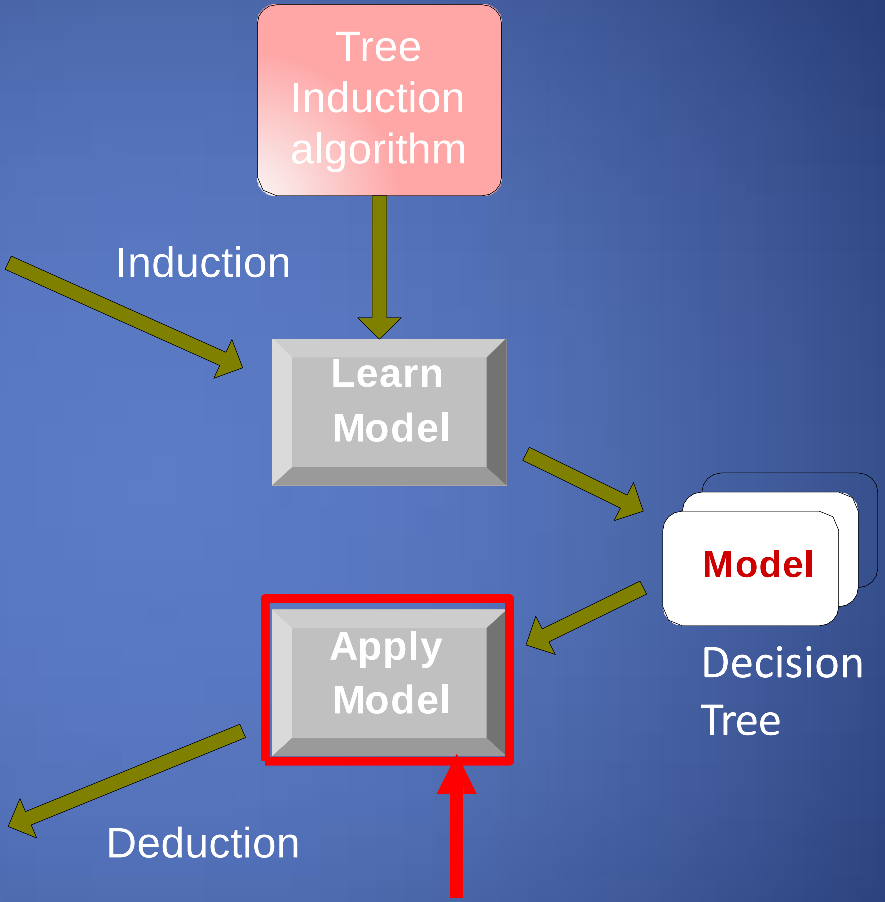


There could be more than one tree that fits the same data!

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tree Induction algorithm

Induction

Learn Model

Model

Decision Tree

Apply Model

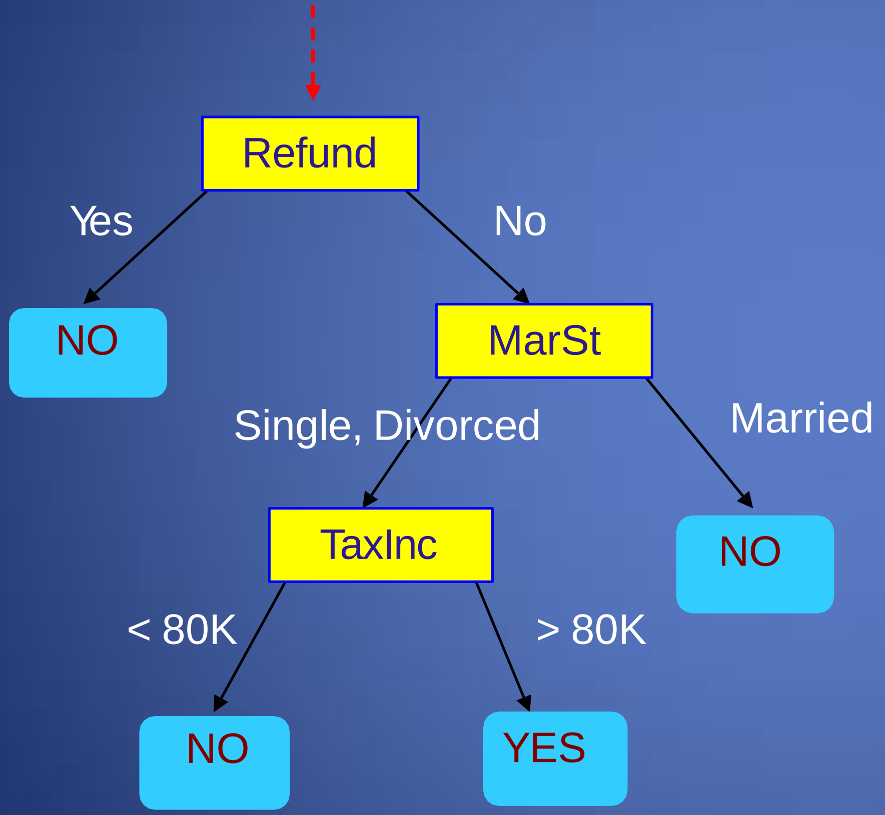
Deduction

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Apply Model to Test Data

Start from the root of tree.



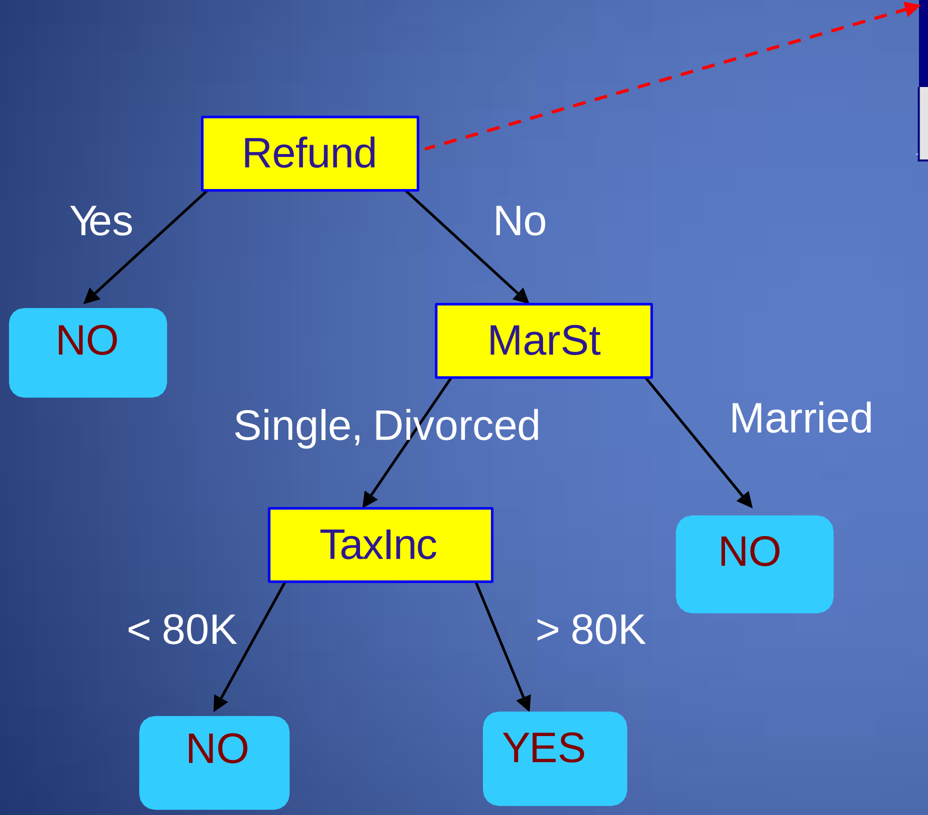
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

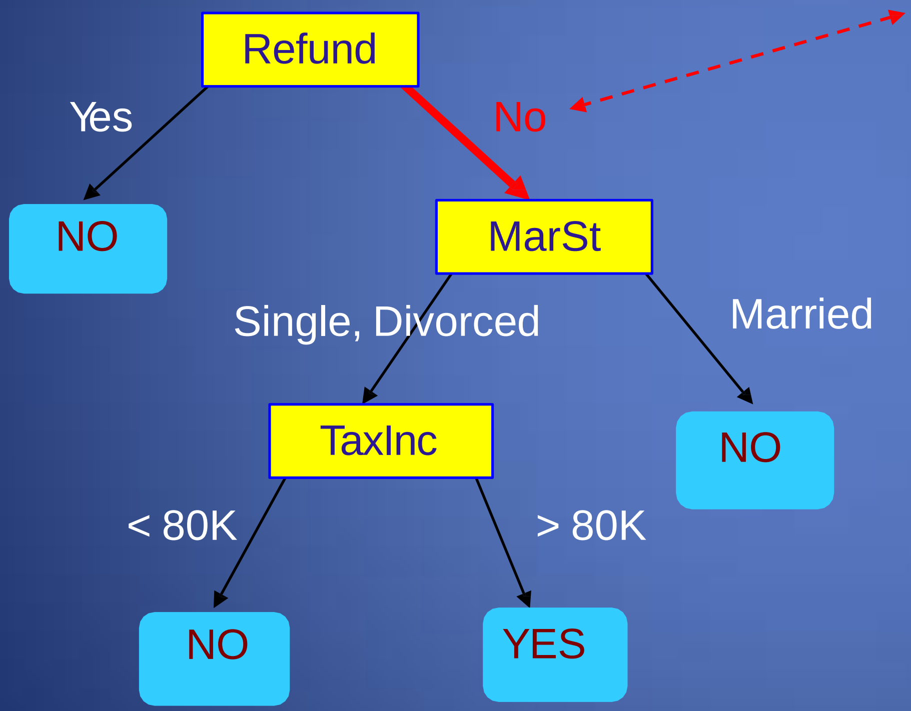
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

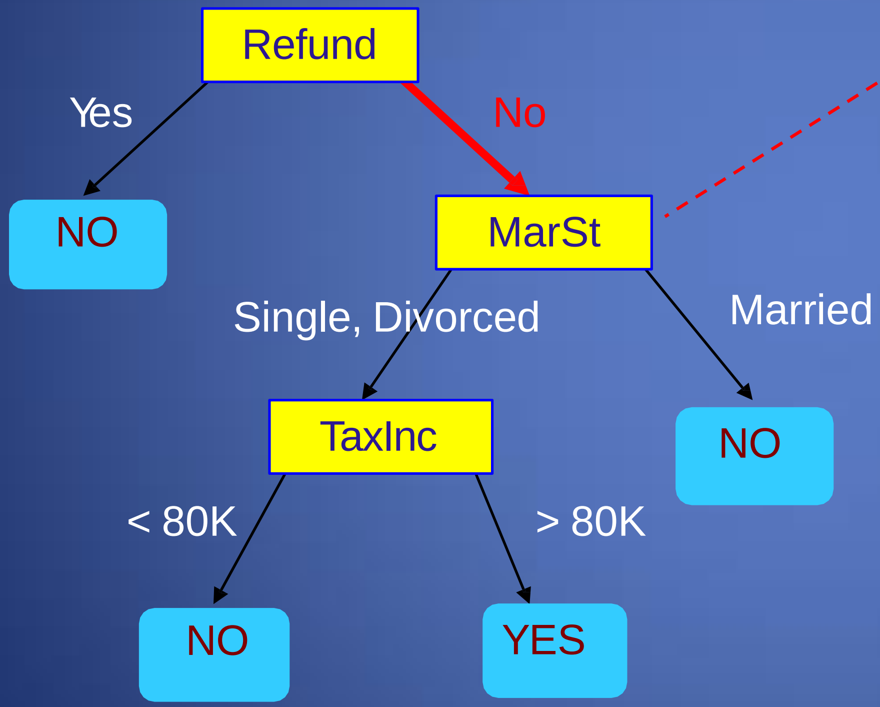
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

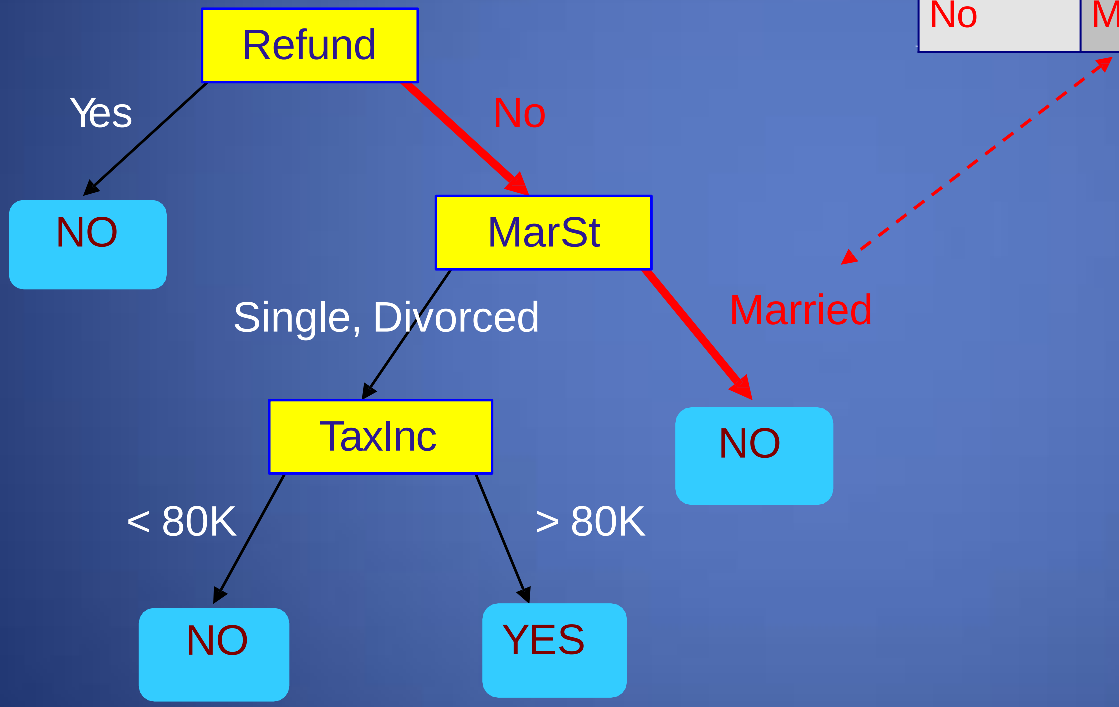
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

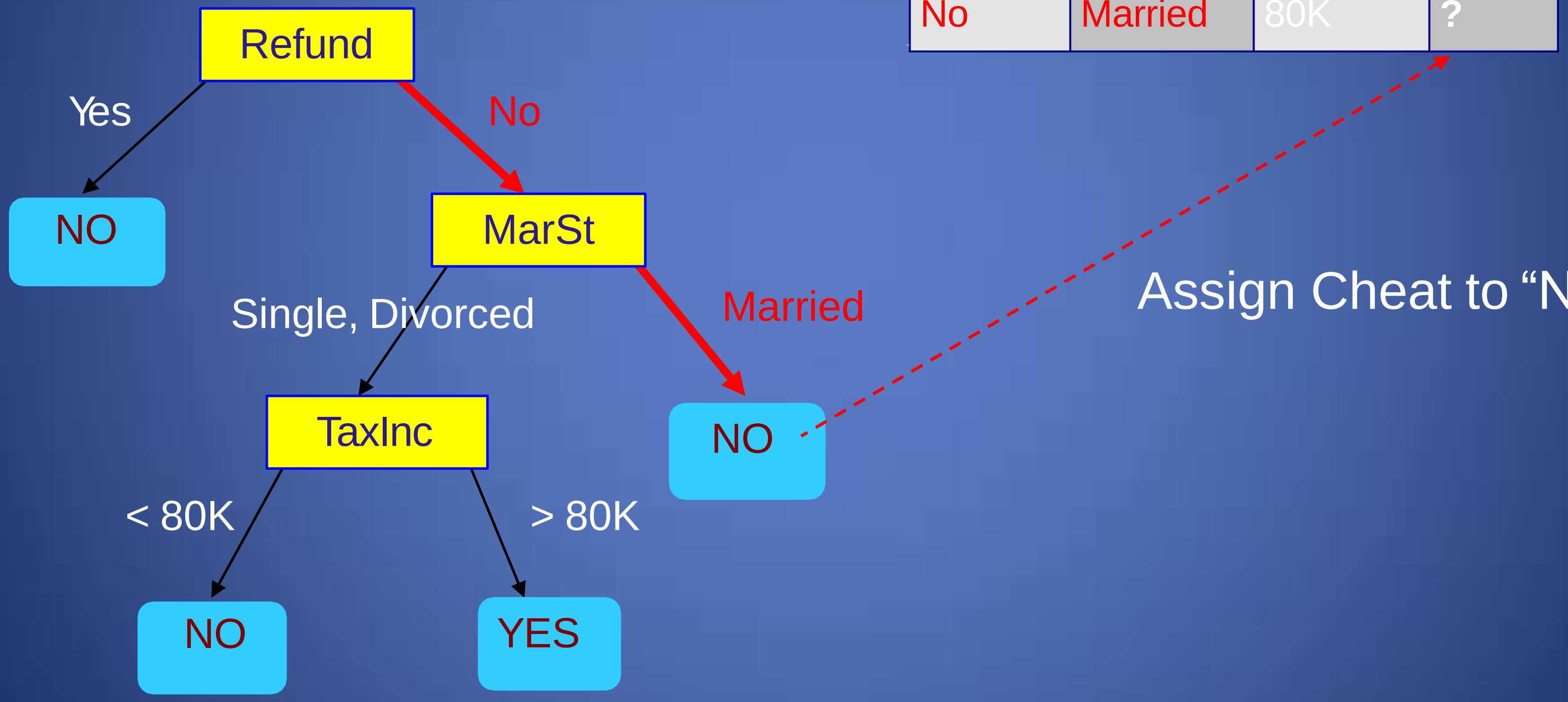
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

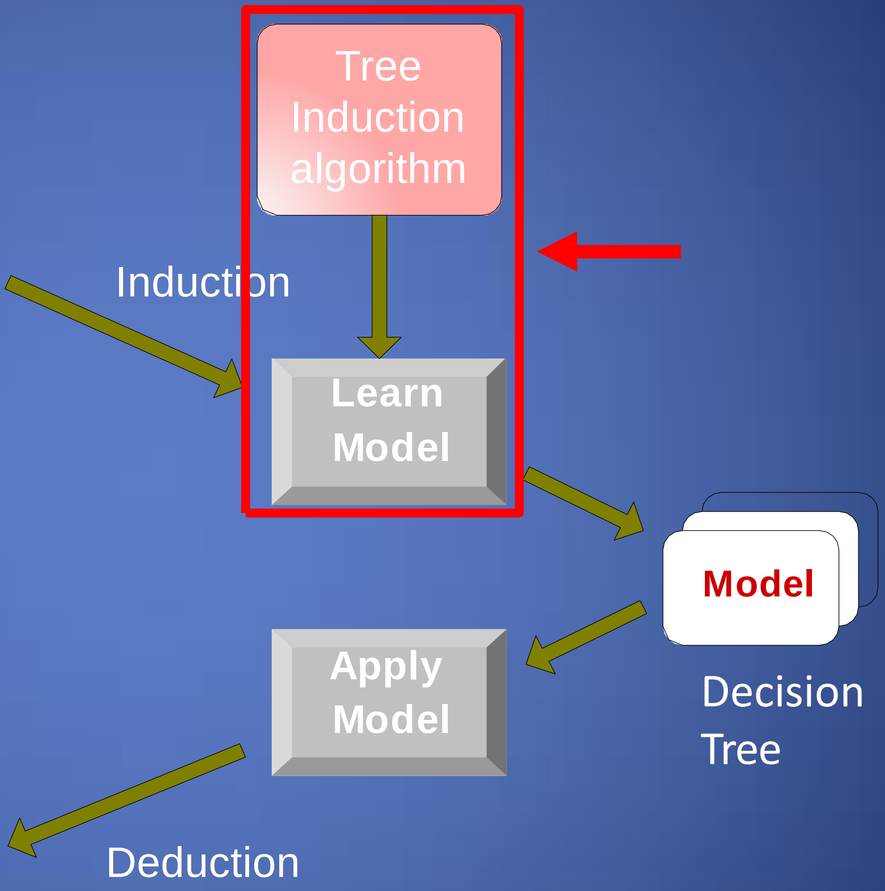


Assign Cheat to “No”

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tree Induction algorithm

Induction

Learn Model

Model

Decision Tree

Apply Model

Deduction

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

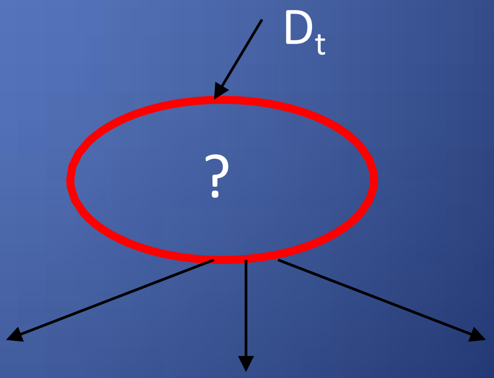
Decision Tree Induction

- Many Algorithms:
  1. Hunt's Algorithm (one of the earliest)
  2. CART (Classification And Regression Tree)
  3. ID3 (Iterative Dichotomiser 3)
  4. C4.5 (Successor of ID3)
  5. SLIQ (It does not require loading the entire dataset into the main memory)
  6. SPRINT (similar approach as SLIQ, induces decision trees relatively quickly)
  7. CHAID (CHi-squared Automatic Interaction Detector). Performs multi-level splits when computing classification trees.
  8. MARS: extends decision trees to handle numerical data better.
  9. Conditional Inference Trees. Statistics-based approach that uses non-parametric tests as splitting criteria, corrected for multiple testing to avoid overfitting.

General Structure of Hunt’s Algorithm

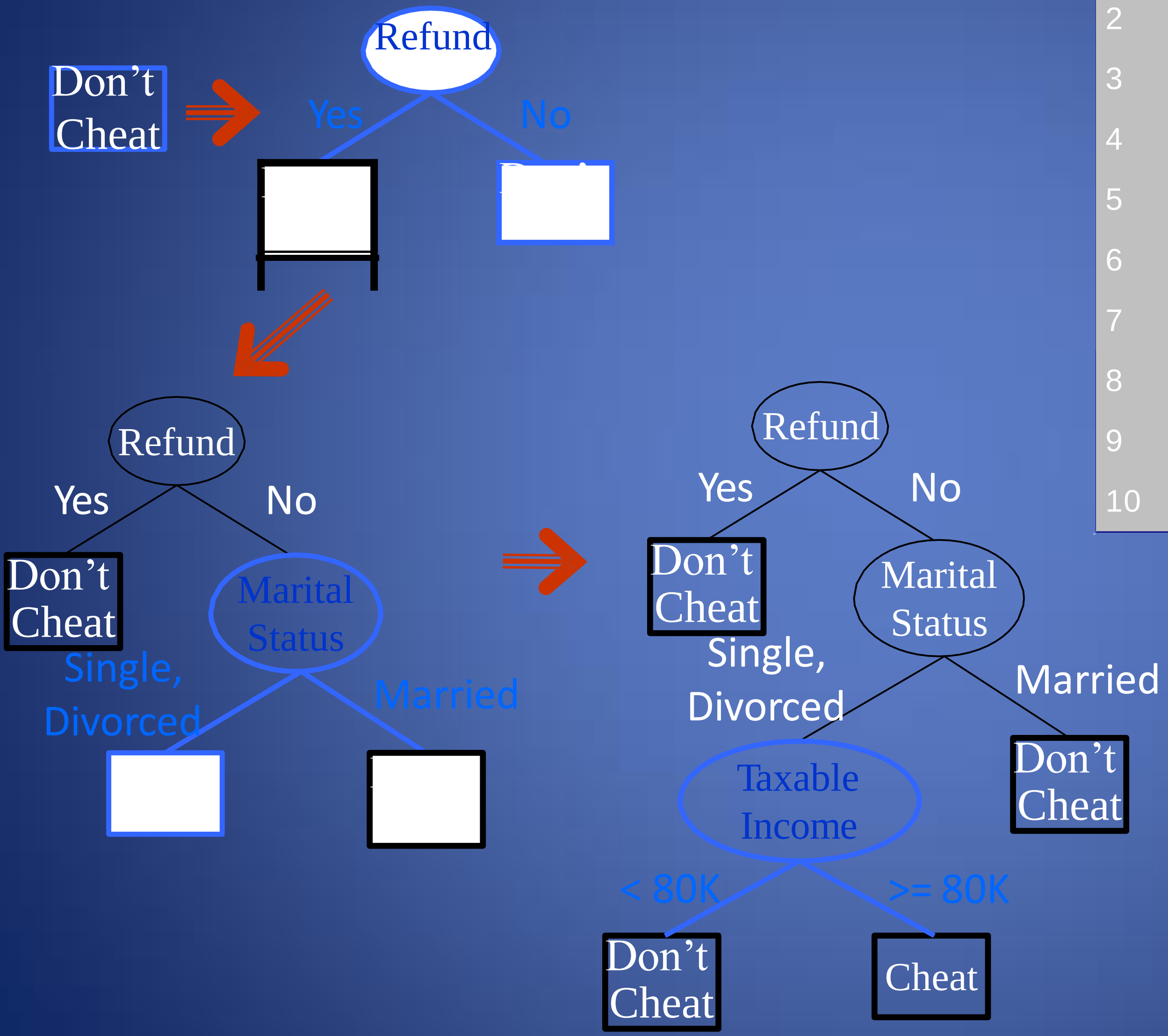
- Let D_t be the set of training records that reach a node t
- General Procedure:
  - If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  - If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
  - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**



Hunt’s Algorithm

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**



Evaluation of a Classifier

- How predictive is the model we learned?
  - Which performance measure to use?
- Natural performance measure for classification problems: **error rate** on a test set
  - ***Success:*** instance's class is predicted correctly
  - ***Error:*** instance's class is predicted incorrectly
  - ***Error rate:*** proportion of errors made over the whole set of instances
  - ***Accuracy:*** proportion of correctly classified instances over the whole set of instances

accuracy = 1 – error rate

Confusion Matrix

- A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

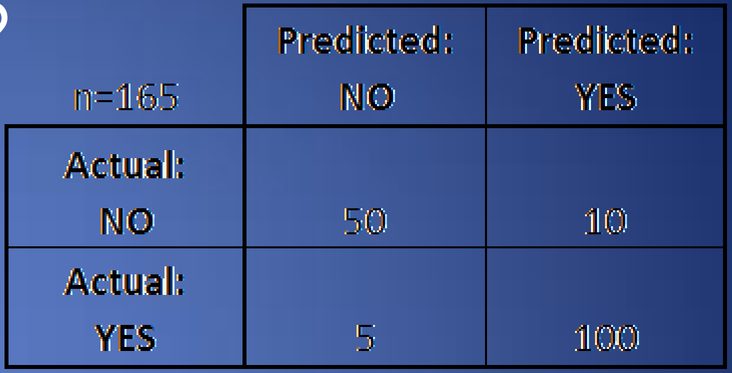
	PREDICTED CLASS		
ACTUAL CLASS		Class = Yes	Class = No
	Class = Yes	a	b
	Class = No	c	d

a: TP (true positive)
b: FN (false negative)

c: FP (false positive)
d: TN (true negative)

Confusion Matrix - Example

- What can we learn from this matrix?



n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

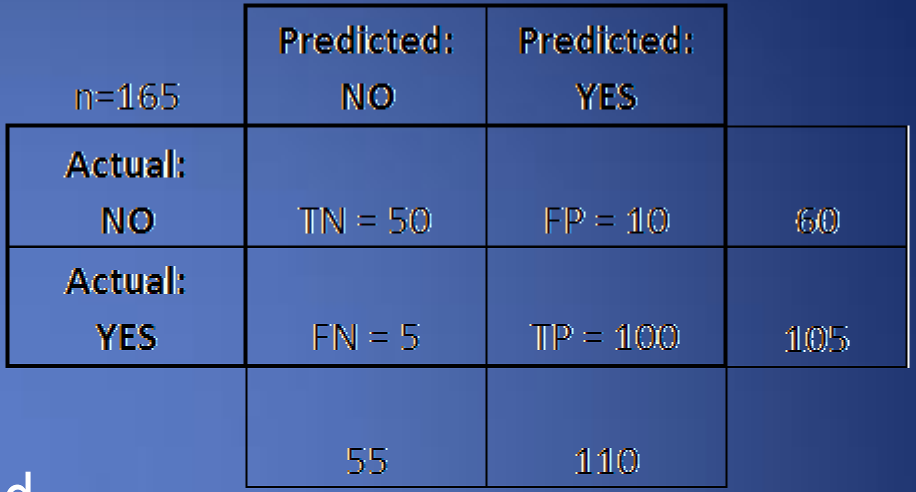
  - There are two possible predicted classes: "yes" and "no". If we were predicting the presence of a disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.
  - The classifier made a total of 165 predictions (e.g., 165 patients were being tested for the presence of that disease).
  - Out of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.
  - In reality, 105 patients in the sample have the disease, and 60 patients do not.

Confusion Matrix – Confusion?

- False positives are actually negative
- False negatives are actually positives

Confusion Matrix - Example

- Let's now define the most basic terms, which are whole numbers (not rates):



n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

  - true positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease.
  - true negatives (TN): We predicted no, and they don't have the disease.
  - false positives (FP): We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
  - false negatives (FN): We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

Confusion Matrix - Computations

- This is a list of rates that are often computed from a confusion matrix:

- Accuracy: Overall, how often is the classifier correct?

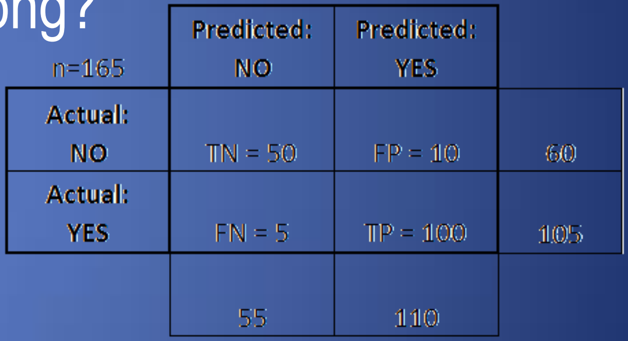
  (TP+TN)/total = (100+50)/165 = 0.91

- Misclassification Rate: Overall, how often is it wrong?

  (FP+FN)/total = (10+5)/165 = 0.09

  equivalent to 1 minus Accuracy

  also known as "Error Rate"



n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

- True Positive Rate: When it's actually yes, how often does it predict yes?

  TP/actual yes = 100/105 = 0.95

  also known as "Sensitivity" or "Recall"

- False Positive Rate: When it's actually no, how often does it predict yes?

  FP/actual no = 10/60 = 0.17

Confusion Matrix - Computations

- This is a list of rates that are often computed from a confusion matrix:

- Specificity: When it's actually no, how often does it predict no?

  TN/actual no = 50/60 = 0.83

  equivalent to 1 minus False Positive Rate
- Precision: When it predicts yes, how often is it correct?

  TP/predicted yes = 100/110 = 0.91
- Prevalence: How often does the yes condition actually occur in our sample?

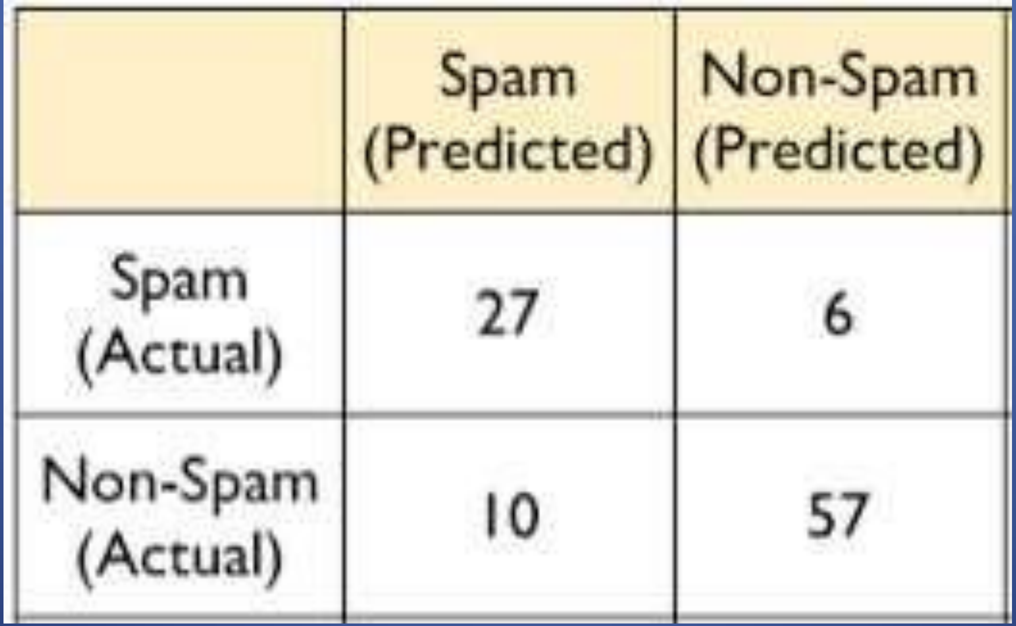
  actual yes/total = 105/165 = 0.64

Confusion Matrix – Example 2

- Imagine that you have a dataset that consists of 33 patterns that are 'Spam' (S) and 67 patterns that are 'Non-Spam' (NS).
- In the example 33 patterns that are 'Spam' (S), 27 were correctly predicted as 'Spams' while 6 were incorrectly predicted as 'Non-Spams'.
- On the other hand, out of the 67 patterns that are 'Non-Spams', 57 are correctly predicted as 'Non-Spams' while 10 were incorrectly classified as 'Spams'.

Confusion Matrix – Example 2

- Accuracy = (TP+TN)/total = (27+57)/100 = 84%
- Misclassification Rate = (FP+FN)/total = (6+10)/100 = 16%
- True Positive Rate = TP/actual yes = 27/33 = 0.81
- False Positive Rate =FP/actual no = 10/67 = 0.15



	Spam (Predicted)	Non-Spam (Predicted)
Spam (Actual)	27	6
Non-Spam (Actual)	10	57

Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

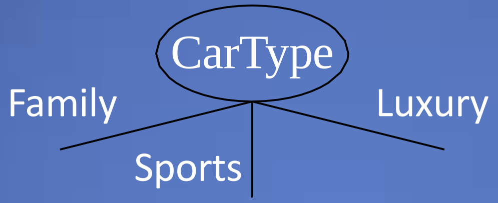
How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

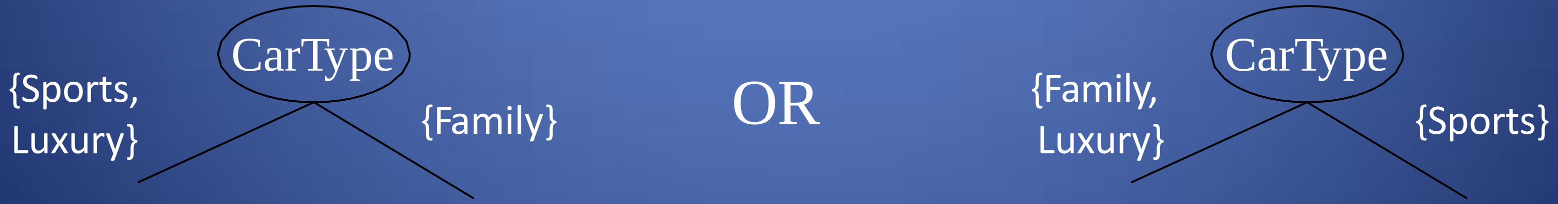
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

- Multi-way split: Use as many partitions as distinct values.

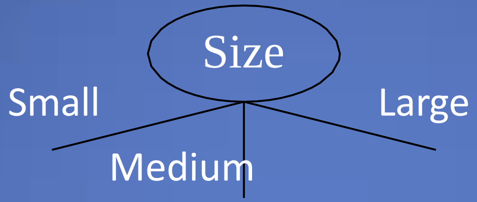


- Binary split: Divides values into two subsets. Need to find optimal partitioning.

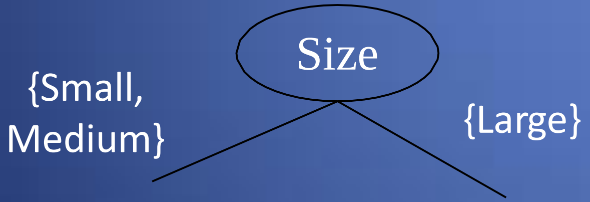


Splitting Based on Ordinal Attributes

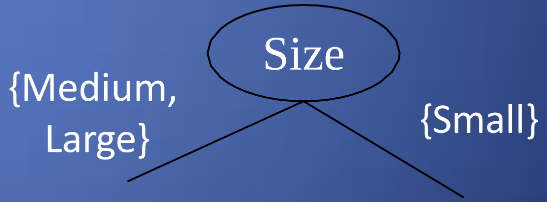
- Multi-way split: Use as many partitions as distinct values.



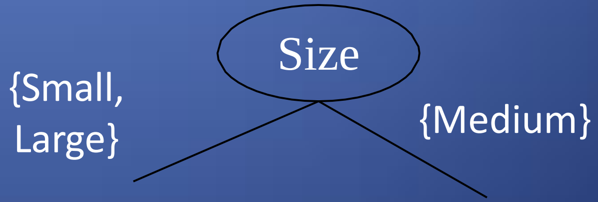
- Binary split: Divides values into two subsets. Need to find optimal partitioning.



OR



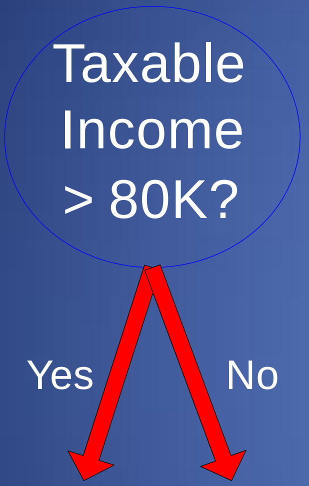
- What about this split?



Splitting Based on Continuous Attributes

- Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - consider all possible splits and finds the best cut
    - can be more compute intensive

Splitting Based on Continuous Attributes

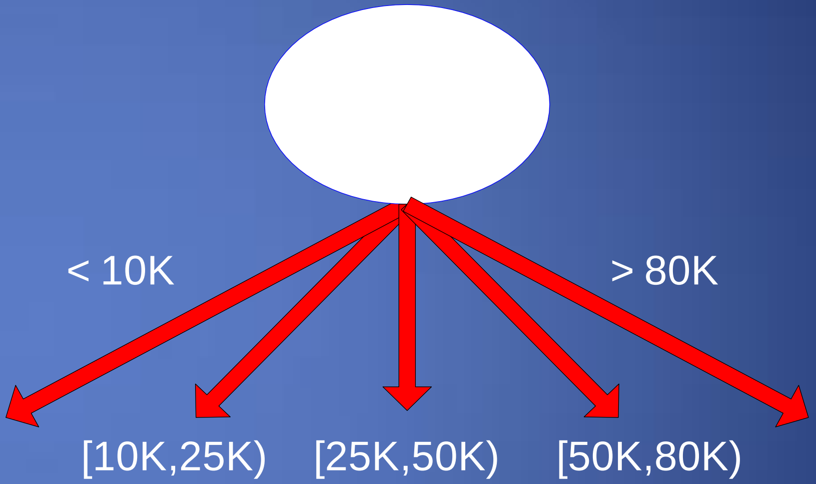


Taxable Income > 80K?

Yes

No

(i) Binary split



< 10K

> 80K

[10K,25K)

[25K,50K)

[50K,80K)

(ii) Multi-way split

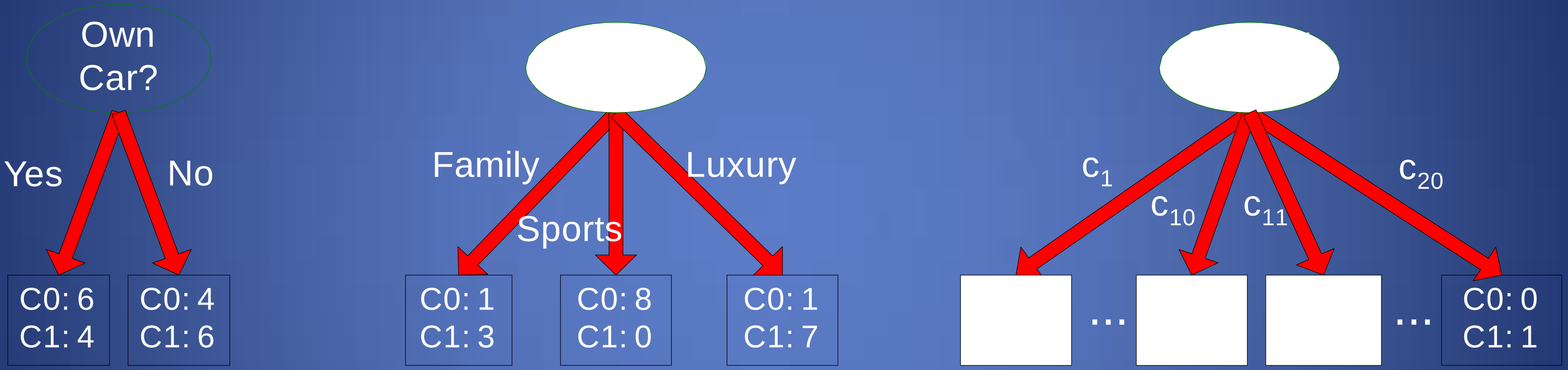
Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

How to determine the Best Split

Before Splitting: 10 records of class 0,
10 records of class 1



Own Car?

Yes

No

C0: 6
C1: 4

C0: 4
C1: 6

Family

Sports

Luxury

C0: 1
C1: 3

C0: 8
C1: 0

C0: 1
C1: 7

c_1

c_{10}

c_{11}

c_{20}

...

...

C0: 0
C1: 1

Which test condition is the best?

How to determine the Best Split

- Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:

C0: 5
C1: 5

Non-homogeneous,

High degree of impurity

C0: 9
C1: 1

Homogeneous,

Low degree of impurity

How to Measure Impurity?

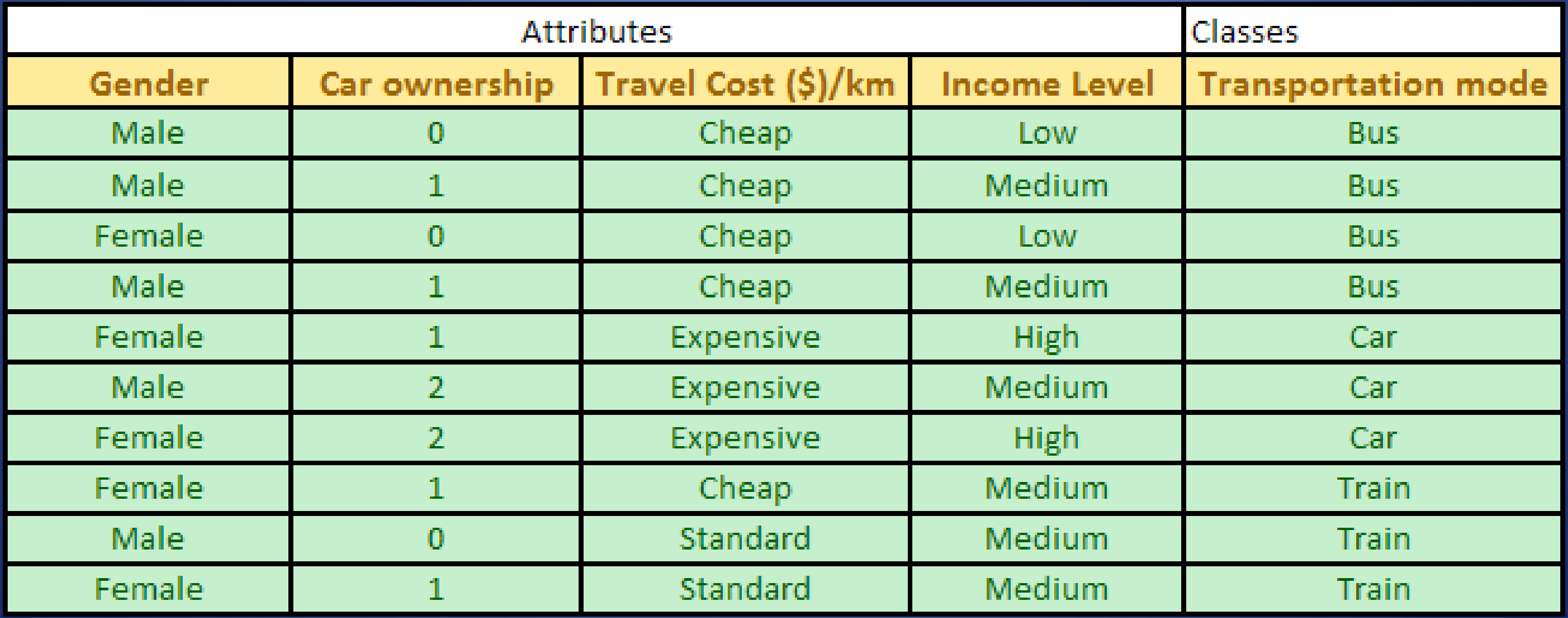
- Given a data table that contains attributes and class of the attributes, we can measure homogeneity (or heterogeneity) of the table based on the classes.
- We say a table is pure or homogenous if it contains only a single class.
- If a data table contains several classes, then we say that the table is impure or heterogeneous.

How to Measure Impurity?

- There are several indices to measure degree of impurity quantitatively.
- Most well known indices to measure degree of impurity are:
  - Entropy $$Entropy = \sum_j -p_j log_2 p_j$$
  - Gini Index $$Gini\ Index = 1 - \sum_j p_j^2$$
  - Misclassification error $$Classification\ Error = 1 - max\{p_j\}$$
- All above formulas contain values of probability of p_j a class j.

How to Measure Impurity? - Example

- In our example, the classes of Transportation mode below consist of three groups of Bus, Car, and Train. In this case, we have 4 buses, 3 cars, and 3 trains (in short we write as 4B, 3C, 3T). The total data is 10 rows.



Attributes				Classes
Gender	**Car ownership**	**Travel Cost ($)/km**	**Income Level**	**Transportation mode**
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car
Female	1	Cheap	Medium	Train
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train

How to Measure Impurity? - Example

- Based on the data, we can compute probability of each class. Since probability is equal to frequency relative, we have
  - Prob(Bus) = 4/10 = 0.4
  - Prob(Car) = 3/10 = 0.3
  - Prob(Train) = 3/10 = 0.3
- Observe that when to compute the probability, we only focus on the classes, not on the attributes. Having the probability of each class, now we are ready to compute the quantitative indices of impurity degrees.

How to Measure Impurity? - Entro

- One way to measure impurity degree is using entropy

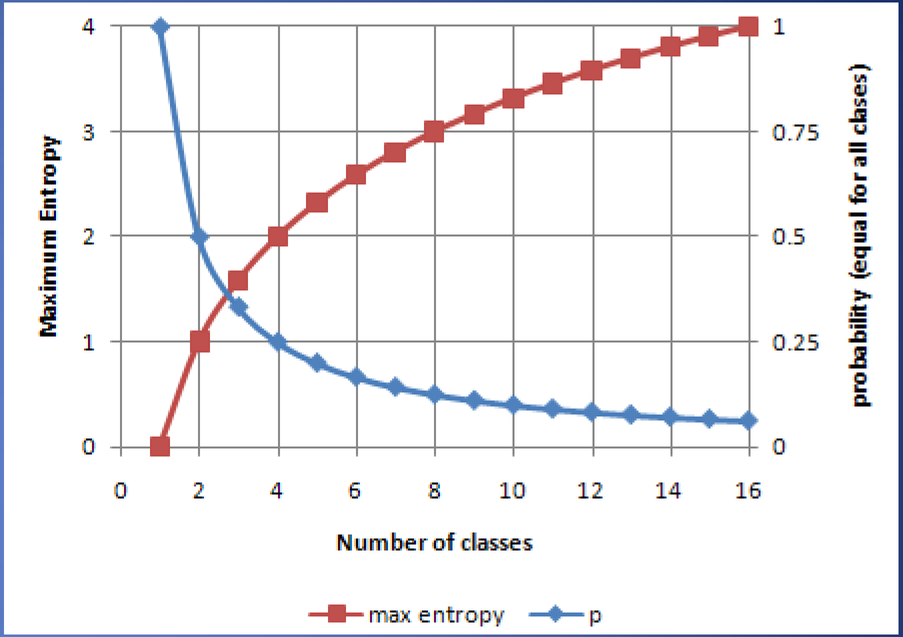
$$Entropy = \sum_{j} -p_j log_2 p_j$$

- Example: Given that Prob(Bus)=0.4, Prob(Car)=0.3, Prob(Train)=0.3, we can now compute entropy as:

- Entropy = - $0.4\log_2(0.4)$ - $0.3\log_2(0.3)$ - $0.3\log_2(0.3)$ = 1.571

How to Measure Impurity? - Entrop

- Entropy of a pure table (consist of single class) is zero because the probability is 1 and $\log_2(1)=0$.
- Entropy reaches maximum value when all classes in the table have equal probability.
- Figure plots the values of maximum entropy for different number of classes n, where probability is equal to p=1/n.
- In this case, maximum entropy is equal to $-n*p*\log_2 p$.
- Notice that the value of entropy is larger than 1 if the number of classes is more than 2.



How to Measure Impurity? - Gini

- Another way to measure impurity degree is using Gini index

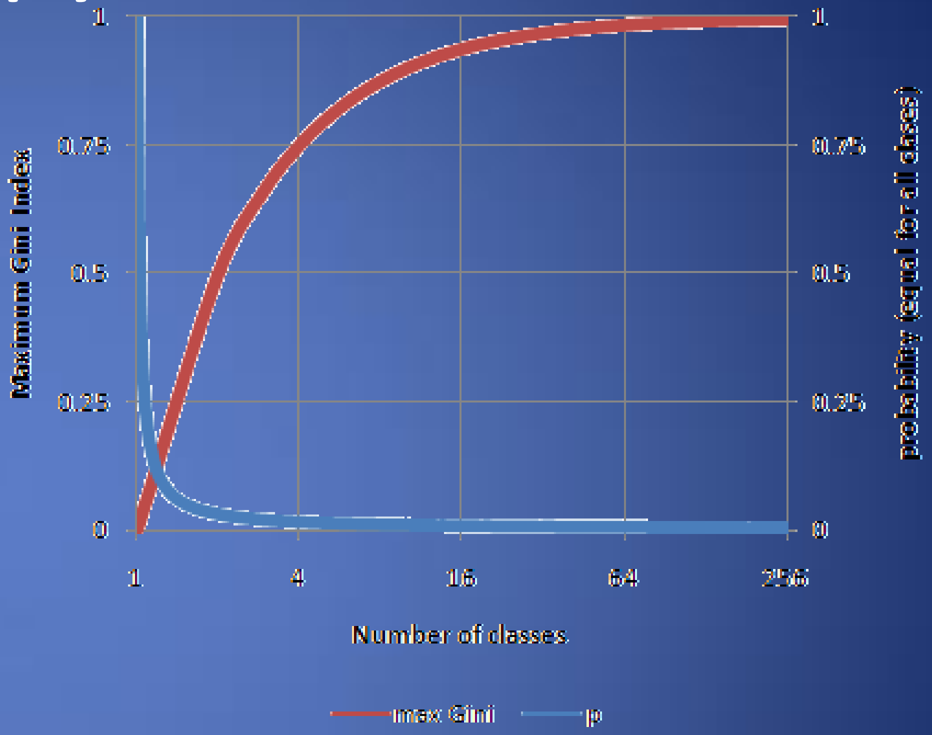
$$Gini\ Index = 1 - \sum_{j} p_j^2$$

- Example: Given that Prob(Bus)=0.4, Prob(Car)=0.3, Prob(Train)=0.3, we can now compute Gini index as:

- Gini Index = 1 - (0.4^2 + 0.3^2 + 0.3^2) = 0.660

How to Measure Impurity? - Entropy

- Gini index of a pure table (consist of a single class is zero because the probability is 1 and 1-(1)^2=0.
- Similar to Entroy, Gini index also reaches maximum value when all classes in the table have equal probability.
- Figure plots the values of maximum Gini index for different number of classes n, where probability is equal to p=1/n.
- Notice that the value of Gini index is always between 0 and 1 regardless the number of classes.



How to Measure Impurity? – Missclassification Error

- Still anot $\text{Classification Error} = 1 - max\{p_j\}$ urity degree

- Example: Given that Prob(Bus)=0.4, Prob(Car)=0.3, Prob(Train)=0.3, we can now compute index as:

- Index = 1 - Max{0.4,0.3,0.3} = 1-0.4 = 0.60

How to Measure Impurity? – Missclassification Error

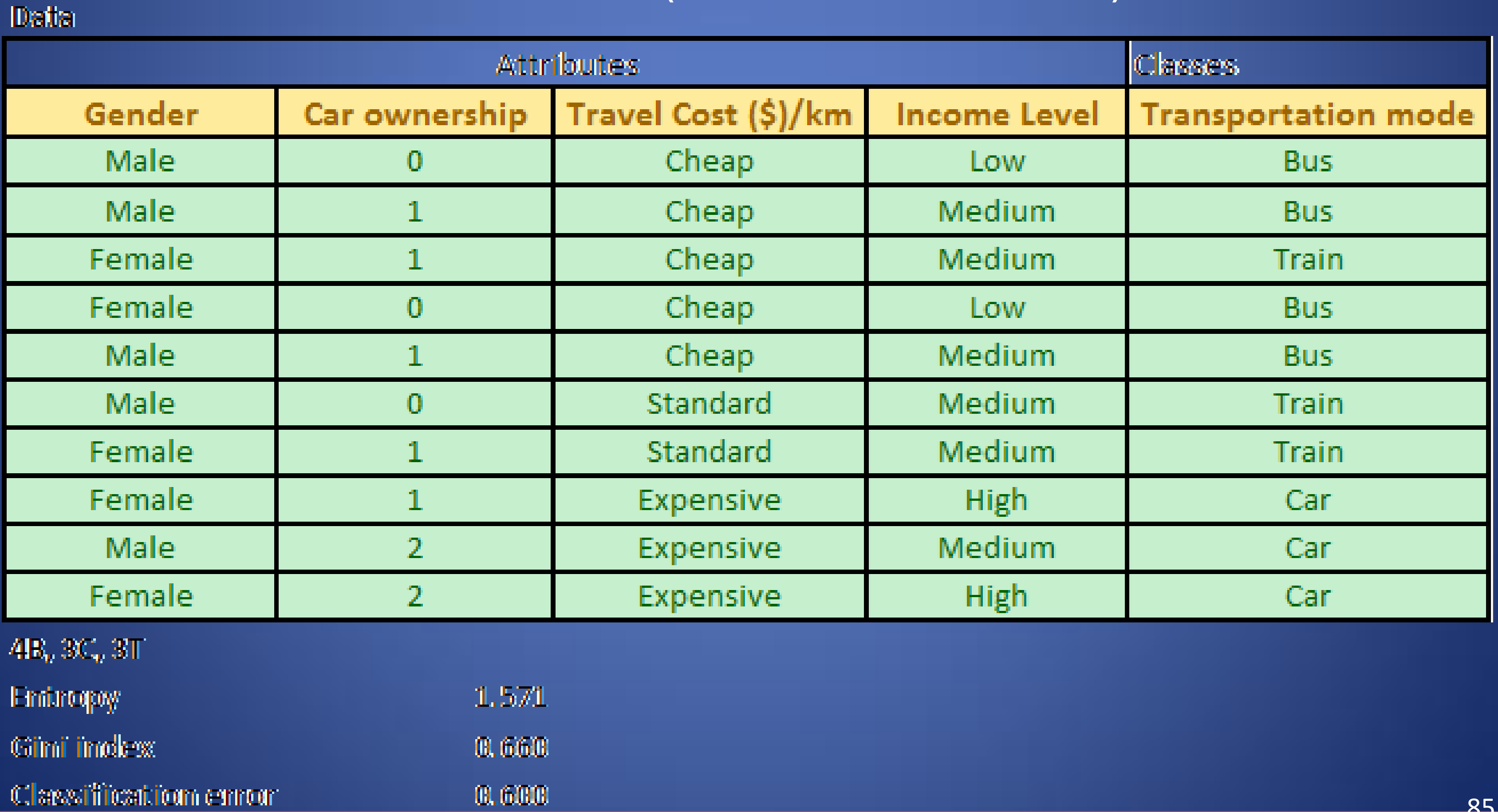
- Misclassification Error Index of a pure table (consist of a single class is zero because the probability is 1 and 1 - Max(1)=0.
- The value of classification error index is always between 0 and 1.
- In fact the maximum Gini index for a given number of classes is always equal to the maximum of misclassification error index because for a number of classes n, we set probability is equal to p=1/n and maximum Gini index happens at $1-n*(1/n)^2=1-1/n$, while maximum misclassification error index also happens at $1-\max\{1/n\}=1-1/n$.

Information Gain

- The reason for different ways of computation of impurity degrees between data table D and subset table S_i is because we would like to compare the difference of impurity degrees before we split the table (i.e. data table D) and after we split the table according to the values of an attribute i (i.e. subset table S_i) . The measure to compare the difference of impurity degrees is called information gain. We would like to know what our gain is if we split the data table based on some attribute values.

Information Gain - Example

- For example, in the parent table below, we can compute degree of impurity based on transportation mode. In this case we have 4 Busses, 3 Cars and 3 Trains (in short 4B, 3C, 3T):



Data

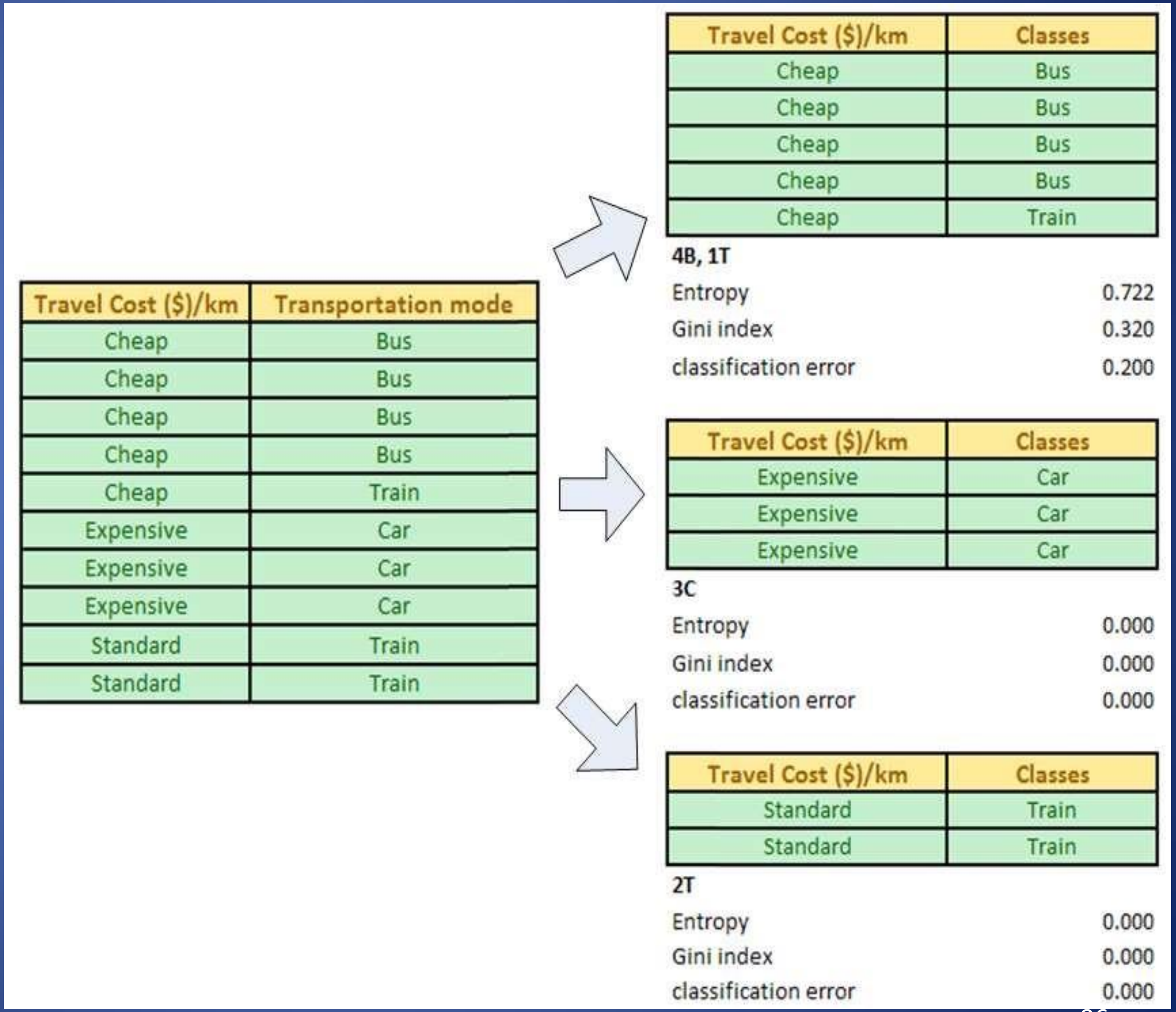
Attributes				Classes
Gender	**Car ownership**	**Travel Cost ($)/km**	**Income Level**	**Transportation mode**
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

4B, 3C, 3T

Entropy	1.571
Gini index	0.660
Classification error	0.600

Information Gain - Example

- For example, we split using travel cost attribute and compute the degree of impurity.



Travel Cost ($)/km	Transportation mode
Cheap	Bus
Cheap	Bus
Cheap	Bus
Cheap	Bus
Cheap	Train
Expensive	Car
Expensive	Car
Expensive	Car
Standard	Train
Standard	Train

Travel Cost ($)/km	Classes
Cheap	Bus
Cheap	Bus
Cheap	Bus
Cheap	Bus
Cheap	Train

4B, 1T

Entropy	0.722
Gini index	0.320
classification error	0.200

Travel Cost ($)/km	Classes
Expensive	Car
Expensive	Car
Expensive	Car

3C

Entropy	0.000
Gini index	0.000
classification error	0.000

Travel Cost ($)/km	Classes
Standard	Train
Standard	Train

2T

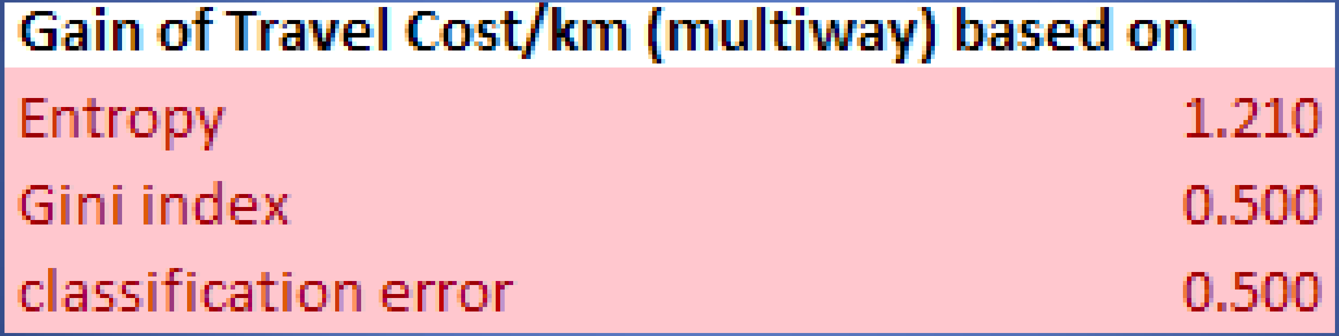
Entropy	0.000
Gini index	0.000
classification error	0.000

Information Gain - Example

- Information gain is computed as impurity degrees of the parent table and weighted summation of impurity degrees of the subset table. The weight is based on the number of records for each attribute values. Suppose we will use entropy as measurement of impurity degree, then we have:
- Information gain (i) = Entropy of parent table D – Sum (n k /n * Entropy of each value k of subset table Si)
- The information gain of attribute Travel cost per km is computed as 1.571 – (5/10 * 0.722+2/10*0+3/10*0) = 1.210

Information Gain - Example

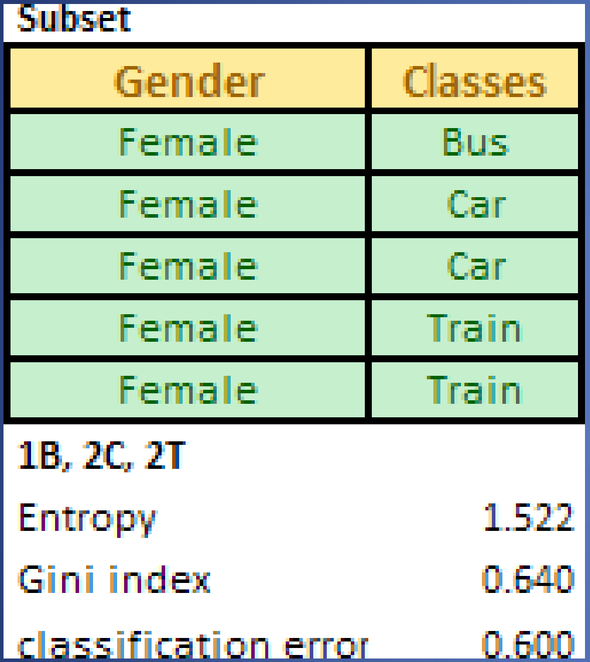
- You can also compute information gain based on Gini index or classification error in the same method. The results are given below.



Gain of Travel Cost/km (multiway) based on	
Entropy	1.210
Gini index	0.500
classification error	0.500

Information Gain – Example

- Split using “Gender” attribute

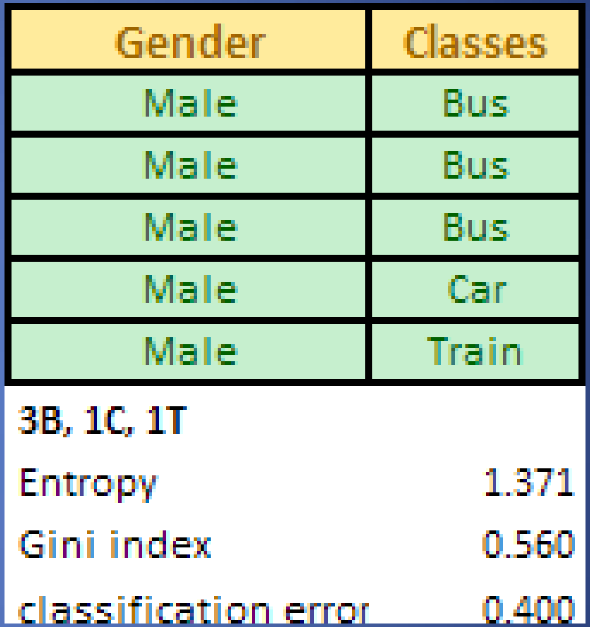


Subset

Gender	Classes
Female	Bus
Female	Car
Female	Car
Female	Train
Female	Train

1B, 2C, 2T

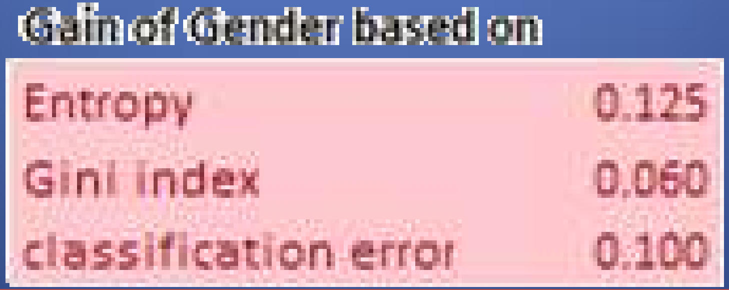
Entropy	1.522
Gini index	0.640
classification error	0.600



Gender	Classes
Male	Bus
Male	Bus
Male	Bus
Male	Car
Male	Train

3B, 1C, 1T

Entropy	1.371
Gini index	0.560
classification error	0.400

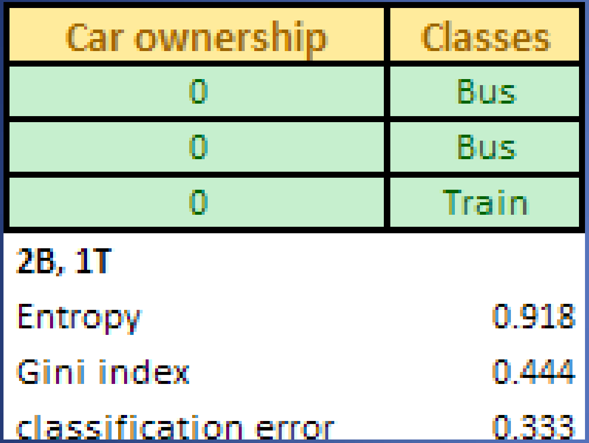


Gain of Gender based on

Entropy	0.125
Gini index	0.060
classification error	0.100

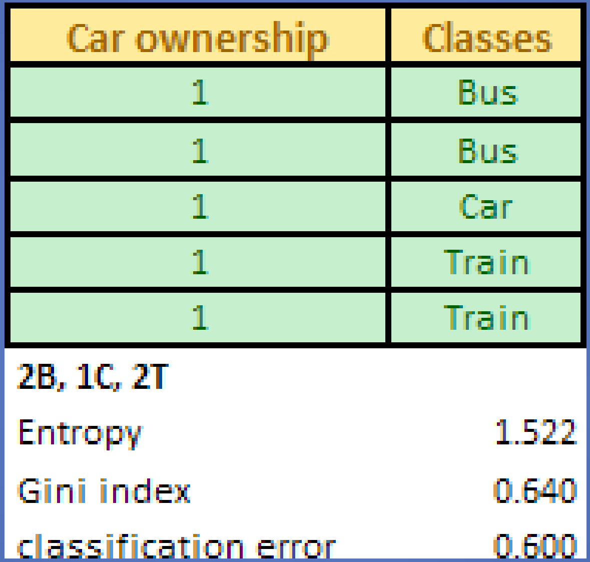
Information Gain - Example

- Split using “Car ownership” attribute



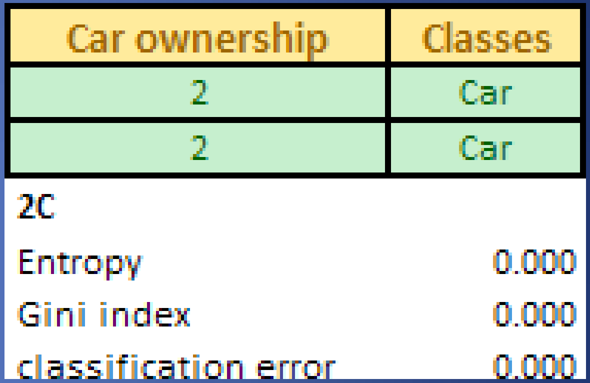
Car ownership	Classes
0	Bus
0	Bus
0	Train

2B, 1T	
Entropy	0.918
Gini index	0.444
classification error	0.333



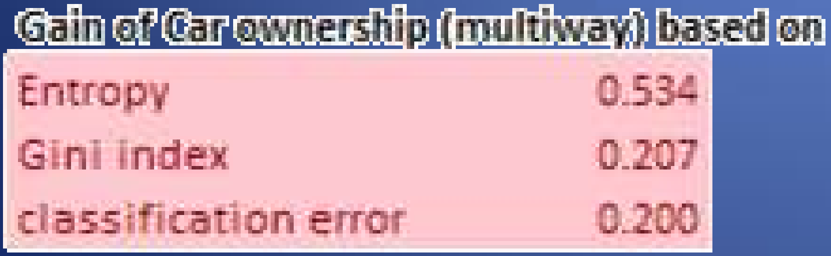
Car ownership	Classes
1	Bus
1	Bus
1	Car
1	Train
1	Train

2B, 1C, 2T	
Entropy	1.522
Gini index	0.640
classification error	0.600



Car ownership	Classes
2	Car
2	Car

2C	
Entropy	0.000
Gini index	0.000
classification error	0.000

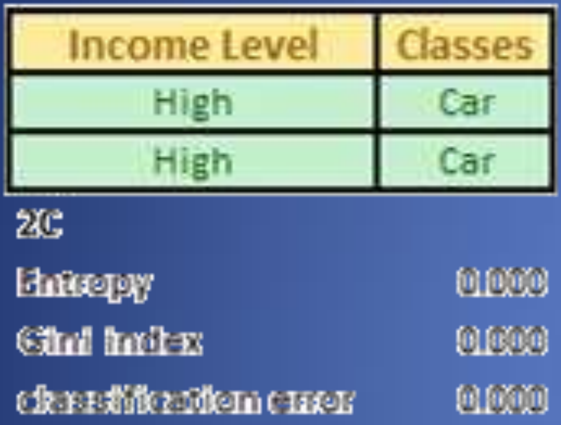


Gain of Car ownership (multiway) based on

Entropy	0.534
Gini index	0.207
classification error	0.200

Information Gain - Example

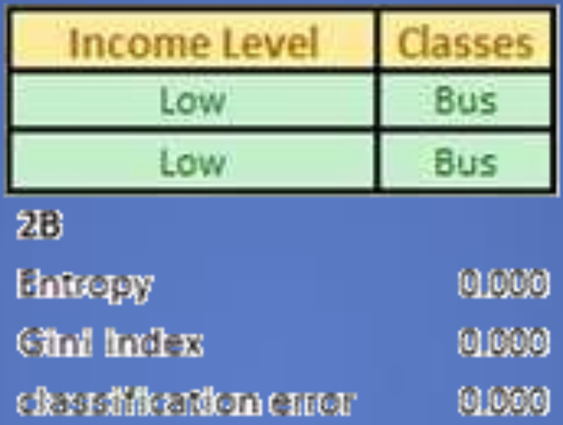
- Split using “Income Level” attribute



Income Level	Classes
High	Car
High	Car

2C

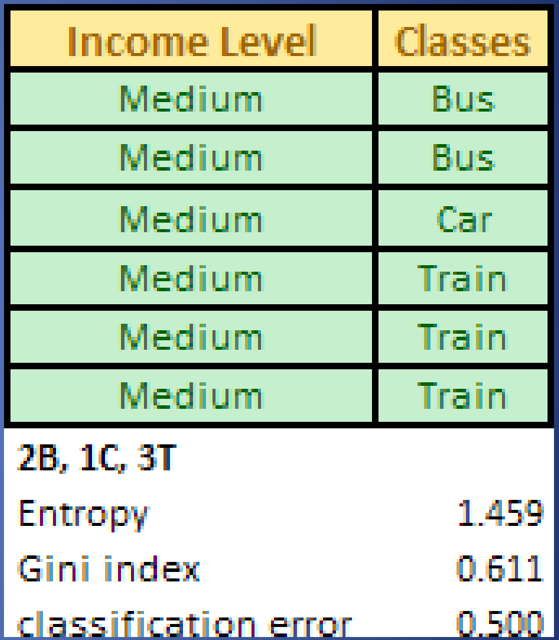
Entropy	0.000
Gini index	0.000
classification error	0.000



Income Level	Classes
Low	Bus
Low	Bus

2B

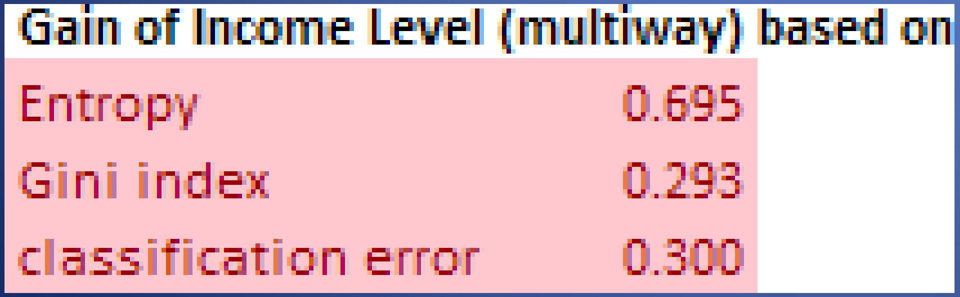
Entropy	0.000
Gini index	0.000
classification error	0.000



Income Level	Classes
Medium	Bus
Medium	Bus
Medium	Car
Medium	Train
Medium	Train
Medium	Train

2B, 1C, 3T

Entropy	1.459
Gini index	0.611
classification error	0.500

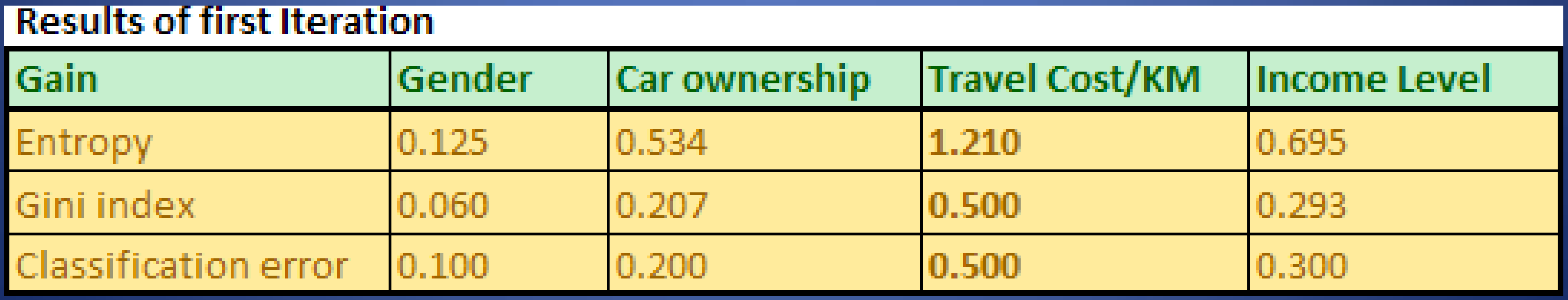


Gain of Income Level (multiway) based on

Entropy	0.695
Gini index	0.293
classification error	0.300

Information Gain - Example

- Table below summarizes the information gain for all four attributes. In practice, you don't need to compute the impurity degree based on three methods. You can use either one of Entropy or Gini index or index of classification error.
- Now we find the optimum attribute that produce the maximum information gain (i* = argmax {information gain of attribute i}). In our case, travel cost per km produces the maximum information gain.

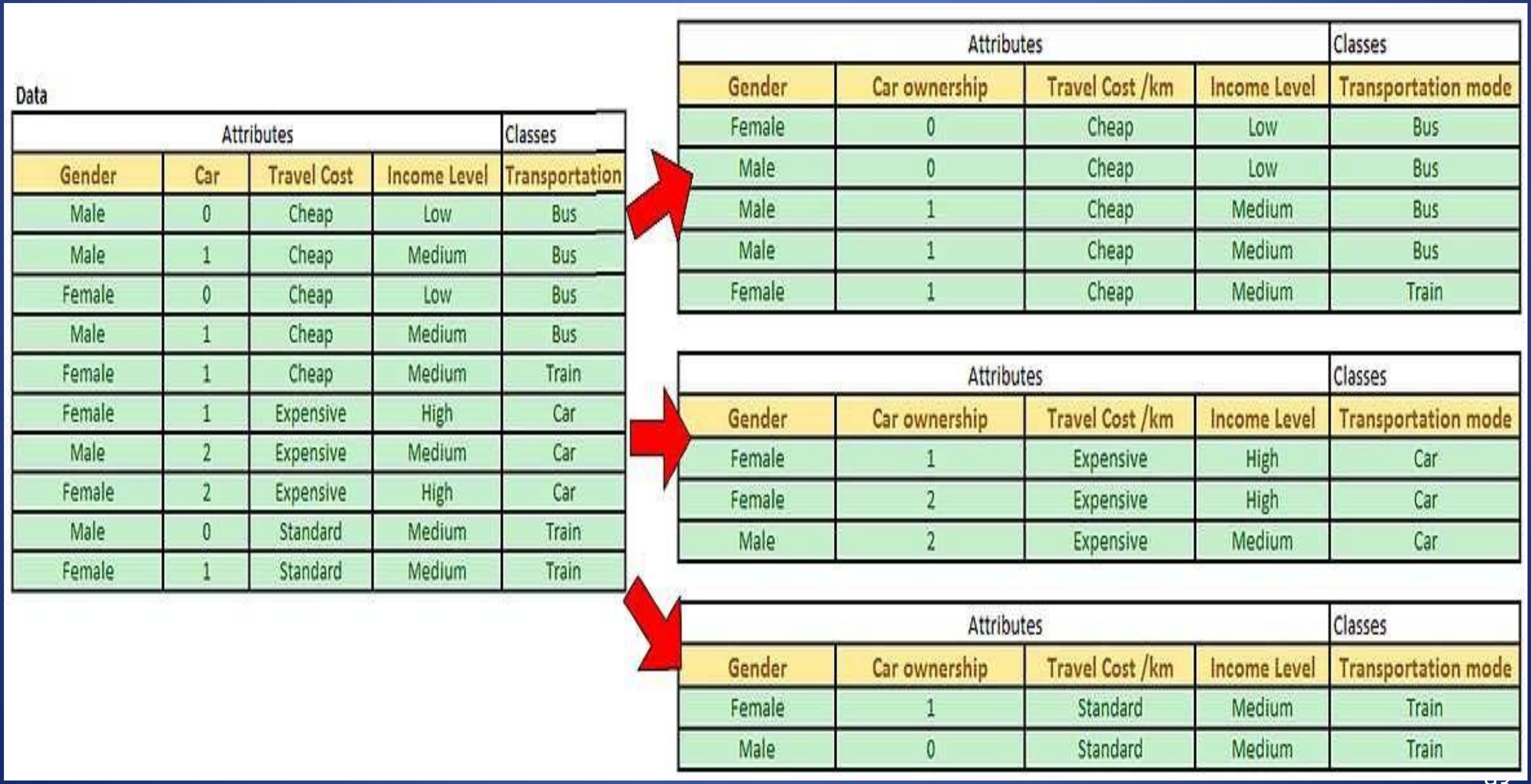


Results of first Iteration

Gain	Gender	Car ownership	Travel Cost/KM	Income Level
Entropy	0.125	0.534	**1.210**	0.695
Gini index	0.060	0.207	**0.500**	0.293
Classification error	0.100	0.200	**0.500**	0.300

Information Gain - Example

- So we split using “travel cost per km” attribute as this produces the maximum information gain.



Data

Attributes				Classes
Gender	Car	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train

Attributes				Classes
Gender	Car ownership	Travel Cost /km	Income Level	Transportation mode
Female	0	Cheap	Low	Bus
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train

Attributes				Classes
Gender	Car ownership	Travel Cost /km	Income Level	Transportation mode
Female	1	Expensive	High	Car
Female	2	Expensive	High	Car
Male	2	Expensive	Medium	Car

Attributes				Classes
Gender	Car ownership	Travel Cost /km	Income Level	Transportation mode
Female	1	Standard	Medium	Train
Male	0	Standard	Medium	Train

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

What is Cluster Analysis?

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation, …*)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Clustering for Data Understanding and Applications

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market resarch

Clustering as a Preprocessing Tool (Utility)

- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing: vector quantization
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection
  - Outliers are often viewed as those “far away” from any cluster

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation, and
  - 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: $d(i, j)$
- The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
- Weights should be associated with different variables based on applications and data semantics

Quality of clustering:

There is usually a separate “quality” function that measures the “goodness” of a cluster.

It is hard to define “similar enough” or “good enough”

The answer is typically highly subjective

Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
  - Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

Requirements and Challenges

- Scalability
  - Clustering all the data instead of only on samples
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters
- Interpretability and usability
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality

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, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Major Clustering Approaches (II)

- 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: p-Cluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Partitioning Algorithms: Basic Concept

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

- Given *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, Lloyd'57/'82): 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 partitioning (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 the assignment does not change

An Example of ***K-Means*** Clustering

K=2

Arbitrarily partition objects into k groups

Update the cluster centroids

The initial data set

Loop if needed

Reassign objects

Update the cluster centroids

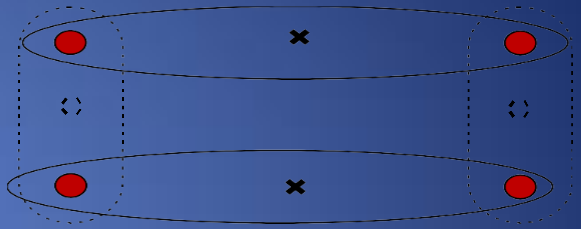
- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

Comments on the *K-Means* Method

- Strength: *Efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
    - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*.
- Weakness
    - Applicable only to objects in a continuous n-dimensional space
        - Using the k-modes method for categorical data
        - In comparison, k-medoids can be applied to a wide range of data
    - Need to specify *k, the number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
    - Sensitive to noisy data and *outliers*
    - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

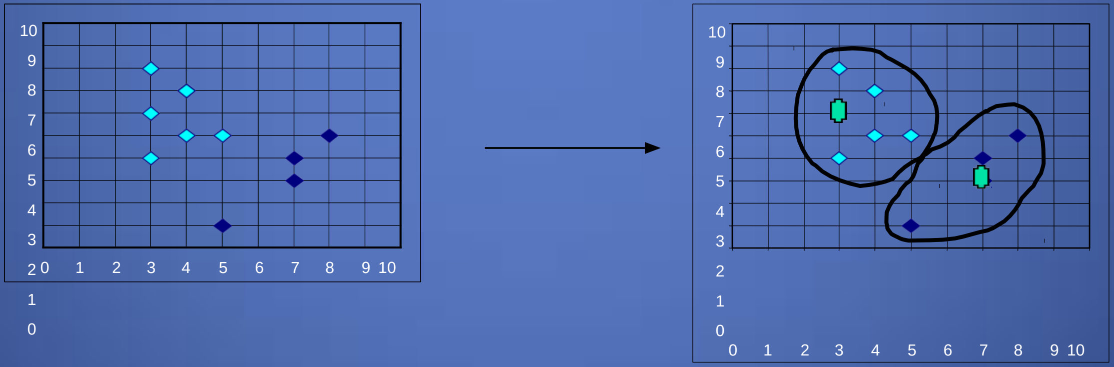
- Most of the variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means



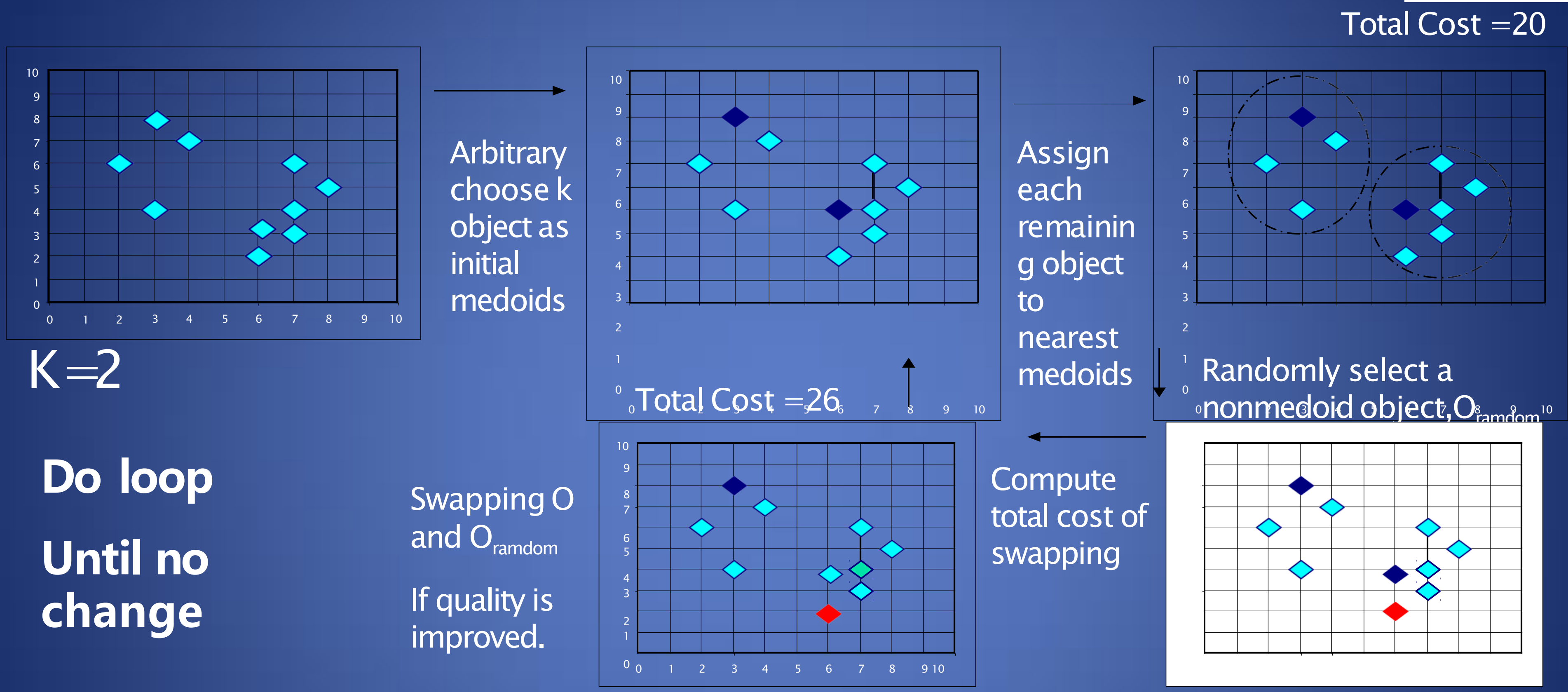
- Handling categorical data: *k-modes*
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

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



PAM: A Typical K-Medoids Algorithm



K=2

Arbitrary choose k object as initial medoids

Assign each remaining object to nearest medoids

Total Cost = 20

Randomly select a nonmedoid object, O_{ramdom}

Compute total cost of swapping

Total Cost = 26

Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

The K-Medoid Clustering Method

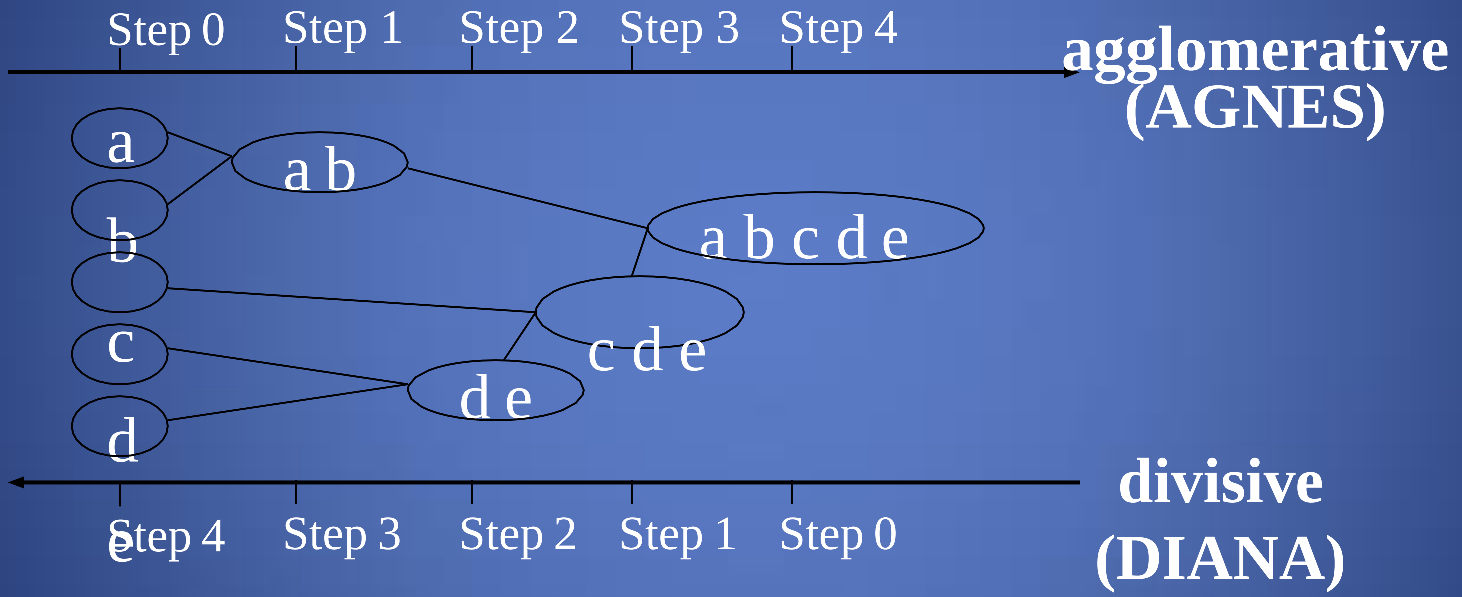
- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
    - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- 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



Step 0 Step 1 Step 2 Step 3 Step 4

agglomerative (AGNES)

a

b

c

d

e

a b

d e

c d e

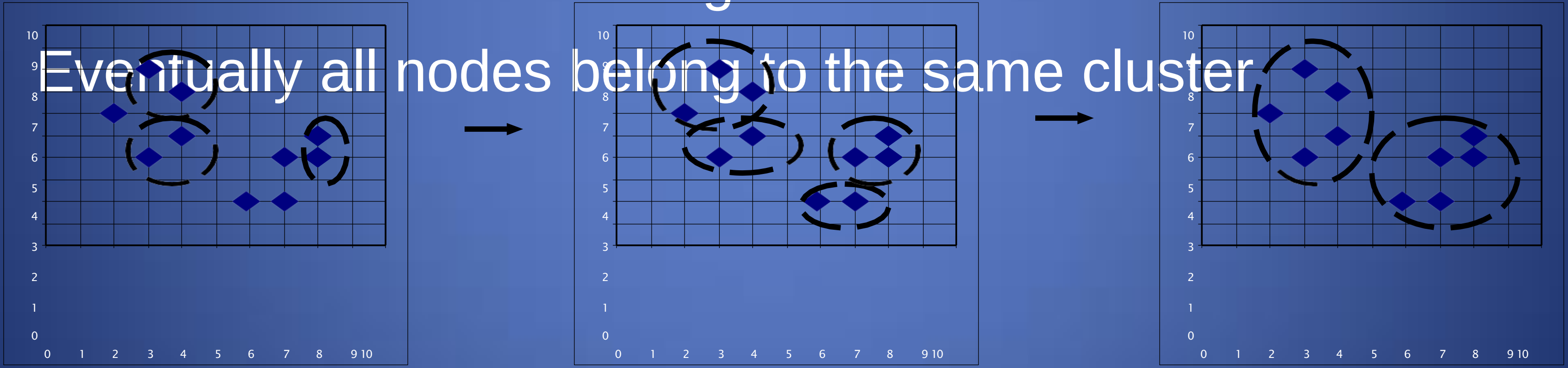
a b c d e

Step 4 Step 3 Step 2 Step 1 Step 0

divisive (DIANA)

AGNES (Agglomerative Nesting)

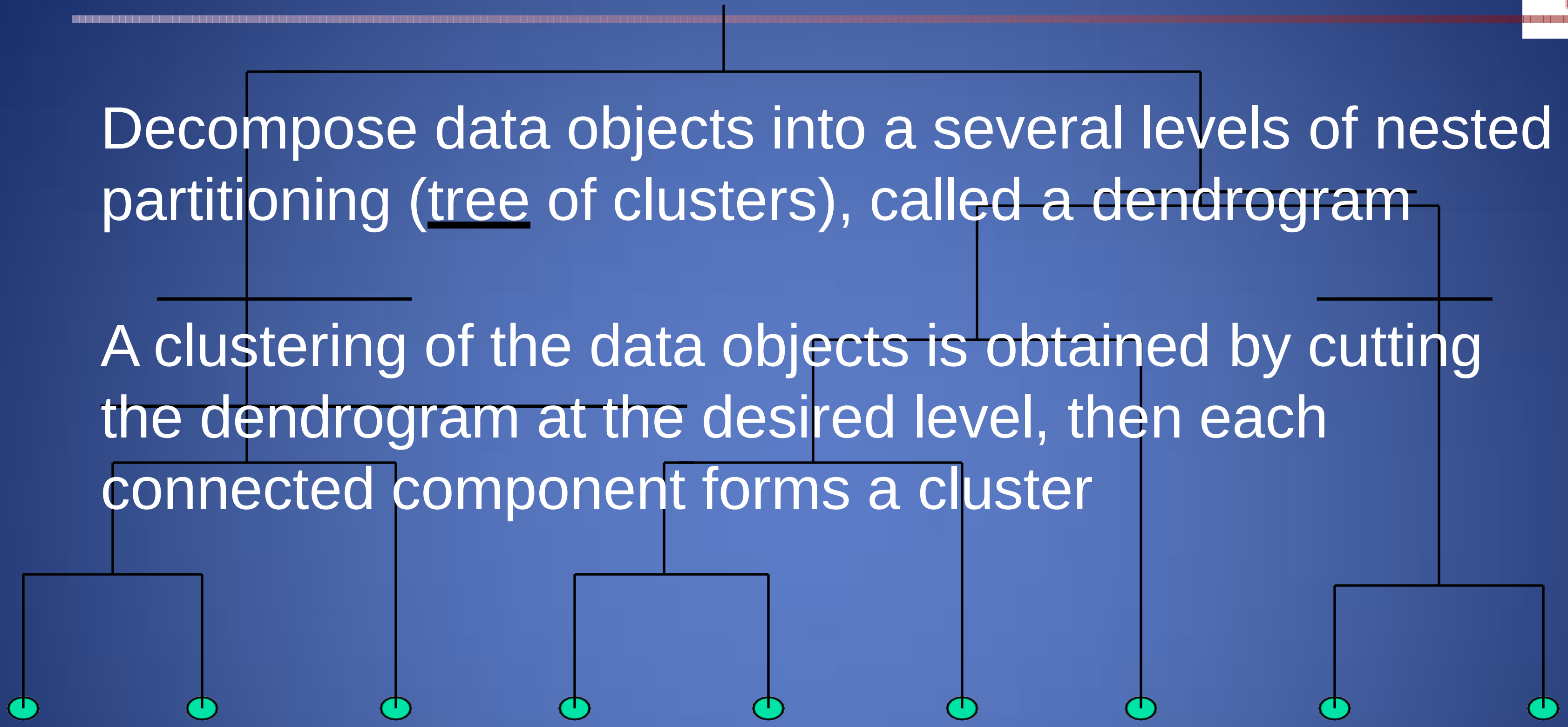
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How Clusters are Merged

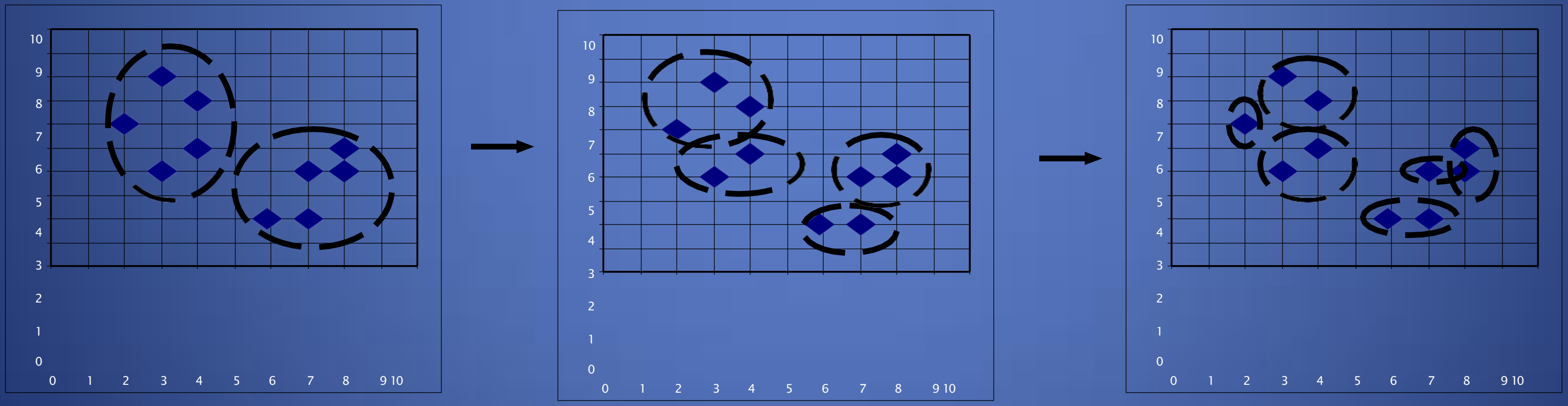
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

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

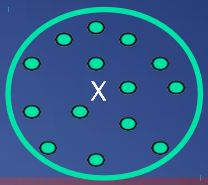


DIANA (Divisive Analysis)

- 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



Distance between Clusters



- Single link: smallest distance between an element in one cluster and an element in the other, i.e., dist(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., dist(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., dist(K_i, K_j) = avg(t_{ip}, t_{jq})
- Centroid: distance between the centroids of two clusters, i.e., dist(K_i, K_j) = dist(C_i, C_j)
- Medoid: a chosen, centrally located object in the cluster

Medoid: distance between the medoids of two clusters, i.e., dist(K_i, K_j) = dist(M_i, M_j)

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

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

Extensions to Hierarchical Clustering

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

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

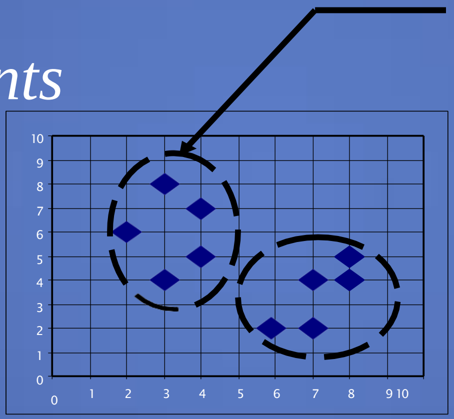
Clustering Feature Vector in BIRCH

Clustering Feature (CF): ***CF = (N, LS, SS)***

N: **Number of data points**

LS: linear sum of N points: $\sum_{i=1}^{N} X_i$

SS: square sum of N points $\sum_{i=1}^{N} X_i^2$



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

(3,4)

(2,6)

(4,5)

(4,7)

(3,8)

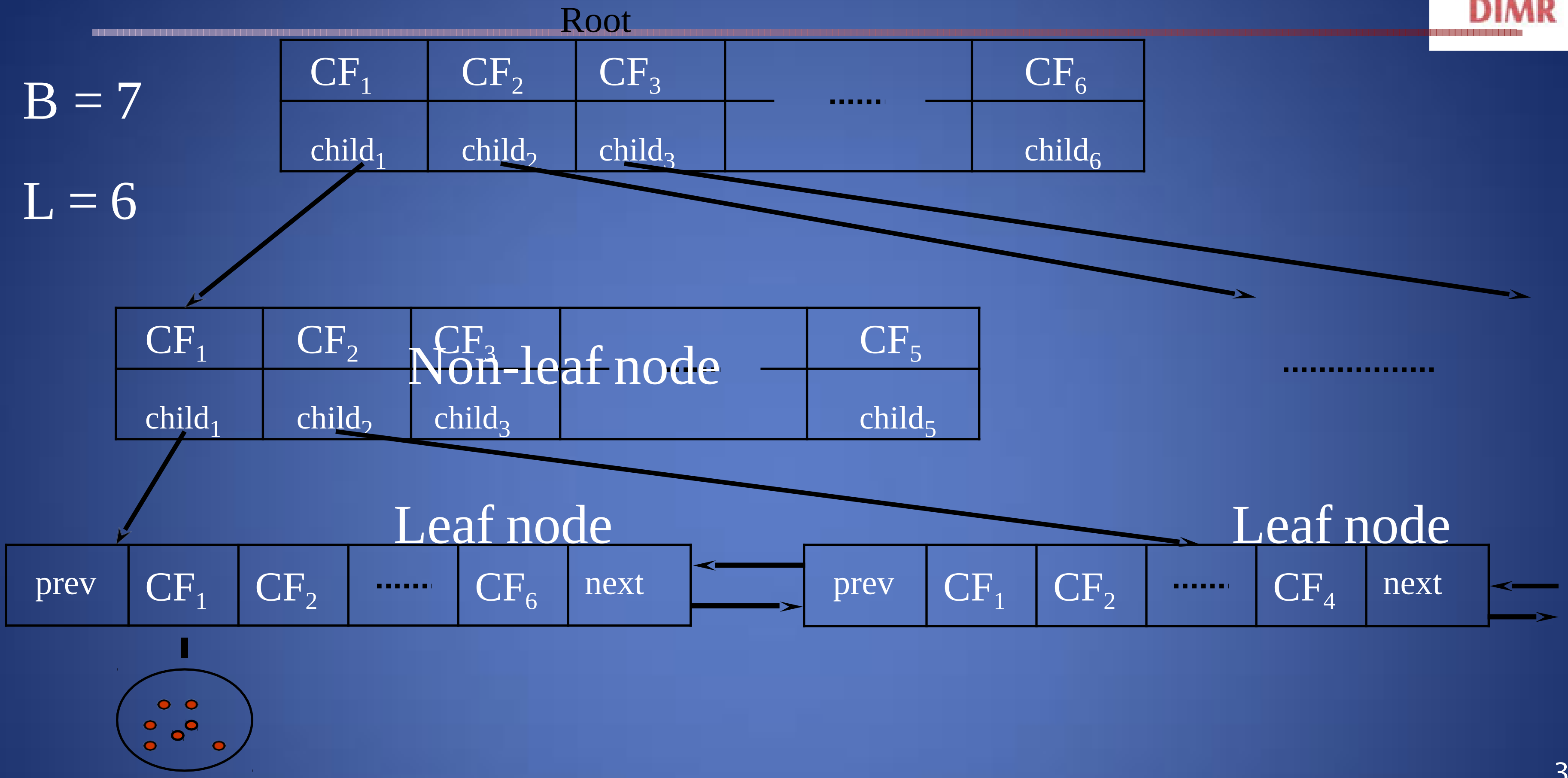
CF-Tree in BIRCH

- 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: max # of children
  - Threshold: max diameter of sub-clusters stored at the leaf nodes

The CF Tree Structure

B = 7

L = 6



Root

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

Leaf node

prev | CF_1 | CF_2 | .... | CF_4 | next

The Birch Algorithm

- Cluster Diameter

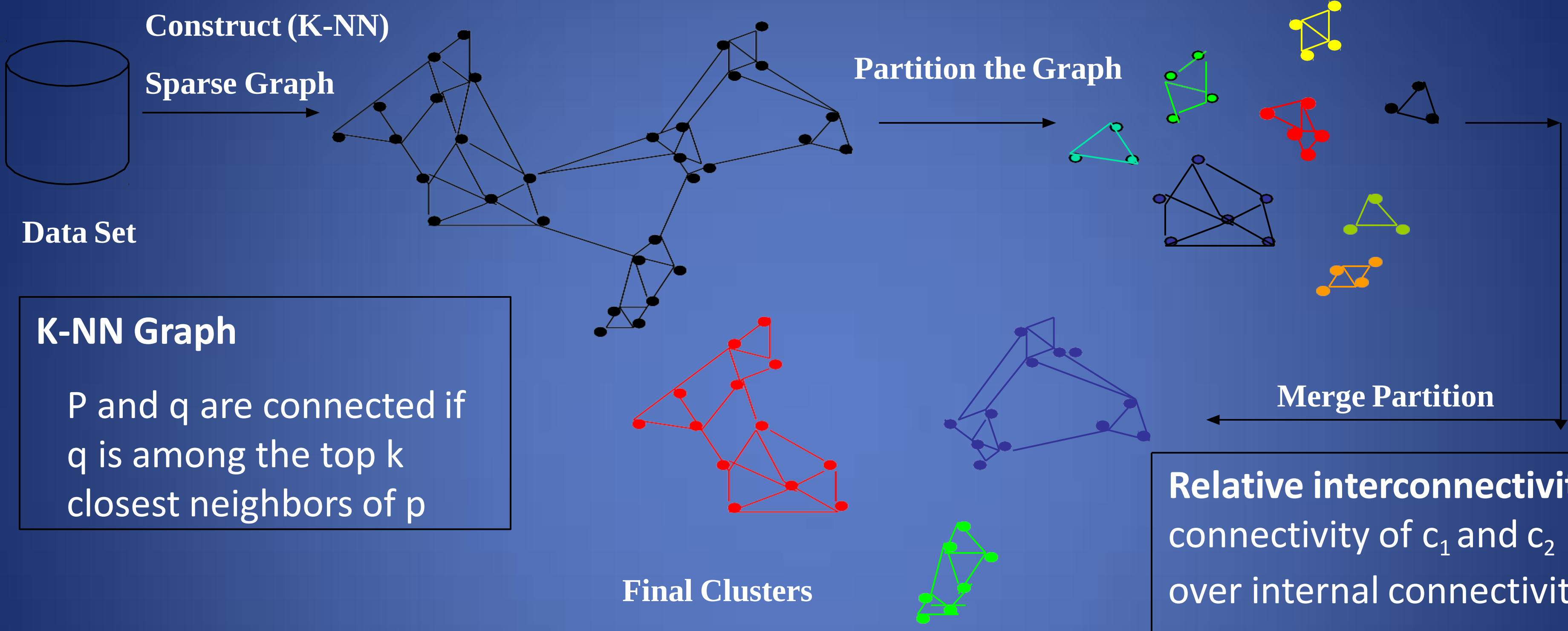
$$\sqrt{\frac{1}{n(n-1)}\sum (x_i - x_j)^2}$$

- For each point in the input
  - Find closest leaf entry
  - Add point to leaf entry and update CF
  - If entry diameter > max_diameter, then split leaf, and possibly parents
- Algorithm is O(n)
- Concerns
  - Sensitive to insertion order of data points
  - Since we fix the size of leaf nodes, so clusters may not be so natural
  - Clusters tend to be spherical given the radius and diameter measures

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999

- CHAMELEON: G. Karypis, E. H. Han, and V. Kumar, 1999
- 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
- Graph-based, and 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



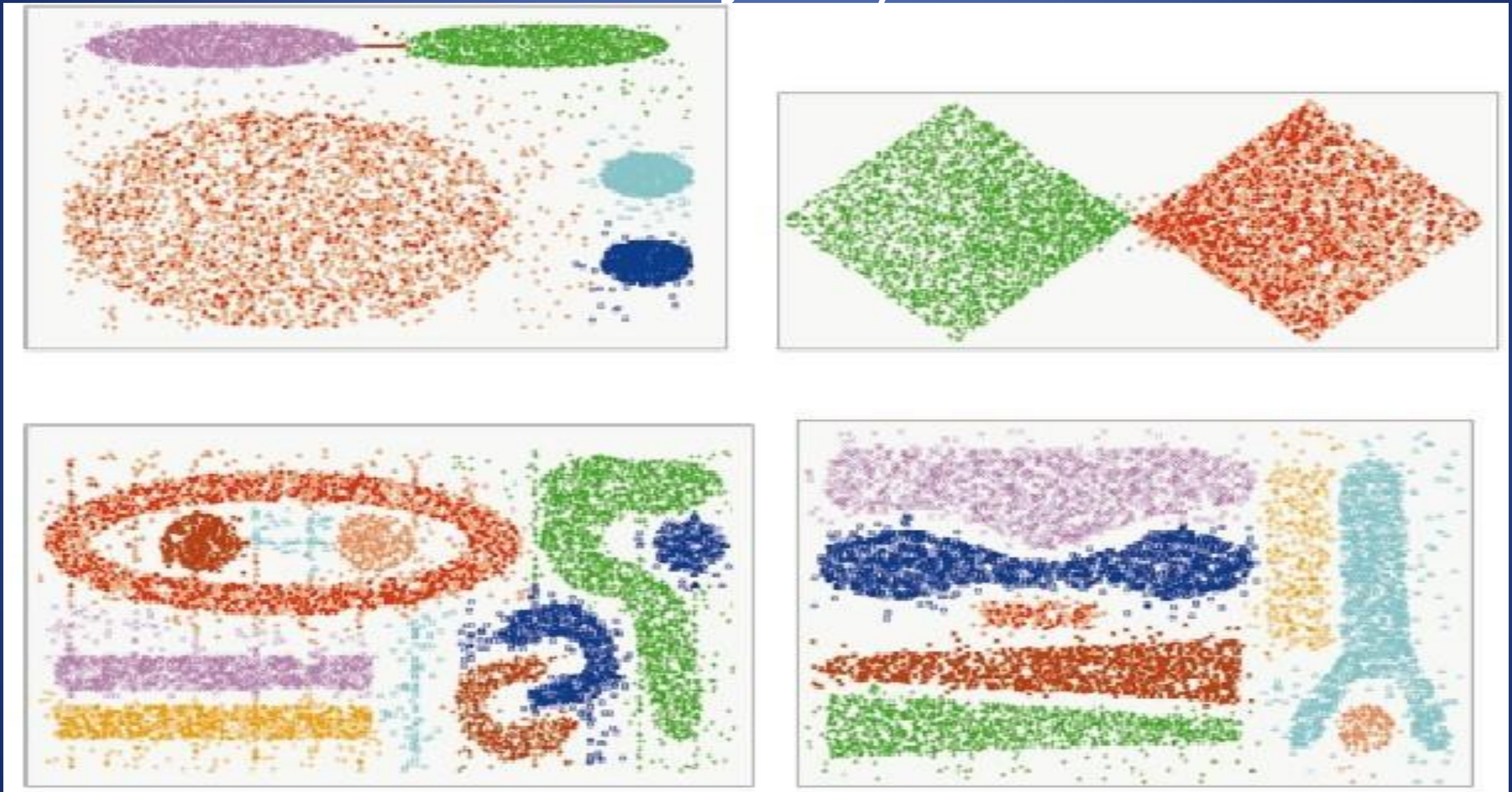
K-NN Graph

P and q are connected if q is among the top k closest neighbors of p

Relative interconnectivity: connectivity of c_1 and c_2 over internal connectivity

Relative closeness: closeness of c_1 and c_2 over internal closeness

CHAMELEON (Clustering Complex Objects)



Probabilistic Hierarchical Clustering

- Algorithmic hierarchical clustering
  - Nontrivial to choose a good distance measure
  - Hard to handle missing attribute values
  - Optimization goal not clear: heuristic, local search
- Probabilistic hierarchical clustering
  - Use probabilistic models to measure distances between clusters
  - Generative model: Regard the set of data objects to be clustered as a sample of the underlying data generation mechanism to be analyzed
  - Easy to understand, same efficiency as algorithmic agglomerative clustering method, can handle partially observed data

In practice, assume the generative models adopt common distributions functions, e.g., Gaussian distribution or Bernoulli distribution, governed by parameters

Generative Model

- Given a set of 1-D points $X = \{x_1, \ldots, x_n\}$ for clustering analysis & assuming they are generated by a Gaussian distribution:

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

- The probability that a point $x_i \in X$ is generated by the model

$$P(x_i|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- The likelihood that X is generated by the model:

$$L(\mathcal{N}(\mu, \sigma^2) : X) = P(X|\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- The task of learning the generative model: find the parameters μ and σ^2 such that

the maximum likelihood

$$\mathcal{N}(\mu_0, \sigma_0^2) = \arg\max\{L(\mathcal{N}(\mu, \sigma^2) : X)\}$$

A Probabilistic Hierarchical Clustering Algorithm

- For a set of objects partitioned into m clusters $C_1, \ldots, C_m$, the quality can be measured by, $$Q(\{C_1, \ldots, C_m\}) = \prod_{i=1}^{m} P(C_i)$$

  where $P()$ is the maximum likelihood

- Distance between clusters C_1 and C_2: $$dist(C_i, C_j) = -\log \frac{P(C_1 \cup C_2)}{P(C_1)P(C_2)}$$

- Algorithm: Progressively merge points and clusters

  Input: $D = \{o_1, \ldots, o_n\}$: a data set containing n objects

  Output: A hierarchy of clusters

  Method

```
    Create a cluster for each object $C_i = \{o_i\}$, $1 \le i \le n$;
    For i = 1 to n {
        Find pair of clusters $C_i$ and $C_j$ such that
            $C_i, C_j = \text{argmax}_{i \ne j}\{\log (P(C_i \cup C_j)/(P(C_i)P(C_j))\}$;
        If $\log (P(C_i \cup C_j)/(P(C_i)P(C_j)) > 0$ then merge $C_i$ and $C_j$ }
```

Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

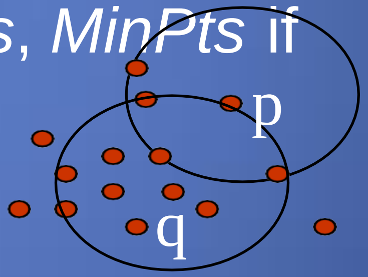
Density-Based Clustering Methods

- 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 belongs to D | dist(p,q) ≤ 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:

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

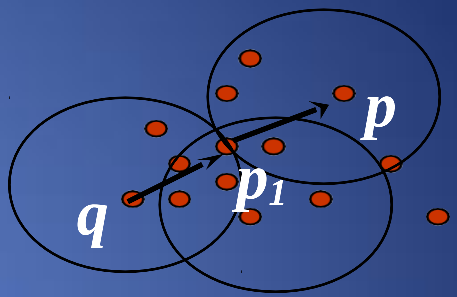


MinPts = 5

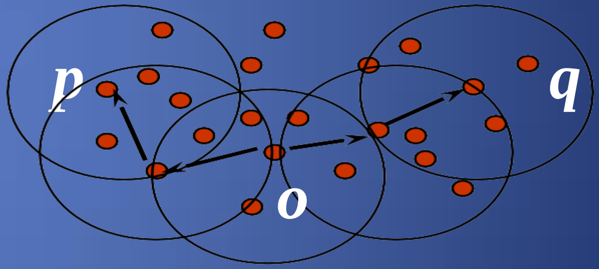
Eps = 1 cm

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 $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

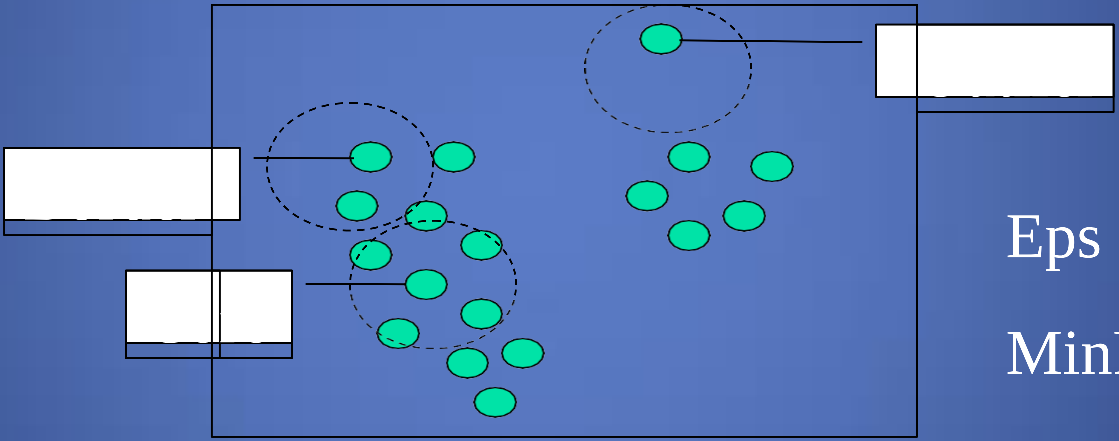


- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



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



Eps = 1cm

MinPts = 5

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.

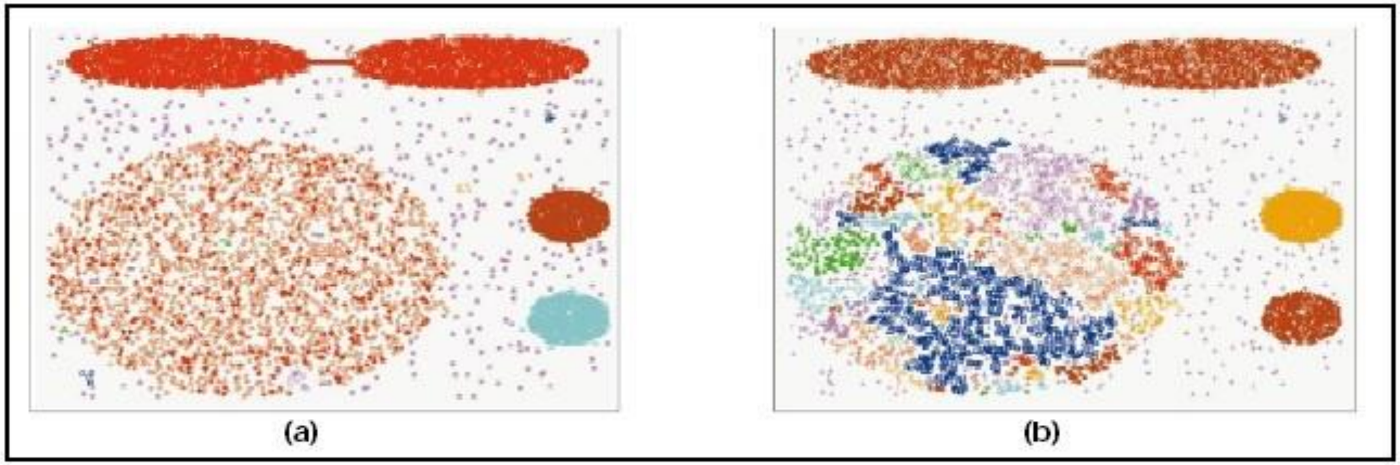
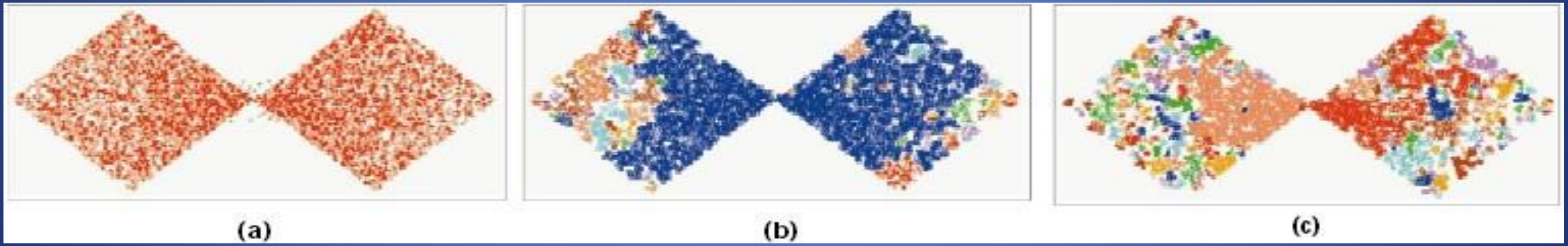


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



OPTICS: A Cluster-Ordering Method (1999)

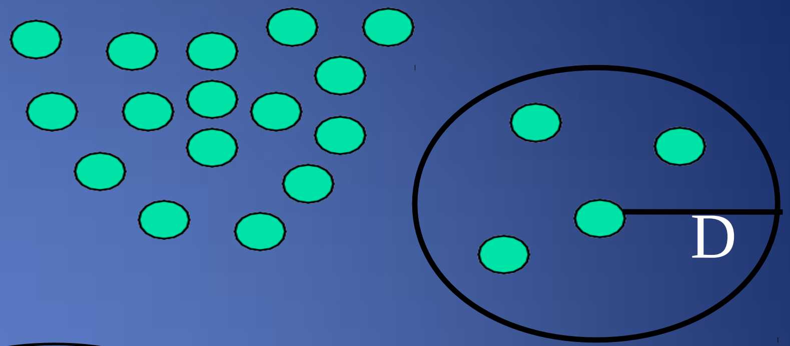
- OPTICS: Ordering Points To Identify the Clustering Structure
  - Ankerst, Breunig, Kriegel, and Sander (SIGMOD’99)
  - Produces a special order of the database wrt its density-based clustering structure
  - This cluster-ordering contains info equiv to the density-based clusterings corresponding to 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

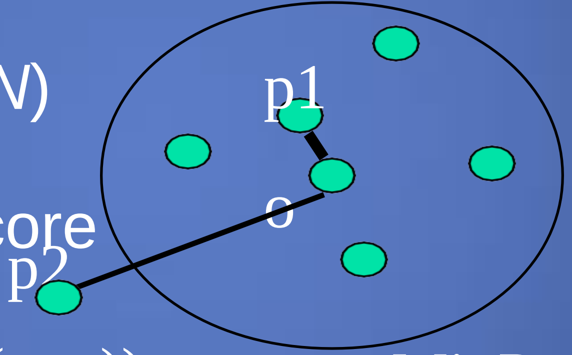
OPTICS: Some Extension from DBSCAN

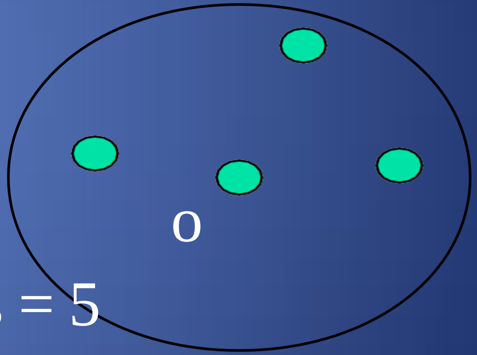
- Index-based:
  - k = number of dimensions
  - N = 20
  -
  - p = 75%
  - M = N(1-p) = 5
  - Complexity: O(*NlogN*)
- Core Distance:
  - min eps s.t. point is core
- Reachability Distance

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

r(p1, o) = 2.8cm. r(p2,o) = 4cm



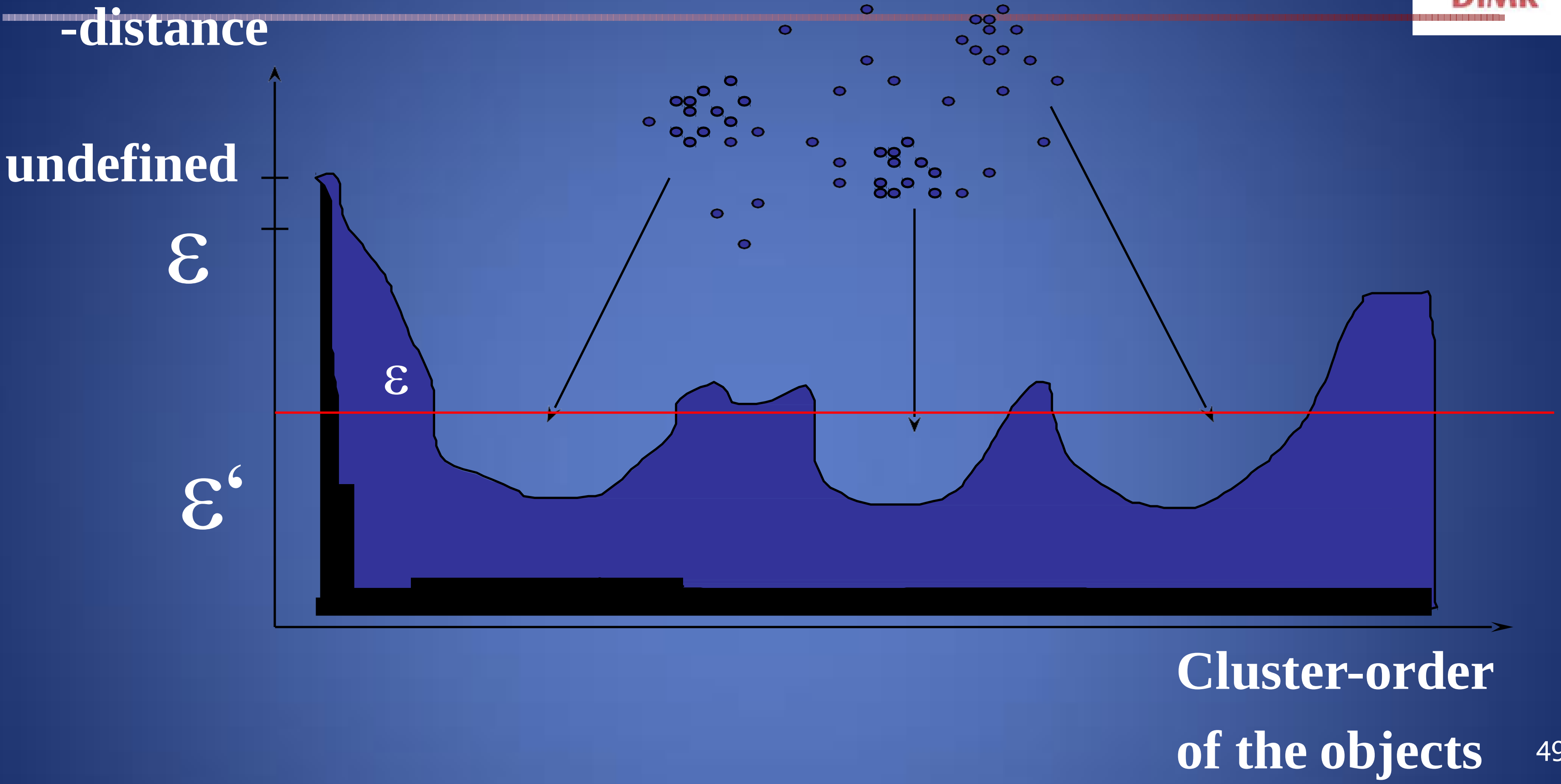




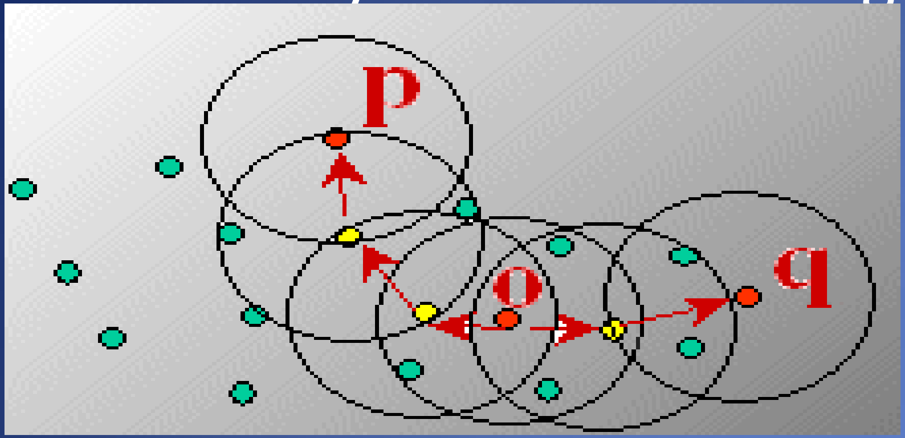
MinPts = 5

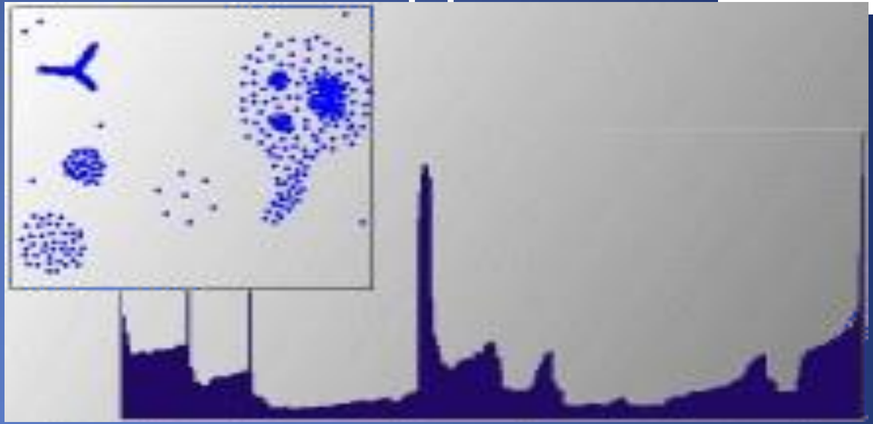
ε = 3 cm

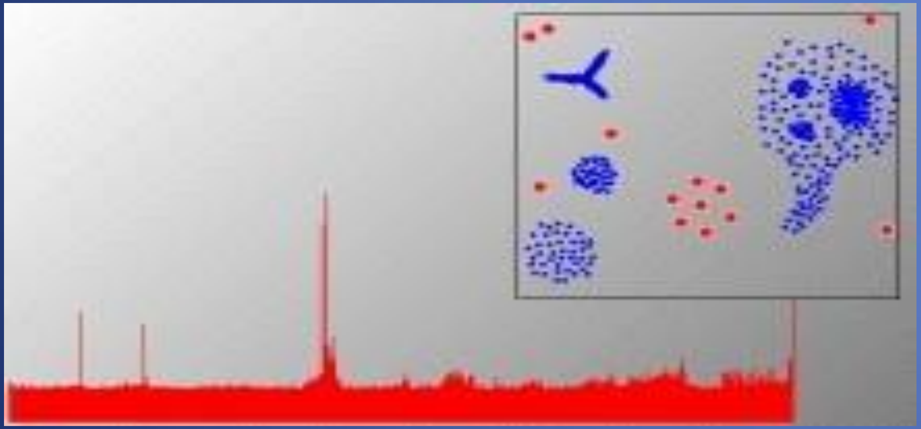
Reachability -distance

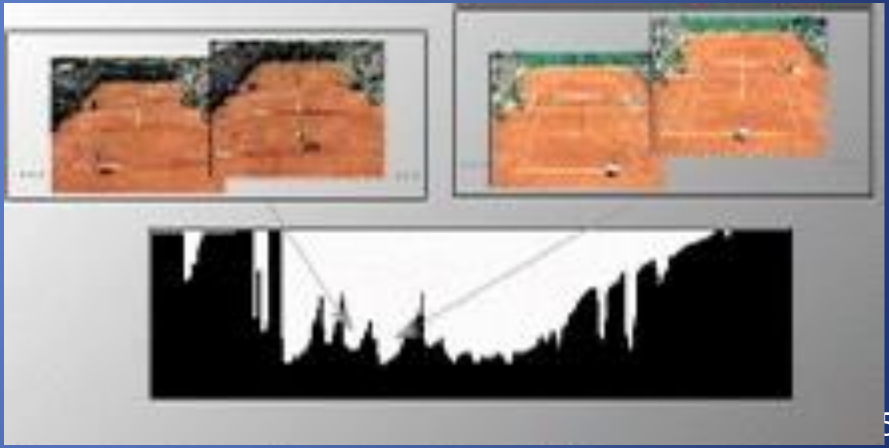


Density-Based Clustering: OPTICS & Its Applications









DENCLUE: Using Statistical Densit Functions

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

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

influence of y on x

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

total influence on x

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

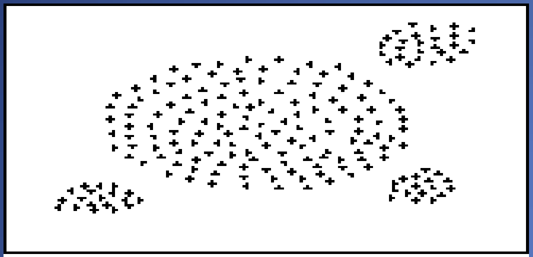
gradient of x in the direction of x_i

- 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

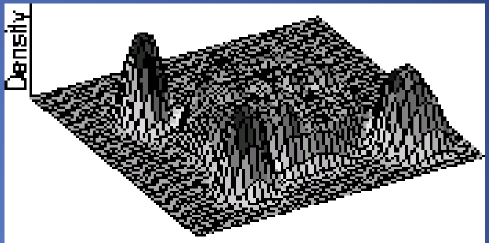
Denclue: Technical Essence

- 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
- Center defined clusters: assign to each density attractor the points density attracted to it
- Arbitrary shaped cluster: merge density attractors that are connected through paths of high density (> threshold)

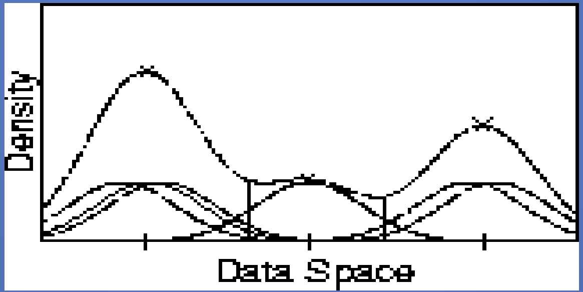
Density Attractor



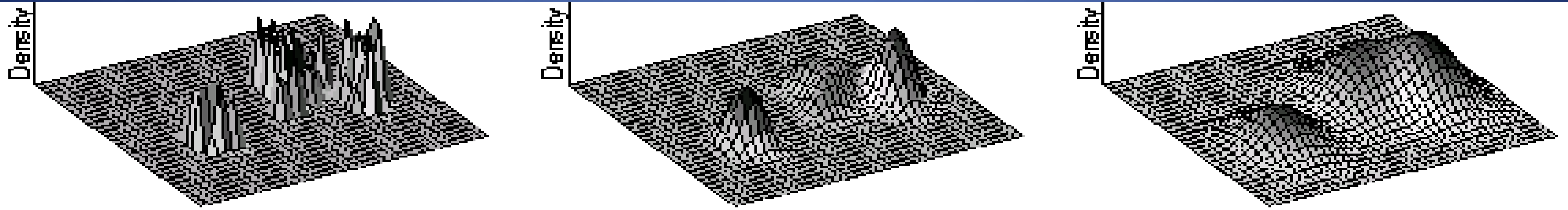
(a) Data Set



(c) Gaussian

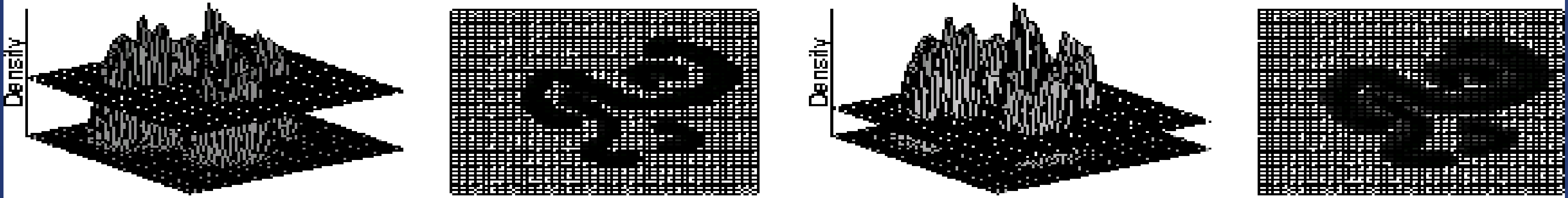


Center-Defined and Arbitrary



(a) $\sigma = 0.2$ (b) $\sigma = 0.6$ (d) $\sigma = 1.5$

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



(a) $\xi = 2$ (b) $\xi = 2$ (c) $\xi = 1$ (d) $\xi = 1$

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

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Grid-Based Clustering Method

- Using multi-resolution grid data structure
- Several interesting methods
  - A multi-resolution clustering approach using wavelet method
  - CLIQUE: Agrawal, et al. (SIGMOD’98)
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB’98)
  - Both grid-based and subspace clustering

STING: A Statistical Information Grid Approach

- Wang, Yang and Muntz (VLDB’97)
- The spatial 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

STING Algorithm and Its Analysis

- 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

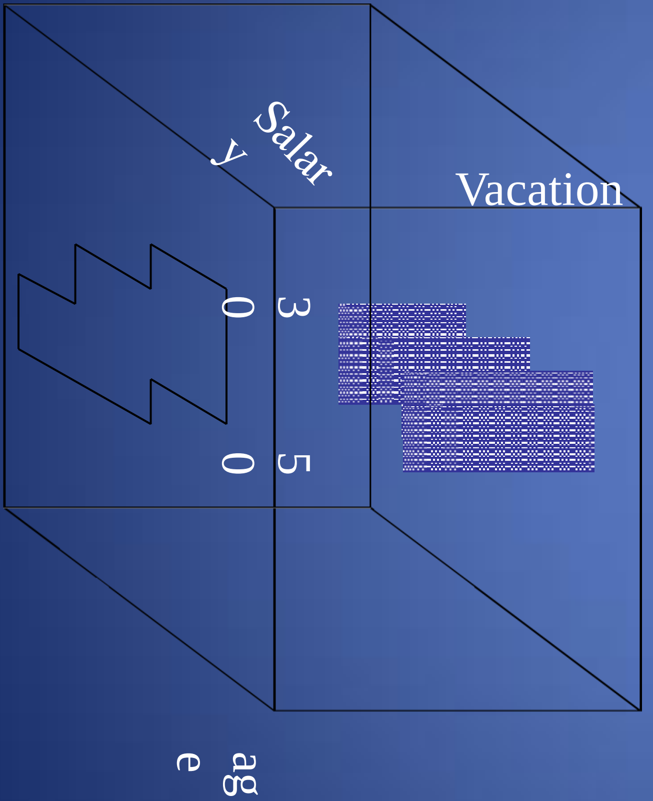
CLIQUE (Clustering In QUEst)

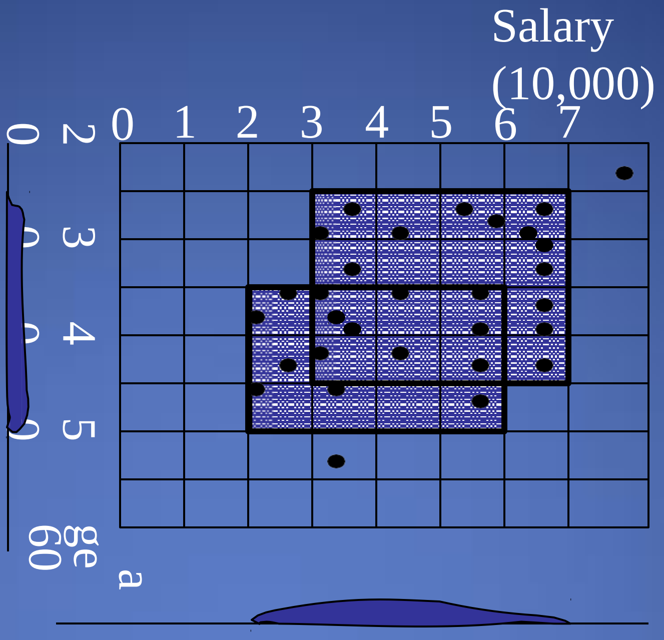
- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD’98)
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
  - It partitions each dimension into the same number of equal length 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

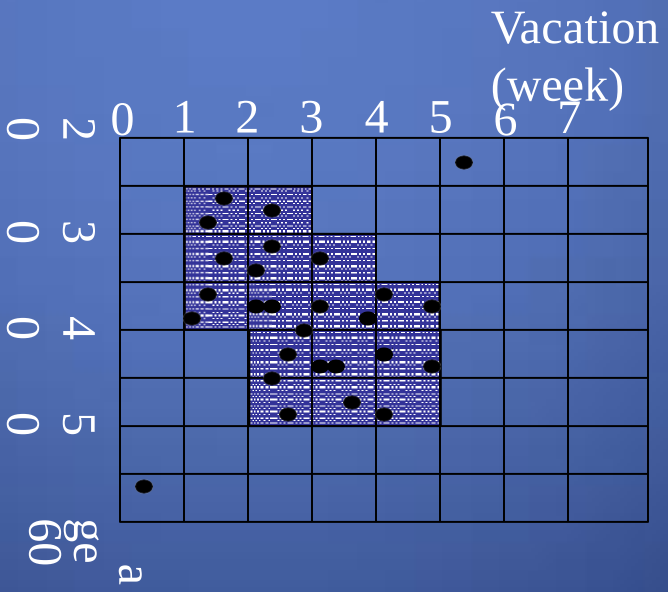
CLIQUE: The Major Steps

- 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

τ = 3


Salary
Vacation
30
50
age


Salary (10,000)
0
1
2
3
4
5
6
7
age
20
30
40
50
60


Vacation (week)
0
1
2
3
4
5
6
7
age
20
30
40
50
60

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

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering ⬅
- Summary

Assessing Clustering Tendency

- Assess if non-random structure exists in the data by measuring the probability that the data is generated by a uniform data distribution
- Test spatial randomness by statistic test: Hopkins Static
  - Given a dataset D regarded as a sample of a random variable o, determine how far away o is from being uniformly distributed in the data space
  - Sample *n* points, $p_1, \ldots, p_n$, uniformly from D. For each p_i, find its nearest neighbor in D: $x_i = min\{dist\ (p_i, v)\}$ where *v* in D
  - Sample *n* points, $q_1, \ldots, q_n$, unifor nd its nearest neighbor in D – $\{q_i\}$: $y_i =$ D and v ≠ q_i
  - Calculate the Hopkins Statistic:

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

  - If D is uniformly distributed, ∑ x_i and ∑ y_i will be close to each other and H is close to 0.5. If D is highly skewed, H is close to 0

Determine the Number of Clusters

- Empirical method
  - # of clusters ≈√n/2 for a dataset of n points
- Elbow method
  - Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters
- Cross validation method
  - Divide a given data set into *m* parts
  - Use *m* – 1 parts to obtain a clustering model
  - Use the remaining part to test the quality of the clustering
    - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
  - For any k > 0, repeat it *m* times, compare the overall quality measure w.r.t. different *k's*, and find # of clusters that fits the data the best

Measuring Clustering Quality

- Two methods: extrinsic vs. intrinsic
- Extrinsic: supervised, i.e., the ground truth is available
  - Compare a clustering against the ground truth using certain clustering quality measure
  - Ex. BCubed precision and recall metrics
- Intrinsic: unsupervised, i.e., the ground truth is unavailable
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
  - Ex. Silhouette coefficient

Measuring Clustering Quality: Extrinsic Methods

- Clustering quality measure: *Q(C, C$_g$)*, for a clustering *C* given the ground truth *C$_g$*.
- *Q* is good if it satisfies the following 4 essential criteria
  - Cluster homogeneity: the purer, the better
  - Cluster completeness: should assign objects belong to the same category in the ground truth to the same cluster
  - Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., “miscellaneous” or “other” category)
  - Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- 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
- K-means and K-medoids algorithms are popular partitioning-based clustering algorithms
- Birch and Chameleon are interesting hierarchical clustering algorithms, and there are also probabilistic hierarchical clustering algorithms
- DBSCAN, OPTICS, and DENCLU are interesting density-based algorithms
- STING and CLIQUE are grid-based methods, where CLIQUE is also a subspace clustering algorithm
- Quality of clustering results can be evaluated in various ways

CS512-Spring 2011: An Introduction

- Coverage
  - Cluster Analysis: Chapter 11
  - Outlier Detection: Chapter 12
  - Mining Sequence Data: BK2: Chapter 8
  - Mining Graphs Data: BK2: Chapter 9
  - Social and Information Network Analysis
    - BK2: Chapter 9
    - Partial coverage: Mark Newman: “Networks: An Introduction”, Oxford U., 2010
    - Scattered coverage: Easley and Kleinberg, “Networks, Crowds, and Markets: Reasoning About a Highly Connected World”, Cambridge U., 2010
    - Recent research papers
  - Mining Data Streams: BK2: Chapter 8
- Requirements
  - One research project
  - One class presentation (15 minutes)
  - Two homeworks (no programming assignment)
  - Two midterm exams (no final exam)

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.
- 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.
- V. Ganti, J. Gehrke, R. Ramakrishan. CACTUS Clustering Categorical Data Using Summaries. KDD'99.

References (2)

- 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.l 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.

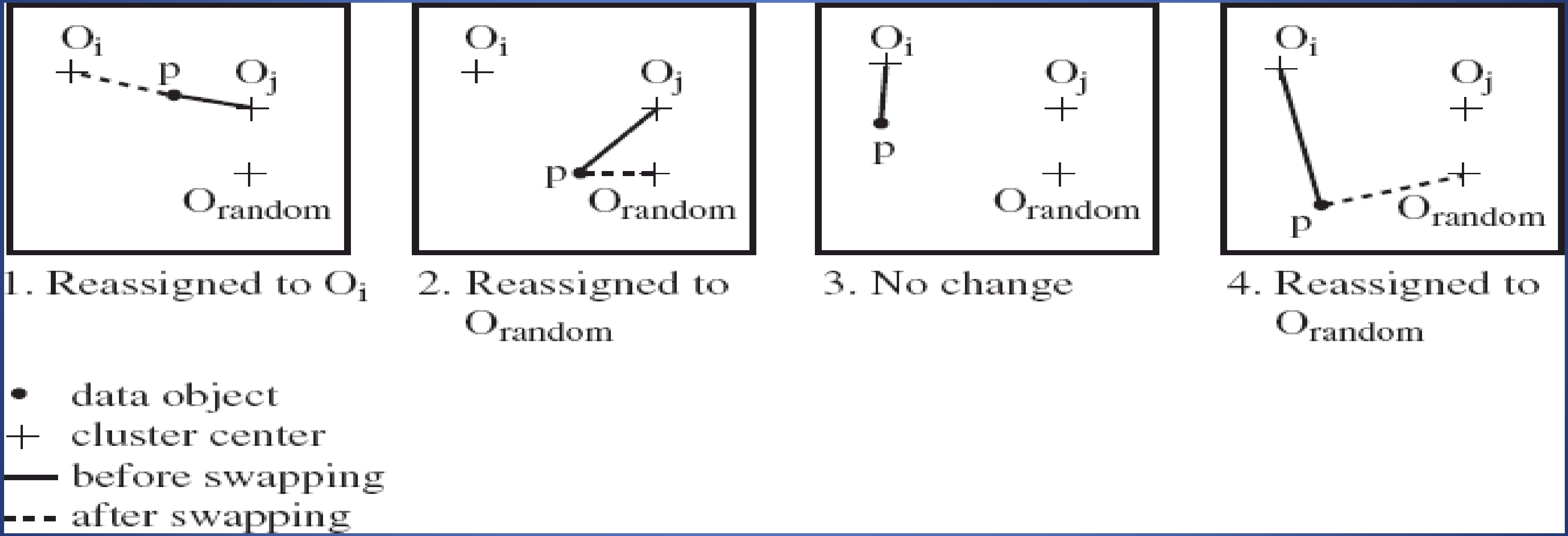
References (3)

- G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- 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
- X. Yin, J. Han, and P. S. Yu, “LinkClus: Efficient Clustering via Heterogeneous Semantic Links”, VLDB'06

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

- Case 1: p currently belongs to o_j. If o_j is replaced by o_{random} as a representative object and p is the closest to one of the other representative object o_i, then p is reassigned to o_i



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 Larg Applications) (1990)

- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as SPlus
  - 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)
  - 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, *it* starts with new randomly selected node in search for a new local optimum
- Advantages: More efficient and scalable than both *PAM* and *CLARA*
- Further improvement: Focusing techniques and spatial access structures (Ester et al.'95)

ROCK: Clustering Categorical Data

- ROCK: RObust Clustering using linKs
  - S. Guha, R. Rastogi & K. Shim, ICDE'99
- Major ideas
  - Use links to measure similarity/proximity
  - Not distance-based
- Algorithm: sampling-based clustering
  - Draw random sample
  - Cluster with links
  - Label data in disk
- Experiments
  - Congressional voting, mushroom data

Similarity Measure in ROCK

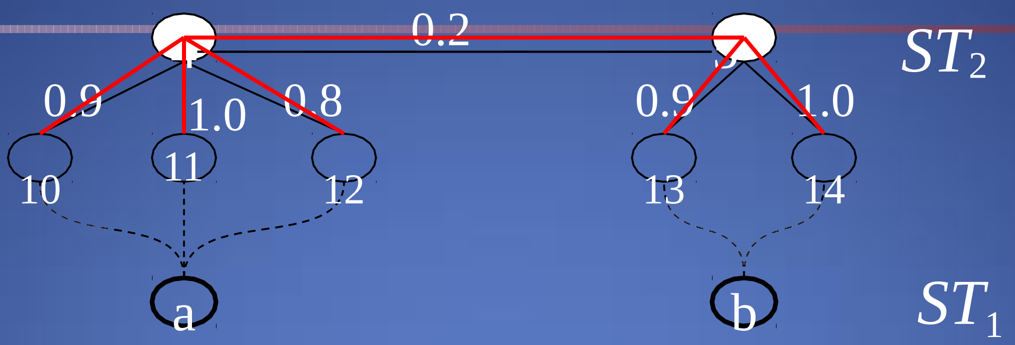
- Traditional measures for categorical data may not work well, e.g., Jaccard coefficient
- Example: Two groups (clusters) of transactions
  - C_1. <a, b, c, d, e>: {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}
  - C_2. <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}
- Jaccard co-efficient may lead to wrong clustering result
  - C_1: 0.2 ({a, b, c}, {b, d, e}} to 0.5 ({a, b, c}, {a, b, d})
  - C_1 & C_2: could be as high as 0.5 ({a, b, c}, {a, b, f})
- Jaccard co-efficient-based similarity function: $Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$
  - Ex. Let T_1= {a, b, c}, T_2= {c, d, e}

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

Link Measure in ROCK

- Clusters
  - C_1:<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}
  - C_2: <a, b, f, g>: {a, b, f}, {a, b, g}, {a, f, g}, {b, f, g}
- Neighbors
  - Two transactions are neighbors if sim(T_1,T_2) > threshold
  - Let T_1 = {a, b, c}, T_2 = {c, d, e}, T_3 = {a, b, f}
    - T_1 connected to: {a,b,d}, {a,b,e}, {a,c,d}, {a,c,e}, {b,c,d}, {b,c,e}, {a,b,f}, {a,b,g}
    - T_2 connected to: {a,c,d}, {a,c,e}, {a,d,e}, {b,c,e}, {b,d,e}, {b,c,d}
    - T_3 connected to: {a,b,c}, {a,b,d}, {a,b,e}, {a,b,g}, {a,f,g}, {b,f,g}
- Link Similarity
  - Link similarity between two transactions is the # of common neighbors
  - link(T_1, T_2) *= 4, since they have 4 common neighbors*
    - {a, c, d}, {a, c, e}, {b, c, d}, {b, c, e}
  - link(T_1, T_3) *= 3, since they have 3 common neighbors*
    - {a, b, d}, {a, b, e}, {a, b, g}

Aggregation-Based Similarity Computation



For each node $n_k \in \{n_{10}, n_{11}, n_{12}\}$ and $n_l \in \{n_{13}, n_{14}\}$, their path-based similarity $sim_p(n_k, n_l) = s(n_k, n_4) \cdot s(n_4, n_5) \cdot s(n_5, n_l)$.

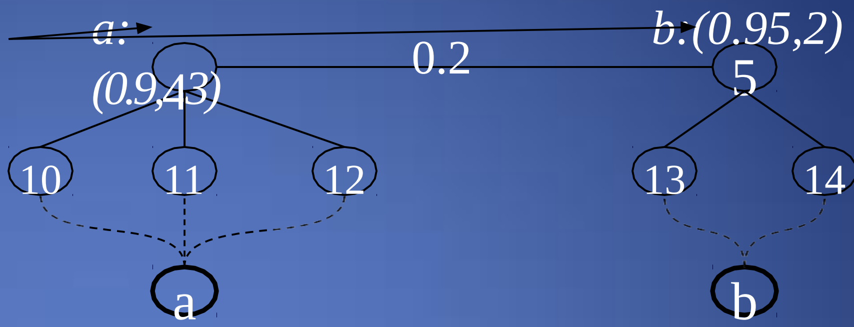
$$sim(n_a, n_b) = \frac{\sum_{k=10}^{12} s(n_k, n_4)}{3} \cdot s(n_4, n_5) \cdot \frac{\sum_{l=13}^{14} s(n_l, n_5)}{2} = 0.171$$

takes $O(3+2)$ time

After aggregation, we reduce quadratic time computation to linear time computation.

Computing Similarity with Aggregation

Average similarity and total weight



sim(n_a, n_b) can be computed from aggregated similarities

$$sim(n_a, n_b) = avg\_sim(n_a,n_4) \times s(n_4, n_5) \times avg\_sim(n_b,n_5)$$

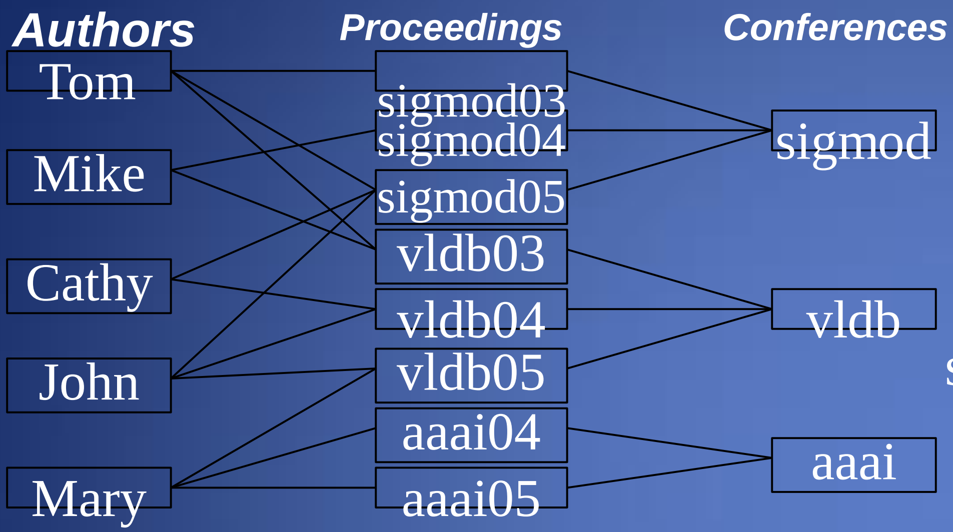
$$= 0.9 \times 0.2 \times 0.95 = 0.171$$

To compute *sim(n*$_a$*,n*$_b$*)*:

- Find all pairs of sibling nodes n_i and n_j, so that n_a linked with n_i and n_b with n_j.
- Calculate similarity (and weight) between n_a and n_b w.r.t. n_i and n_j.
- Calculate weighted average similarity between n_a and n_b w.r.t. all such pairs.

- Cluster Analysis: Basic Concepts
- Overview of Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Summary

Link-Based Clustering: Calculate Similarities Based On Links



Jeh & Widom, KDD’2002: *SimRank*

Two objects are similar if they are linked with the same or similar objects

- The similarity between two objects *x* and *y* is defined as the average similarity between objects linked with *x* and those with *y*:

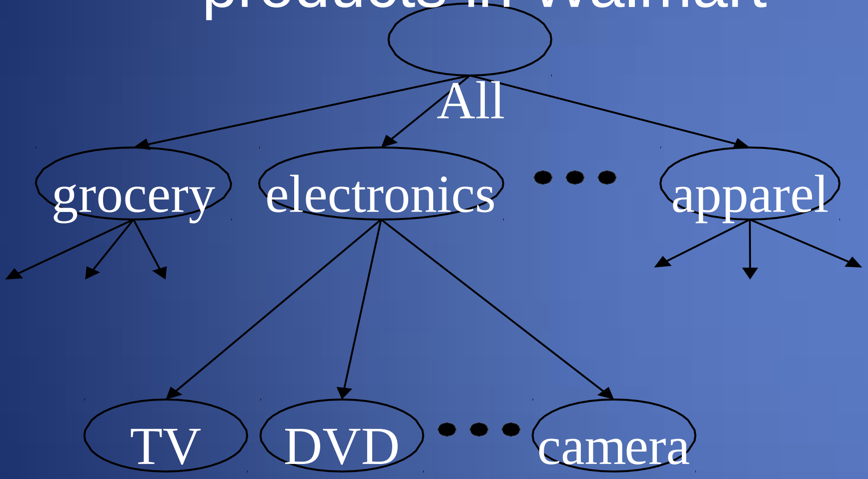
$$\mathrm{sim}(a,b)=\frac{C}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}\mathrm{sim}\left(I_i(a),I_j(b)\right)$$

- Issue: Expensive to compute:
    - For a dataset of *N* objects and *M* links, it takes *O*(*N*^{2}) space and *O*(*M*^{2}) time to compute all similarities.

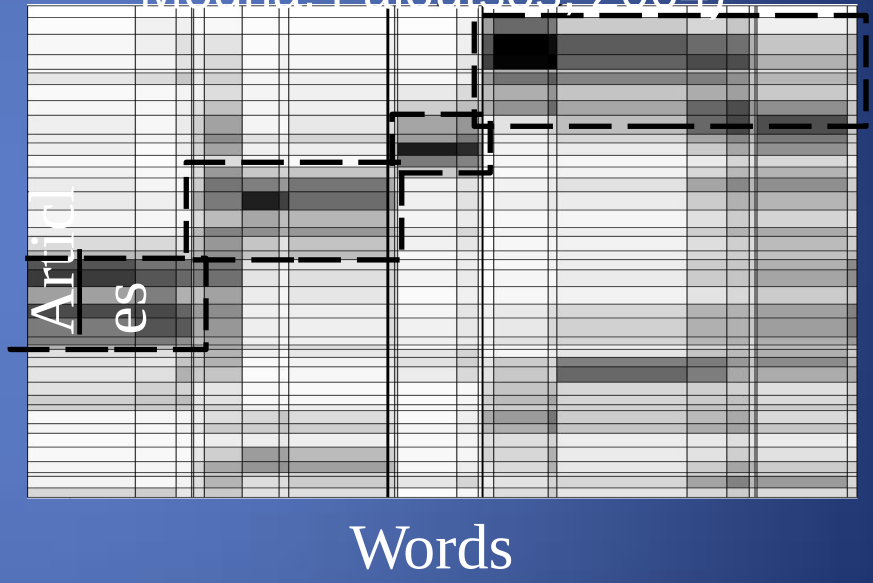
Observation 1: Hierarchical Structures

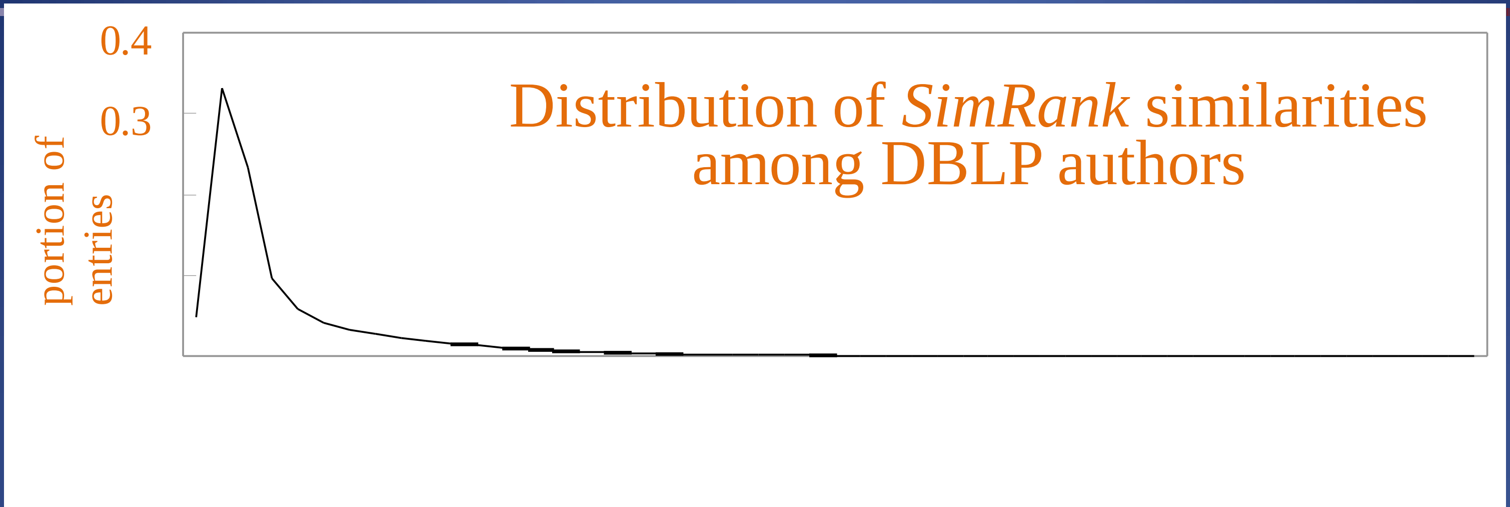
- Hierarchical structures often exist naturally among objects (e.g., taxonomy of animals)

A hierarchical structure of products in Walmart



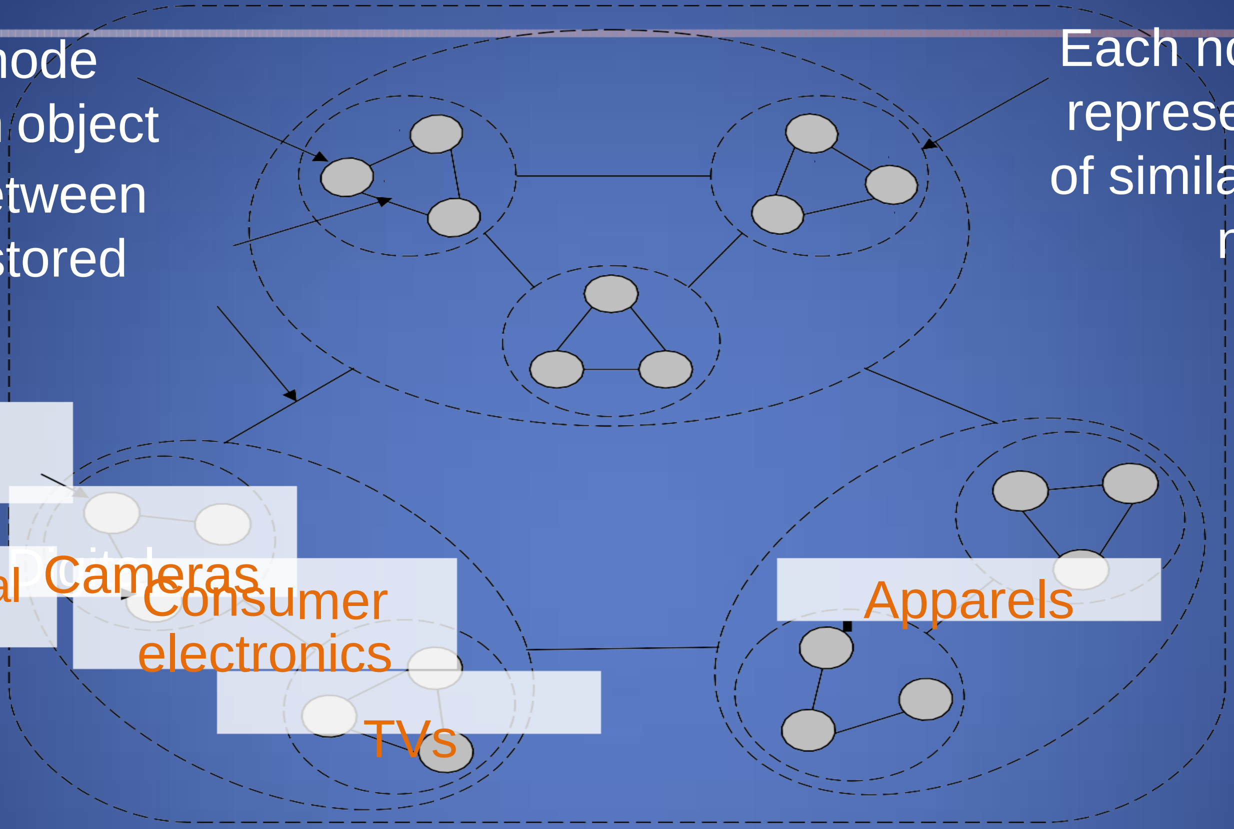
Relationships between articles and words (Chakrabarti, Papadimitriou, Modha, Faloutsos, 2004)





- Power law distribution exists in similarities
  - 56% of similarity entries are in [0.005, 0.015]
  - 1.4% of similarity entries are larger than 0.1
  - Can we design a data structure that stores the significant similarities and compresses insignificant ones?

A Novel Data Structure: SimTree



Each leaf node represents an object

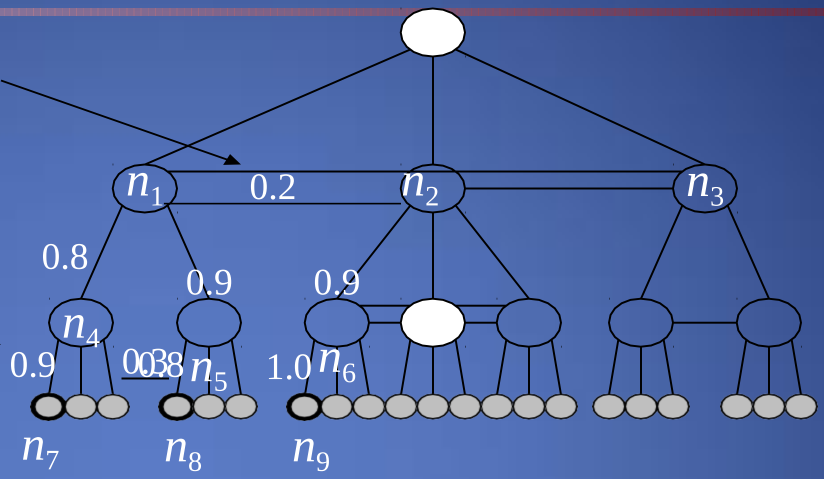
Similarities between siblings are stored

Each non-leaf node represents a group of similar lower-level nodes

Similarity Defined by SimTree

Similarity between two sibling nodes n_1 and n_2

Adjustment ratio for node n_7



- Path-based node similarity
  - $sim_p(n_7,n_8) = s(n_7, n_4) \text{ x } s(n_4, n_5) \text{ x } s(n_5, n_8)$
- Similarity between two nodes is the average similarity between objects linked with them in other SimTrees
- Adjust/ ratio for x = $\dfrac{\text{Average similarity between x and all other nodes}}{\text{Average similarity between x's parent and all other nodes}}$

LinkClus: Efficient Clustering via Heterogeneous Semantic Links

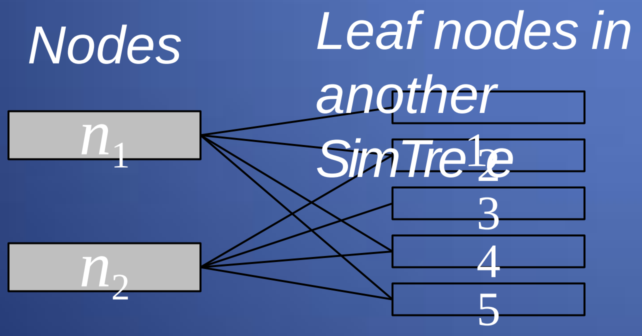
Method

- Initialize a SimTree for objects of each type
- Repeat until stable
  - For each SimTree, update the similarities between its nodes using similarities in other SimTrees
    - Similarity between two nodes *x* and *y* is the average similarity between objects linked with them
  - Adjust the structure of each SimTree
    - Assign each node to the parent node that it is most similar to

For details: X. Yin, J. Han, and P. S. Yu, “LinkClus: Efficient Clustering via Heterogeneous Semantic Links”, VLDB'06

Initialization of SimTrees

- Initializing a SimTree
  - Repeatedly find groups of tightly related nodes, which are merged into a higher-level node
- Tightness of a group of nodes
  - For a group of nodes $\{n_1, \ldots, n_k\}$, its tightness is defined as the number of leaf nodes in other SimTrees that are connected to all of $\{n_1, \ldots, n_k\}$



Nodes

Leaf nodes in another SimTree

n_1

n_2

1

2

3

4

5

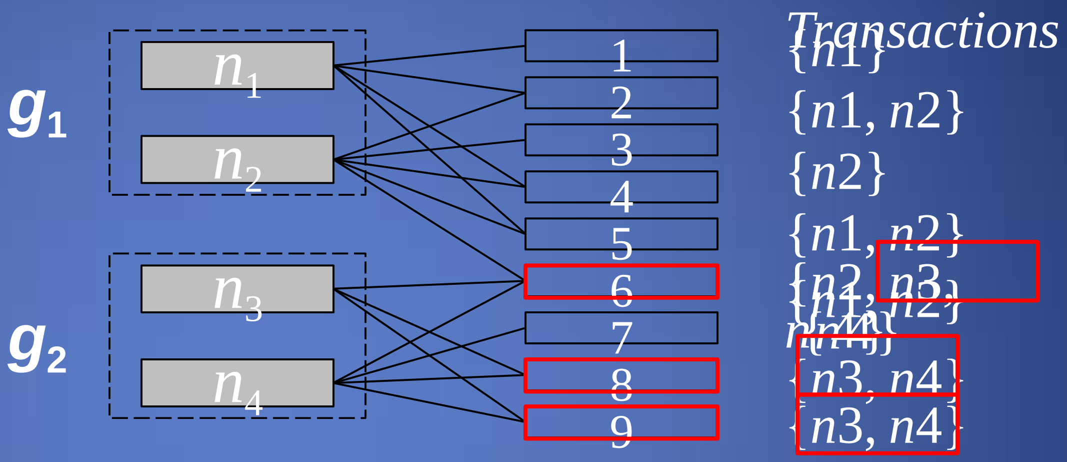
The tightness of $\{n_1, n_2\}$ is 3

Finding Tight Groups by Freq. Pattern Mining

- Finding tight groups → Frequent pattern mining

  Reduced to

The tightness of a group of nodes is the support of a frequent pattern



***g*₁** ***g*₂**

n_1 n_2 n_3 n_4

1 2 3 4 5 6 7 8 9

Transactions

{*n*1}

{*n*1, *n*2}

{*n*2}

{*n*1, *n*2}

{*n*2, *n*3,

{*n*1, *n*2}

*n*4}

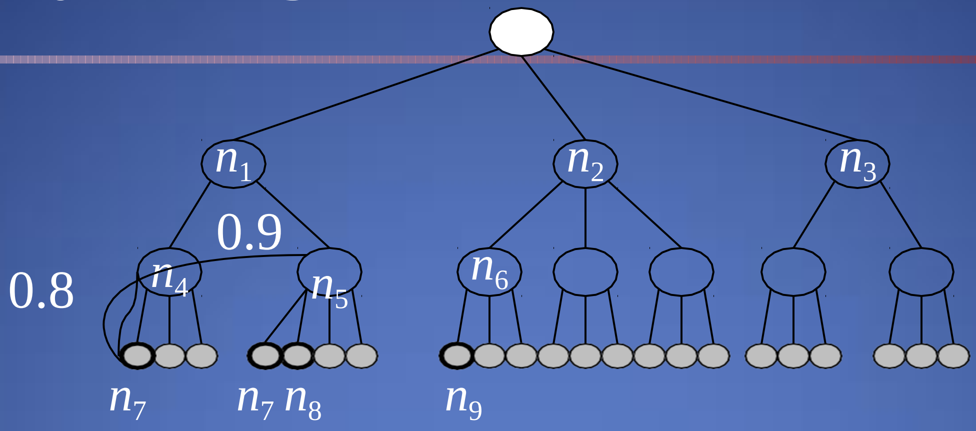
{*n*4}

{*n*3, *n*4}

{*n*3, *n*4}

- Procedure of initializing a tree
  - Start from leaf nodes (level-0)
  - At each level *l*, find non-overlapping groups of similar nodes with frequent pattern mining

Adjusting SimTree Structures



- After similarity changes, the tree structure also needs to be changed
  - If a node is more similar to its parent's sibling, then move it to be a child of that sibling
  - Try to move each node to its parent's sibling that it is most similar to, under the constraint that each parent node can have at most *c* children

Complexity

For two types of objects, N in each, and M linkages between them.

	Time	Space
Updating similarities	$O(M(\log N)^2)$	$O(M+N)$
Adjusting tree structures	$O(N)$	$O(N)$
LinkClus	$O(M(\log N)^2)$	$O(M+N)$
SimRank	$O(M^2)$	$O(N^2)$

Experiment: Email Dataset

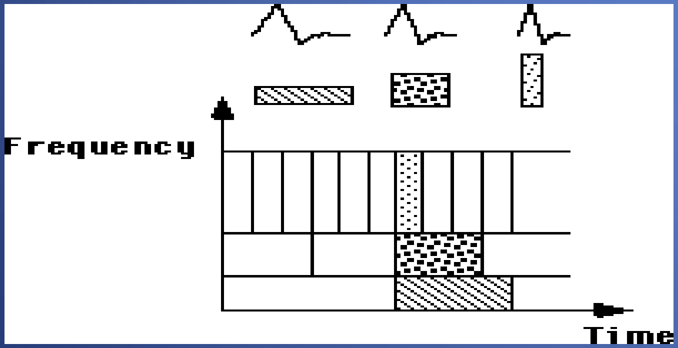
- F. Nielsen. Email dataset. www.imm.dtu.dk/~rem/data/Email-1431.zip
- 370 emails on conferences, 272 on jobs, and 789 spam emails
- Accuracy: measured by manually labeled data
- Accuracy of clustering: % of pairs of objects in the same cluster that share common label
- Approaches compared:
  - SimRank (Jeh & Widom, KDD 2002): Computing pair-wise similarities
  - SimRank with FingerPrints (F-SimRank): Fogaras & R´acz, WWW 2005
    - pre-computes a large sample of random paths from each object and uses samples of two objects to estimate SimRank similarity
  - ReCom (Wang et al. SIGIR 2003)

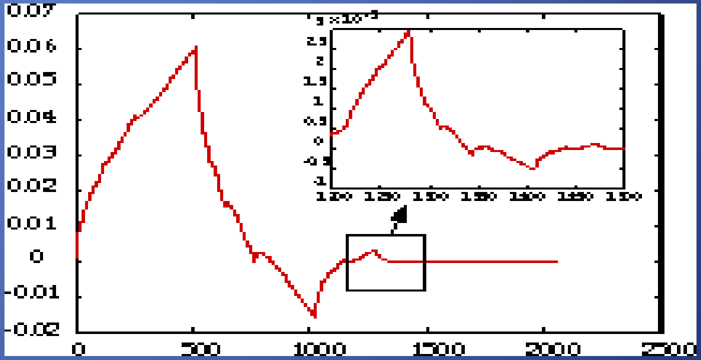
    Iteratively clustering objects using cluster labels of linked objects

Approach	*Accuracy*	*time (s)*
LinkClus	0.8026	1579.6
SimRank	0.7965	39160
ReCom	0.5711	74.6
F-SimRank	0.3688	479.7
CLARANS	0.4768	8.55

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; both grid-based and density-based
- Wavelet transform: A signal processing technique that decomposes a signal into different frequency sub-band
  - Data are transformed to preserve relative distance between objects at different levels of resolution
  - Allows natural clusters to become more distinguishable





The WaveCluster Algorithm

- 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
- 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

Quantization & Transformation

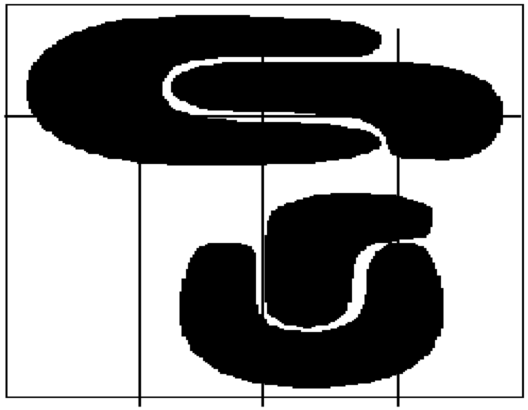


Figure 1: A sample 2-dimensional feature space.

- 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