Data Mining and Knowledge Discovery in Databases (KDD)

Mohammed J. Zaki

Computer Science Department

Rensselaer Polytechnic Institute

Tutorial overview

- Overview of KDD and data mining
- Data mining techniques
- Data pre-processing
- Data mining in bioinformatics
- Summary

Overview of data mining

- What is KDD?
- Why is KDD necessary
- The KDD process
- KDD operations and methods
- Core issues in KDD

What is data mining?

The iterative and interactive process of discovering valid, novel, useful, and understandable patterns or models in

Massive databases

What is data mining?

- Valid: generalize to the future
- Novel: what we don't know
- Useful: be able to take some action
- Understandable: leading to insight
- Iterative: takes multiple passes
- Interactive: human in the loop

Data mining goals

- Prediction
  - What?
  - Opaque
- Description
  - Why?
  - Transparent

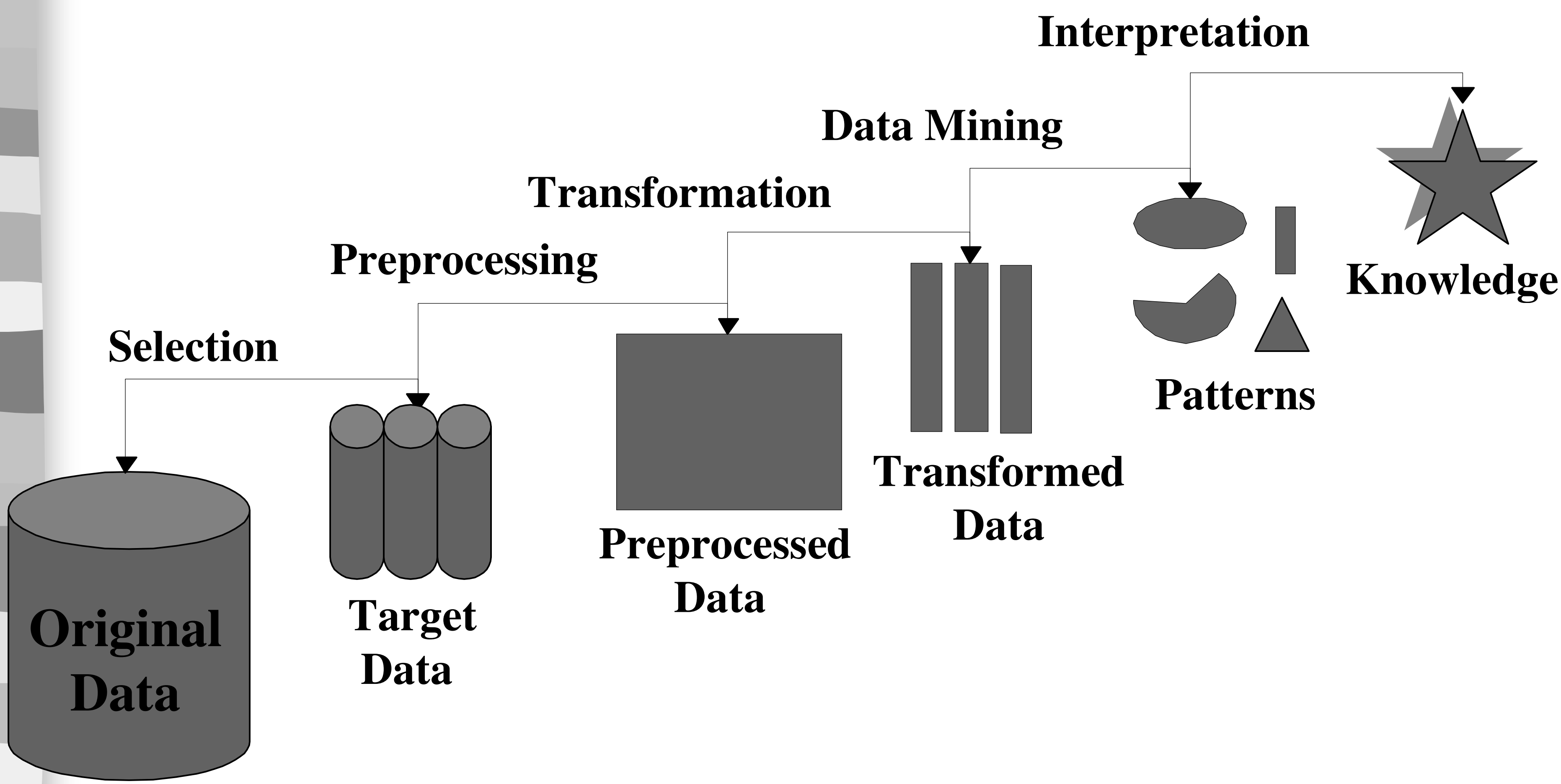
Data mining operations

- Verification driven
  - Validating hypothesis
  - Querying and reporting (spreadsheets, pivot tables)
  - Multidimensional analysis (dimensional summaries); On Line Analytical Processing
  - Statistical analysis

Data mining operations

- Discovery driven
  - Exploratory data analysis
  - Predictive modeling
  - Database segmentation
  - Link analysis
  - Deviation detection

Data mining process



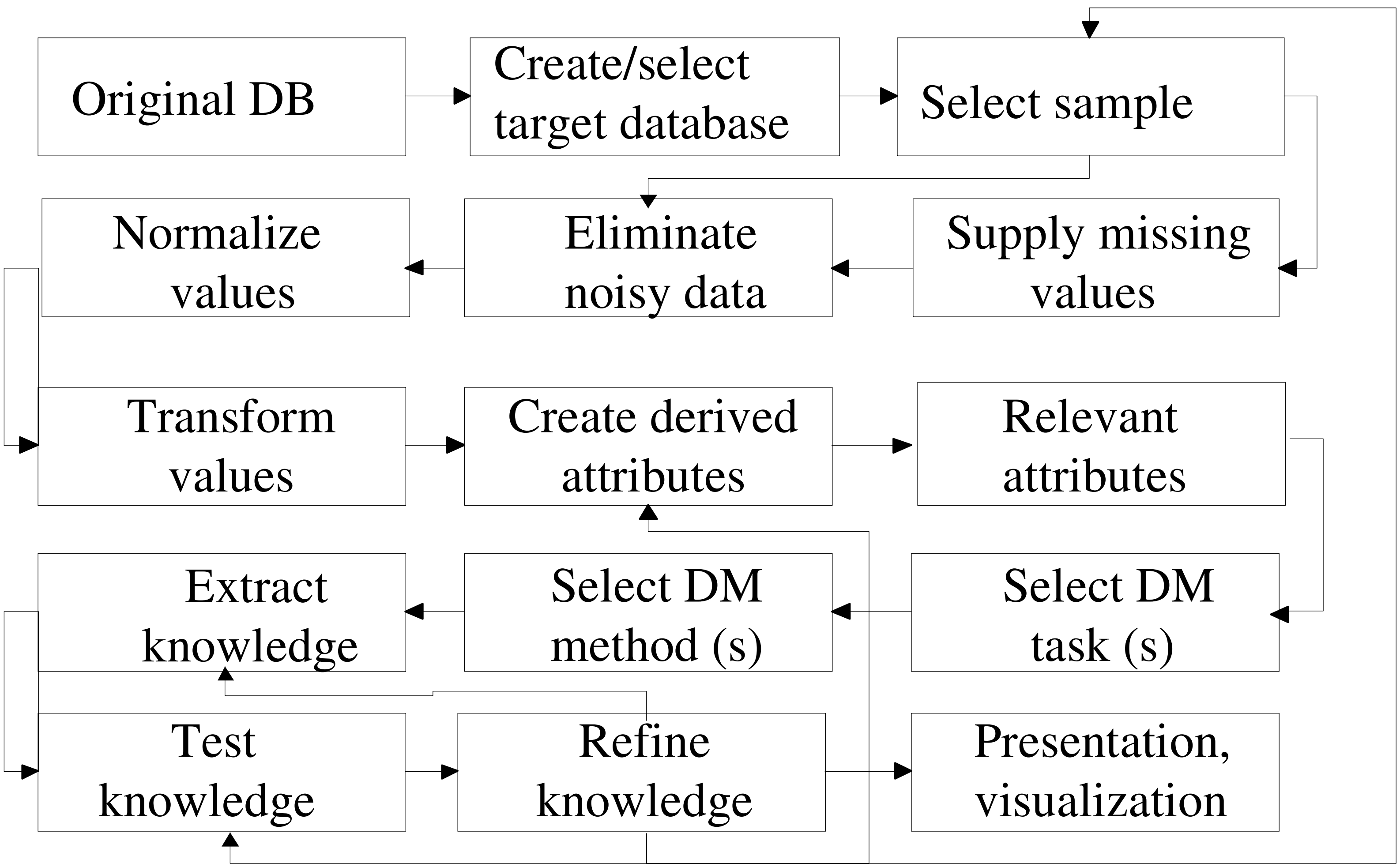
Data mining process

- Understand application domain
  - Prior knowledge, user goals
- Create target dataset
  - Select data, focus on subsets
- Data cleaning and transformation
  - Remove noise, outliers, missing values
  - Select features, reduce dimensions

Data mining process

- Apply data mining algorithm
  - Associations, sequences, classification, clustering, etc.
- Interpret, evaluate and visualize patterns
  - What's new and interesting?
  - Iterate if needed
- Manage discovered knowledge
  - Close the loop

Data mining process



Related fields

- Machine learning
- Statistics
- Databases and data warehousing
- High performance and parallel computing
- Visualization

Why data mining?

- Data volume too large for classical analysis
  – Number of records too large (millions or billions)
  – High dimensional (attributes/features/ fields) data (thousands)
- Increased opportunity for access
  – Web navigation, on-line collections

Need for data mining tools

- Human analysis breaks down with volume and dimensionality
  - How quickly can one digest 1 million records, with 100 attributes
  - High rate of growth, changing sources
- What is done by non-statisticians?
  - Select a few fields and fit simple models or attempt to visualize

Data mining methods

- Predictive modeling (classification, regression)
- Segmentation (clustering)
- Dependency modeling (graphical models, density estimation)
- Summarization (associations)
- Change and deviation detection

Data mining techniques

- Association rules: detect sets of attributes that frequently co-occur, and rules among them, e.g. 90% of the people who buy cookies, also buy milk (60% of all grocery shoppers buy both)
- Sequence mining (categorical): discover sequences of events that commonly occur together, .e.g. In a set of DNA sequences ACGTC is followed by GTCA after a gap of 9, with 30% probability

Data mining techniques

- Similarity search: given a database of objects, and a “query” object, find the object(s) that are within a user-defined distance of the queried object, or find all pairs within some distance of each other.
- Deviation detection: find the record(s) that is (are) the most different from the other records, i.e., find all outliers. These may be thrown away as noise or may be the “interesting” ones.

Data mining techniques

- Classification and regression: assign a new data record to one of several predefined categories or classes. Regression deals with predicting real-valued fields. Also called supervised learning.
- Clustering: partition the dataset into subsets or groups such that elements of a group share a common set of properties, with high within group similarity and small inter-group similarity. Also called unsupervised learning.

Data mining techniques

- Many other methods, such as
  - Decision trees
  - Neural networks
  - Genetic algorithms
  - Hidden markov models
  - Time series
  - Bayesian networks
  - Soft computing: rough and fuzzy sets

Research challenges for KDD

- Scalability
  - Efficient and sufficient sampling
  - In-memory vs. disk-based processing
  - High performance computing
- Automation
  - Ease of use
  - Using prior knowledge

Types of data mining tasks

- General descriptive knowledge
  - Summarizations
  - symbolic descriptions of subsets
- Discriminative knowledge
  - Distinguish between K classes
  - Accurate classification (also black box)
  - Separate spaces

Components of DM methods

- Representation: language for patterns/models, expressive power
- Evaluation: scoring methods for deciding what is a good fit of model to data
- Search: method for enumerating patterns/models

Tutorial overview

- Overview of KDD and data mining
- Data mining techniques
- Data pre-processing
- Data mining in bioinformatics
- Summary

Data mining techniques

- Association rules
- Sequence mining
- Classification
- Clustering
- Deviation detection
- K-nearest neighbors

What is association mining?

- Given a set of items/attributes, and a set of objects containing a subset of the items
- Find rules: if I1 then I2 (sup, conf)
- I1, I2 are sets of items
- I1, I2 have sufficient support: P(I1+I2)
- Rule has sufficient confidence: P(I2|I1)

Association mining

- User specifies "interestingness"
  - Minimum support (minsup)
  - Minimum confidence (minconf)
- Find all frequent itemsets (> minsup)
  - Exponential Search Space
  - Computation and I/O Intensive
- Generate strong rules (> minconf)
  - Relatively cheap

Example

OID	Items
1	A C T W
2	C D W
3	A C T W
4	A C D W
5	A C D T W
6	C D T

Support	Itemsets
100%(6)	
83% (5)	C, CW
67% (4)	A, D, T, AC, AW, CD, CT, ACW
50% (3)	AT, DW, TW, ACT, ATW, CDW, CTW, ACTW

Maximal Frequent Itemsets

ACTW, CDW

Association mining algorithm

- First find frequent sets of size 1, F(1)
- i=2
- C(i) = candidates of size i: sets of size i that could be frequent, given F(i-1)
- F(i) = sets from C(i) that are frequent (count directly from database)
- Apriori trick: a set is frequent only if all its subset are frequent

Algorithm illustration

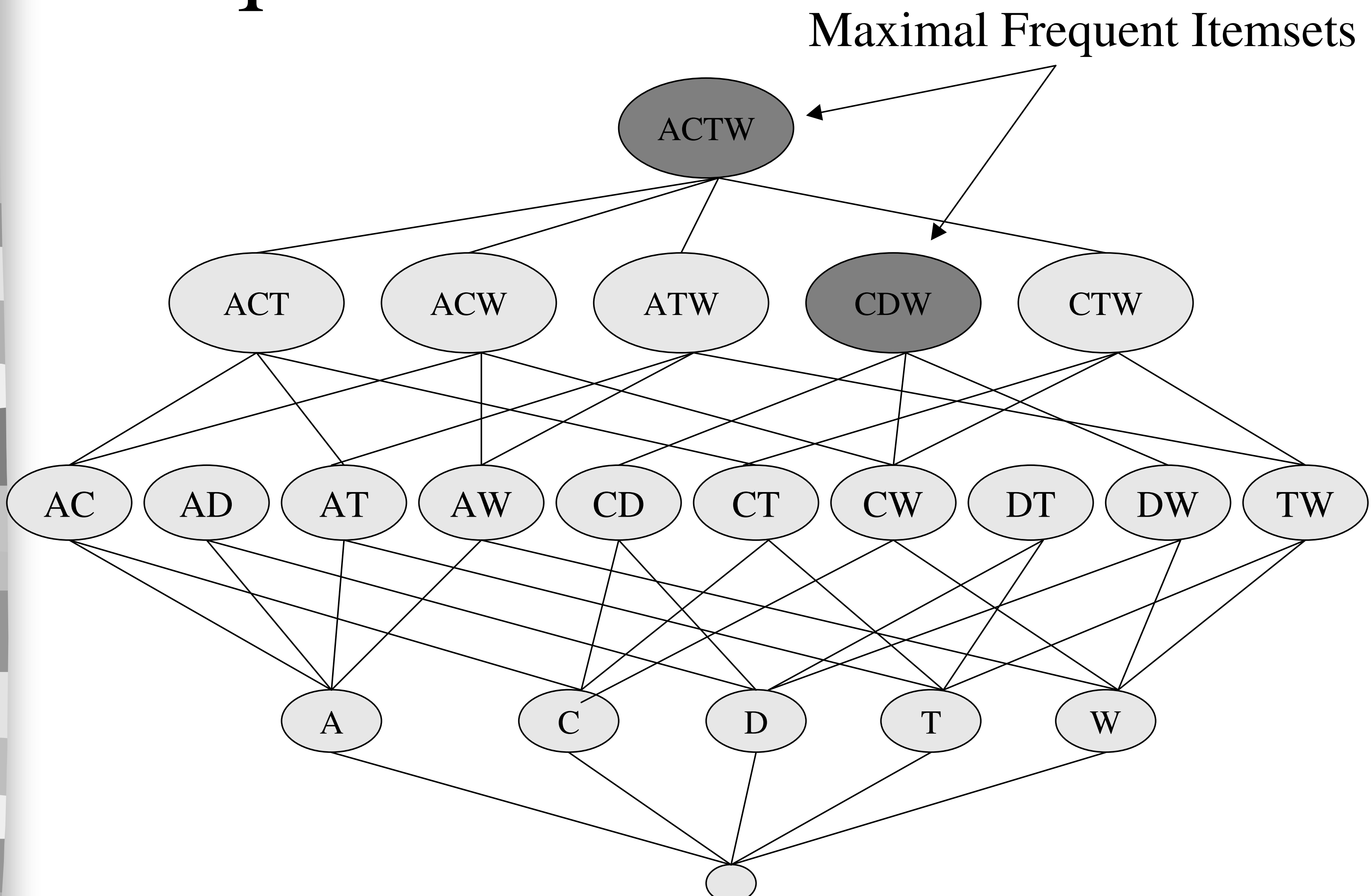
Size	Itemsets
1	A C D T W
2	AC AD AT AW CD CT CW DT DW TW
3	ACT ACW ATW CDW CTW
4	ACTW

OID	Scan 1	Scan 2	Scan 3	Scan 4
1	A C T W	AC AT AW CT CW TW	ACT ACW ATW CTW	ACTW
2	C D W	CD CW DW	CDW	
3	A C T W	AC AT AW CT CW TW	ACT ACW ATW CTW	ACTW
4	A C D W	AC AD AW CD CW DW	ACW CDW CTW	
5	A C D T W	AC AD AT AW CD CT CW DT DW TW	ACT ACW ATW CDW CTW	ACTW
6	C D T	CD CT DT		

Algorithm efficiency

- K = size of largest frequent set
- At most K passes through the data
- In each pass count C(i)
- Fast provided
  – K is bounded (usually in practice)
  – Frequent patterns are sparse
  – Worst case exponential

Frequent itemset lattice



Frequent sets to strong rules

- Given minsup, minconf
- Find all frequent sets, e.g. X
  - sup(X) > minsup
- For any frequent set, X
  - Generate a subset A of X
  - Let B = X-A
  - Generate rule $A \Rightarrow B$
    - provided P(B|A) = sup(X)/sup(A) > minconf

Example: sets to rules

- Maximal set: CDW. Subsets: CDW (3), CD (4), CW (5), DW (3), C (6), D(4), W (5)
- CD $\Rightarrow$ W, conf = 3/4 = 75%
- CW $\Rightarrow$ D, conf = 3/5 = 60%
- DW $\Rightarrow$ C, conf = 3/3 = 100%
- C $\Rightarrow$ DW, conf = 3/6 = 50%
- D $\Rightarrow$ CW, conf = 3/4 = 75%
- W $\Rightarrow$ CD, conf = 3/5 = 60%

What is sequence mining?

- Given a set of items, list of events per sequence ordered in time
- Find rules: if S1 then S2 (sup, conf)
- S1, S2 are sequences of items
- S1, S2 have sufficient support: P(S1+S2)
- Rule has sufficient confidence: P(S2|S1)

Sequence mining

- User specifies "interestingness"
  – Minimum support (minsup)
  – Minimum confidence (minconf)
- Find all frequent sequences (> minsup)
  – Exponential Search Space
  – Computation and I/O Intensive
- Generate strong rules (> minconf)
  – Relatively cheap

Example

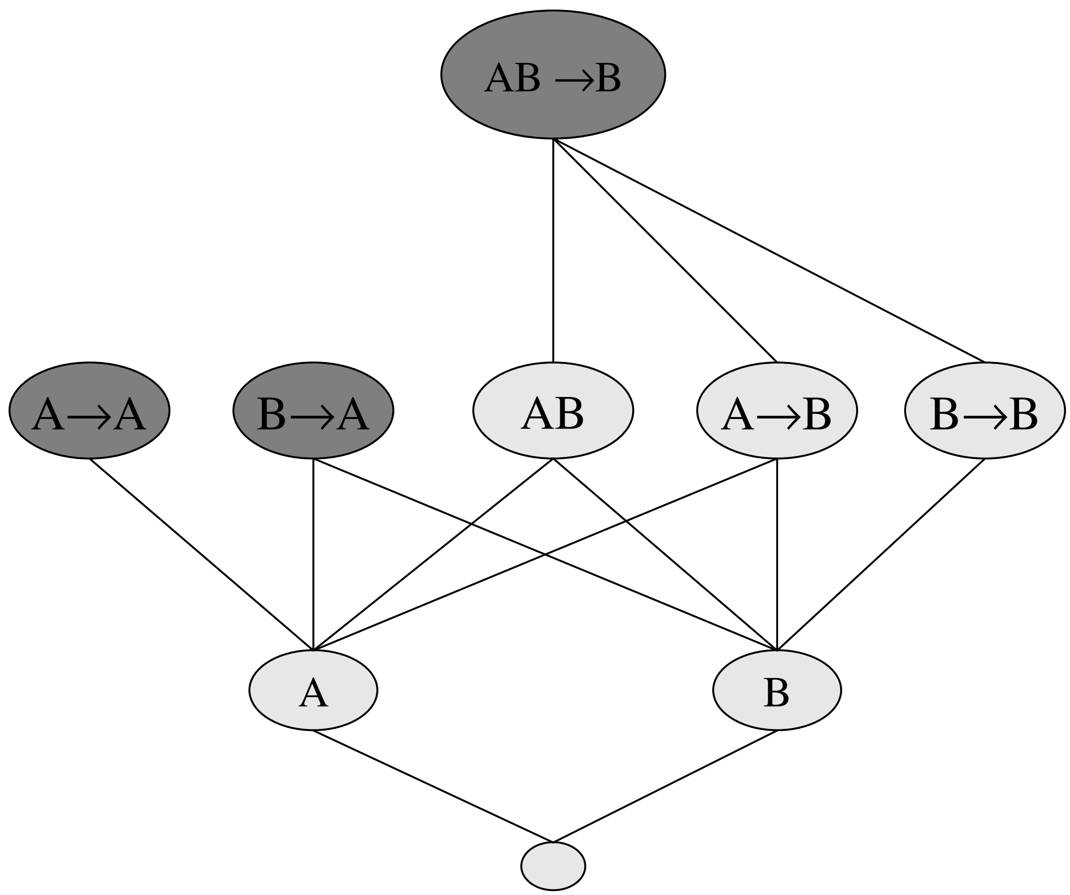
CID	Time	Items
1	10	A B
1	20	B
1	30	A B
2	20	A C
2	30	A B C
2	50	B
3	10	A
3	30	B
3	50	A
4	30	A B
4	40	A
4	50	B

Support	Sequences
100%	A B A→A A→B
75%	AB B→A B→B AB→B

Maximal Frequent Sequences

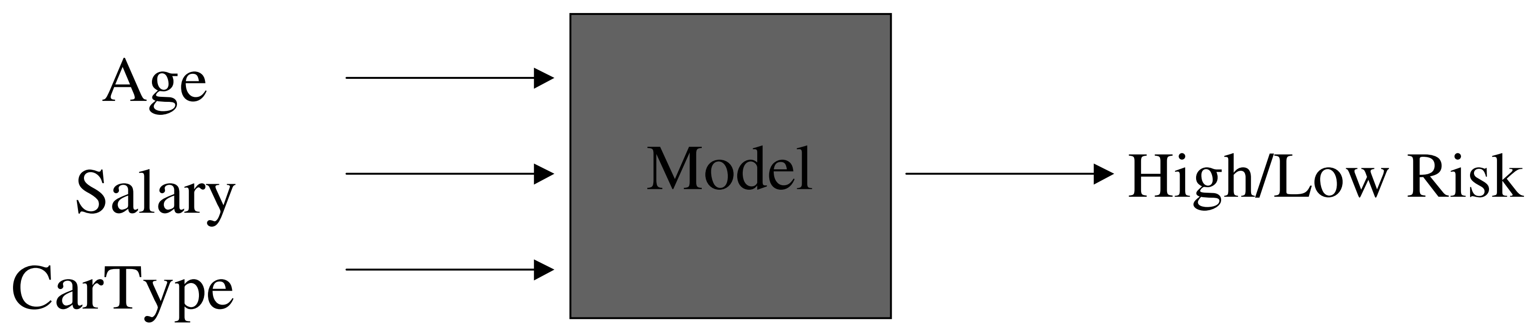
A→A B→A AB→B

Frequent sequence lattice



Predictive modeling

- A "black box" that makes predictions about the future based on information from the past and present



- Large number of input available

Models

- Some models are better than others
  - Accuracy
  - Understandability
- Models range from easy to understand to incomprehensible
  - Decision trees
  - Rule induction
  - Regression models
  - Neural networks

Easier

Harder

What is Classification?

Classification is the process of assigning new objects to predefined categories or classes

- Given a set of labeled records
- Build a model (decision tree)
- Predict labels for future unlabeled records

Classification learning

- Supervised learning (labels known)
- Example described in terms of attributes
  - Categorical (unordered symbolic values)
  - Numeric (integers, reals)
- Class (output/predicted attribute): categorical for classification, numeric for regression

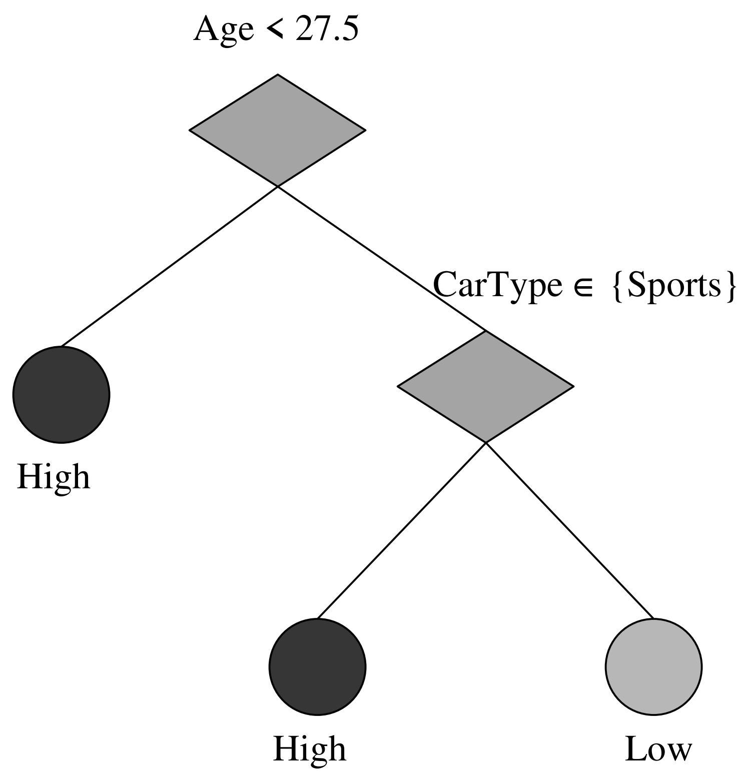
Classification learning

- Training set: set of examples, where each example is a feature vector (i.e., a set of (attribute,value) pairs) with its associated class. Model built on this set.
- Test set: a set of examples disjoint from the training set, used for testing the accuracy of a model.

Decision-tree classification

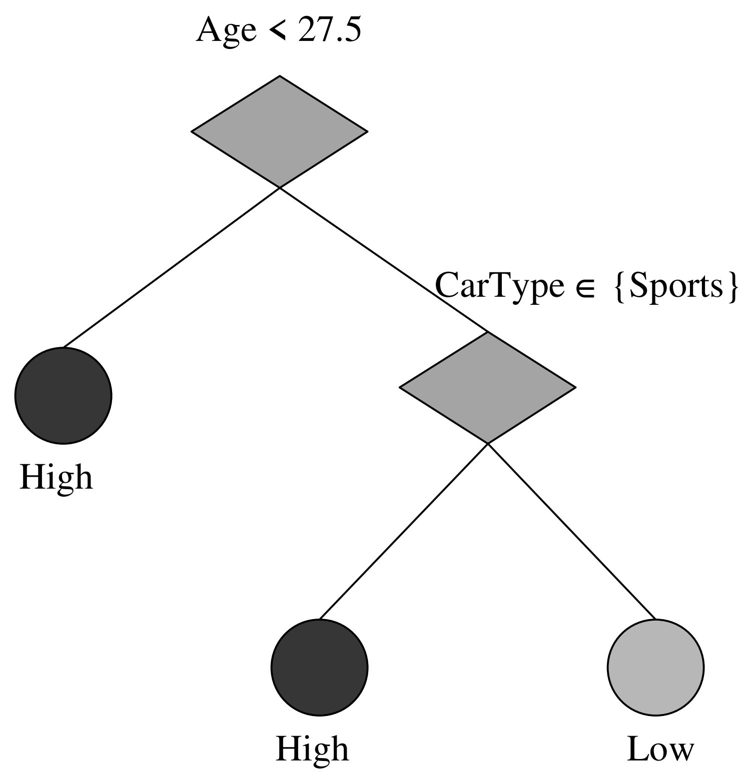
Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

Numeric Categorical



Age=40, CarType=Family $\Rightarrow$ Class=Low

From tree to rules



1) Age $<$ 27.5 $\Rightarrow$ High

2) Age $>=$ 27.5 and
CarType = Sports $\Rightarrow$ High

3) Age $>=$ 27.5 and
CarType $\neq$ Sports $\Rightarrow$ High

Classification algorithm

- Build tree
  - Start with data at root node
  - Select an attribute and formulate a logical test on attribute
  - Branch on each outcome of the test, and move subset of examples satisfying that outcome to corresponding child node

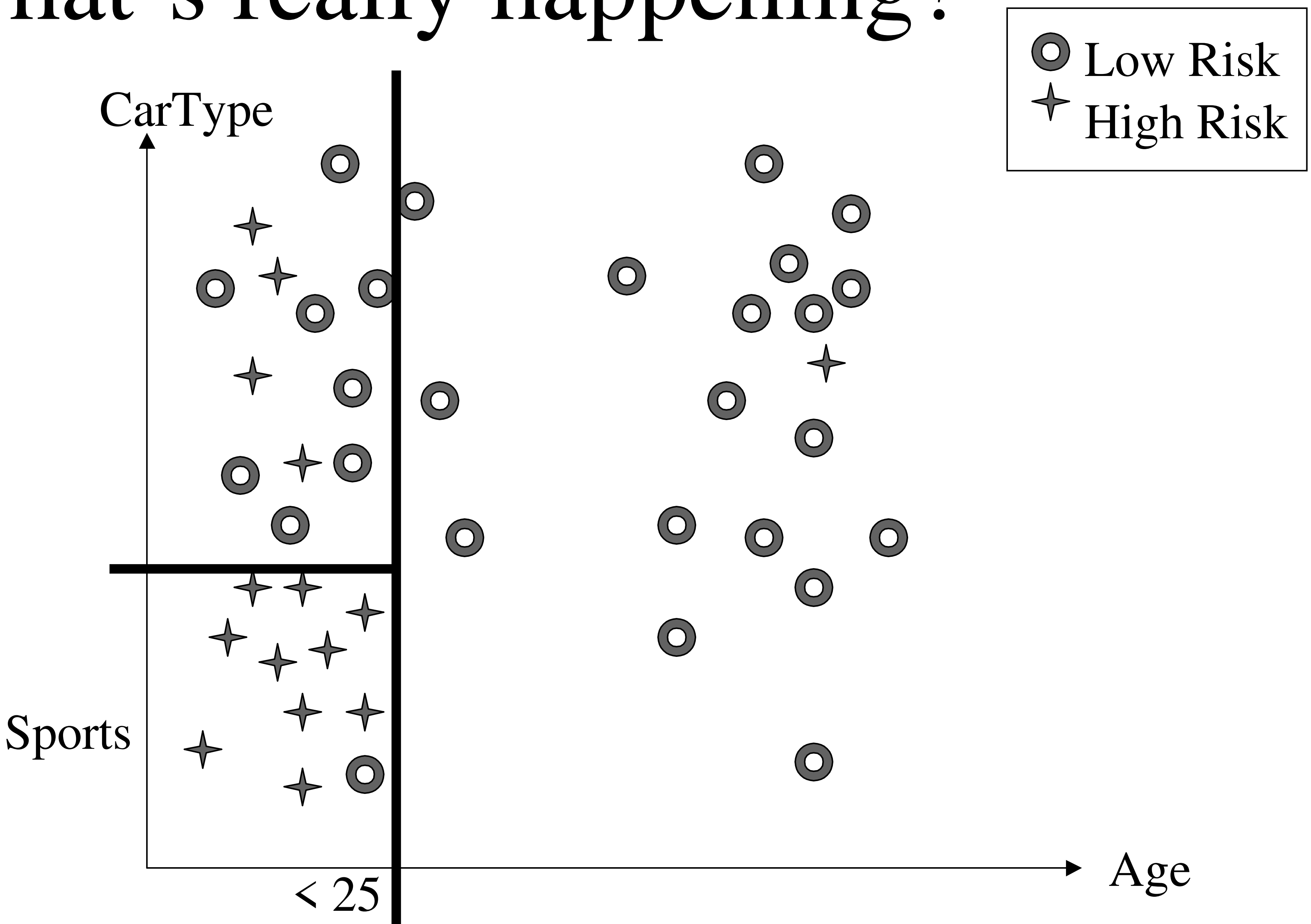
Classification algorithm

  - Recurse on each child node
  - Repeat until leaves are “pure”, i.e., have example from a single class, or “nearly pure”, i.e., majority of examples are from the same class
- Prune tree
  - Remove subtrees that do not improve classification accuracy
  - Avoid over-fitting, i.e., training set specific artifacts

Build tree

- Evaluate split-points for all attributes
- Select the “best” point and the “winning” attribute
- Split the data into two
- Breadth/depth-first construction
- CRITICAL STEPS:
  – Formulation of good split tests
  – Selection measure for attributes

What's really happening?



How to capture good splits?

- Occam’s razor: Prefer the simplest hypothesis that fits the data
- Minimum message/description length
  – Dataset D
  – Hypotheses H1, H2, …, Hx describing D
  – MML(Hi) = Mlength(Hi)+Mlength(D|Hi)
  – Pick Hk with minimum MML
- Mlength given by Gini index, Gain, etc.

Finding good split points

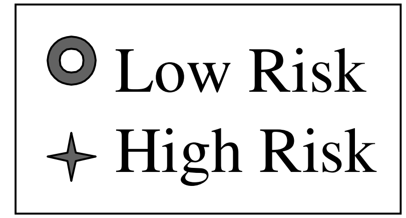
- Use Gini index for partition purity

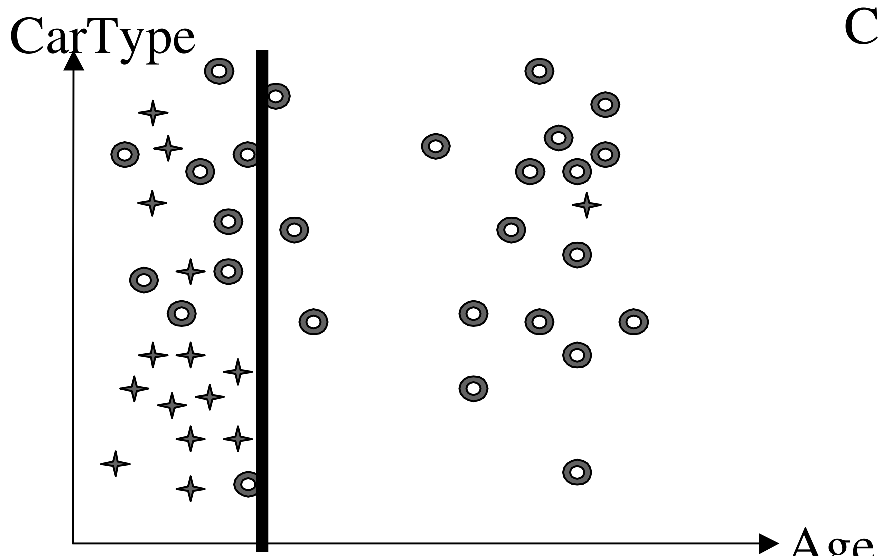
$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2$$

$$Gini(S_1, S_2) = \frac{n_1}{n} Gini(S_1) + \frac{n_2}{n} Gini(S_2)$$

- If S is pure, Gini(S) = 0
- Find split-point with minimum Gini
- Only need class distributions

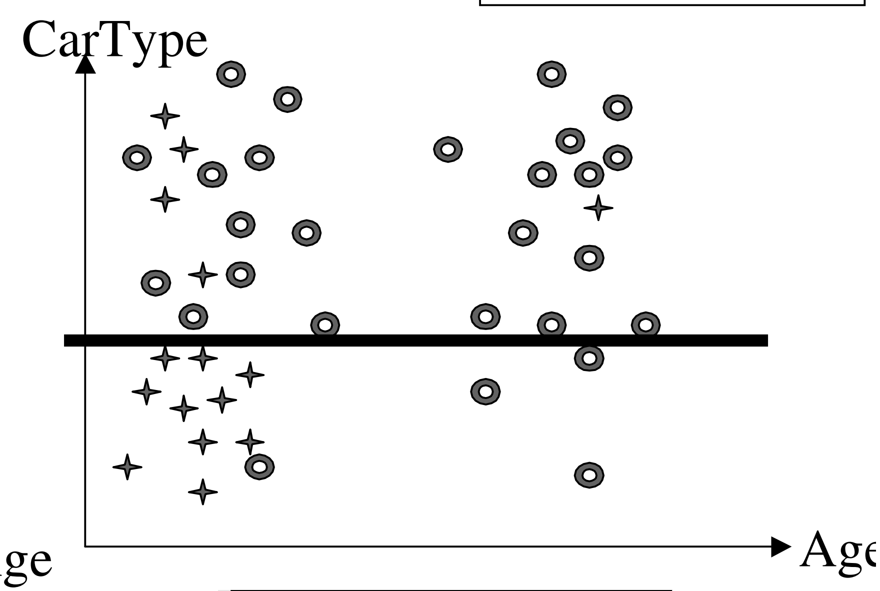
Finding good splits points





	High	Low
Left	14	9
Right	1	18

Gini(split) = 0.31



	High	Low
Top	5	23
Bottom	10	4

Gini(split) = 0.34

Tree pruning using MDL

- Data encoding: sum classification errors
- Model encoding:
  - Encode the tree structure
  - Encode the split points
- Pruning: choose smallest length option
  - Convert to leaf
  - Prune left or right child
  - Do nothing

What is clustering?

Given N k-dimensional feature vectors , find a “meaningful” partition of the N examples into c subsets or groups

- Discover the “labels” automatically
- c may be given, or “discovered”
- much more difficult than classification, since in the latter the groups are given, and we seek a compact description

Clustering

- Have to define some notion of "similarity" between examples
- Goal: maximize intra-cluster similarity and minimize inter-cluster similarity
- Feature vector be
  – All numeric (well defined distances)
  – All categorical or mixed (harder to define similarity; geometric notions don't work)

Clustering schemes

- Distance-based
  - Numeric
    - Euclidean distance (root of sum of squared differences along each dimension)
    - Angle between two vectors
  - Categorical
    - Number of common features (categorical)
- Partition-based
  - Enumerate partitions and score each

Clustering schemes

- Model-based
  - Estimate a density (e.g., a mixture of gaussians)
  - Go bump-hunting
  - Compute P(Feature Vector i | Cluster j)
  - Finds overlapping clusters too
  - Example: bayesian clustering

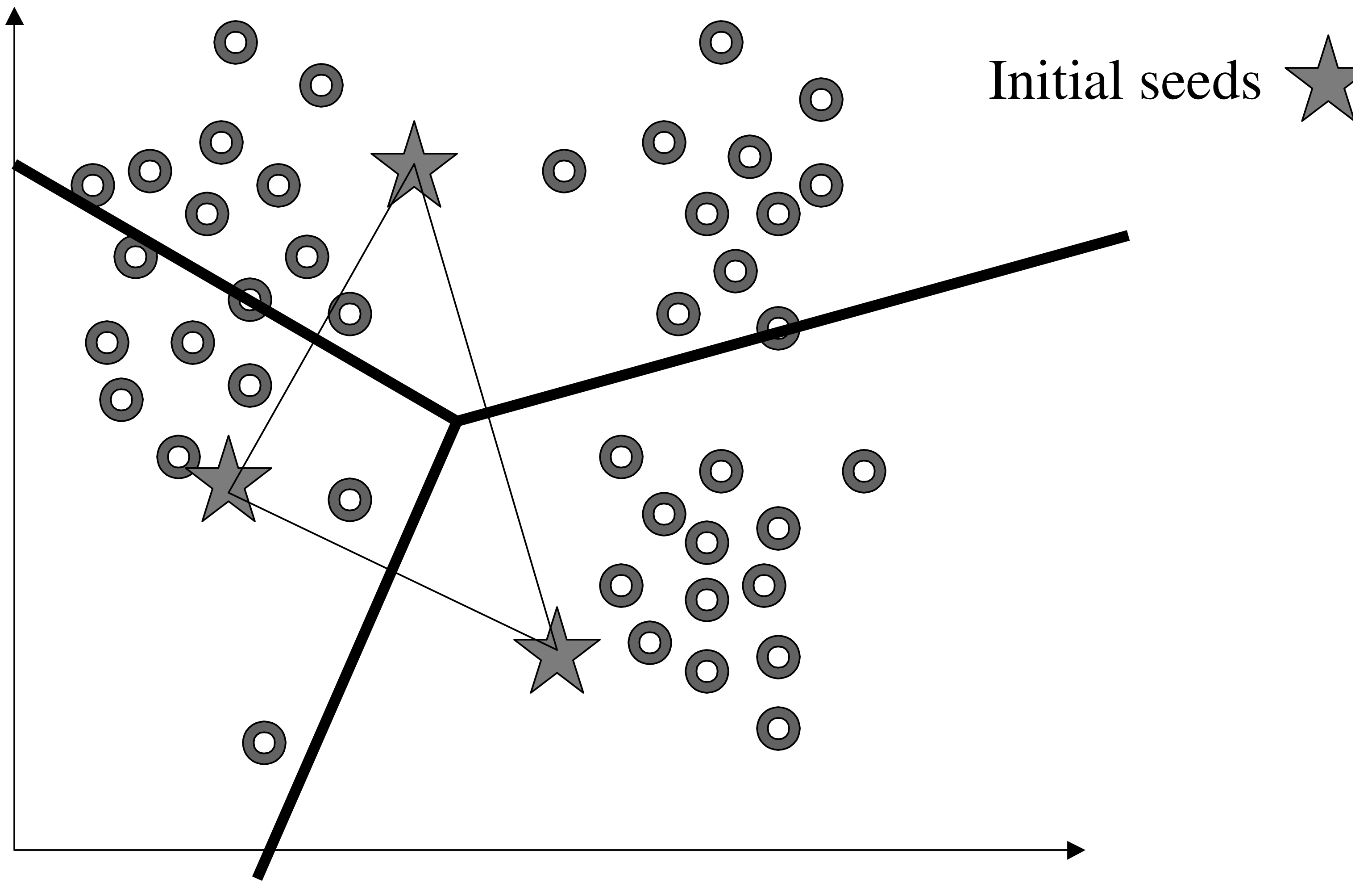
Before clustering

- Normalization:
  - Given three attributes
    - A -- micro-seconds
    - B -- milli-seconds
    - C -- seconds
  - Can't treat differences as the same in all dimensions or attributes
  - Need to scale or normalize for comparison
  - Can assign weight for more importance

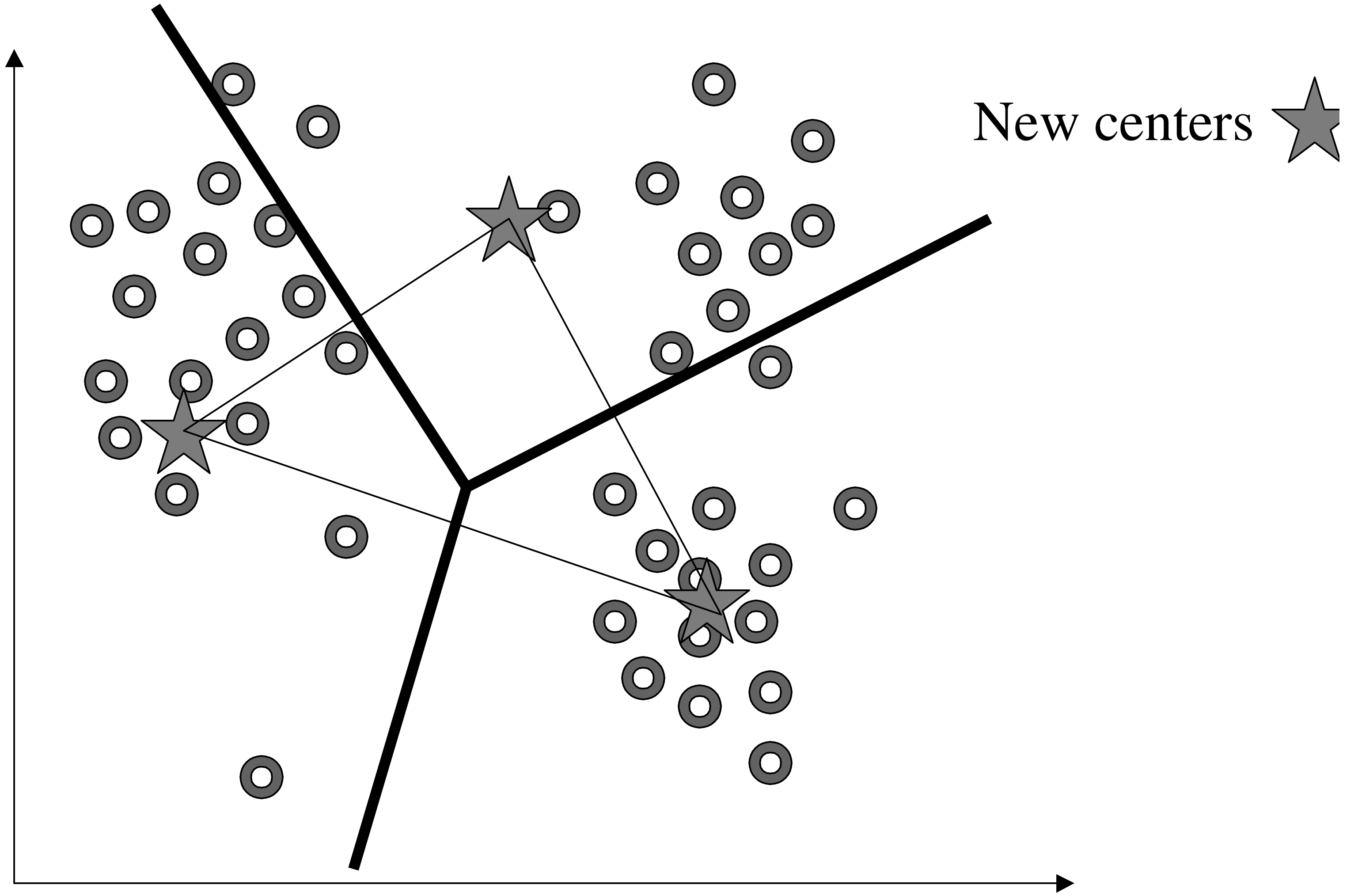
The k-means algorithm

- Specify ‘k’, the number of clusters
- Guess ‘k’ seed cluster centers
- 1) Look at each example and assign it to the center that is closest
- 2) Recalculate the center
- Iterate on steps 1 and 2 till centers converge or for a fixed number of times

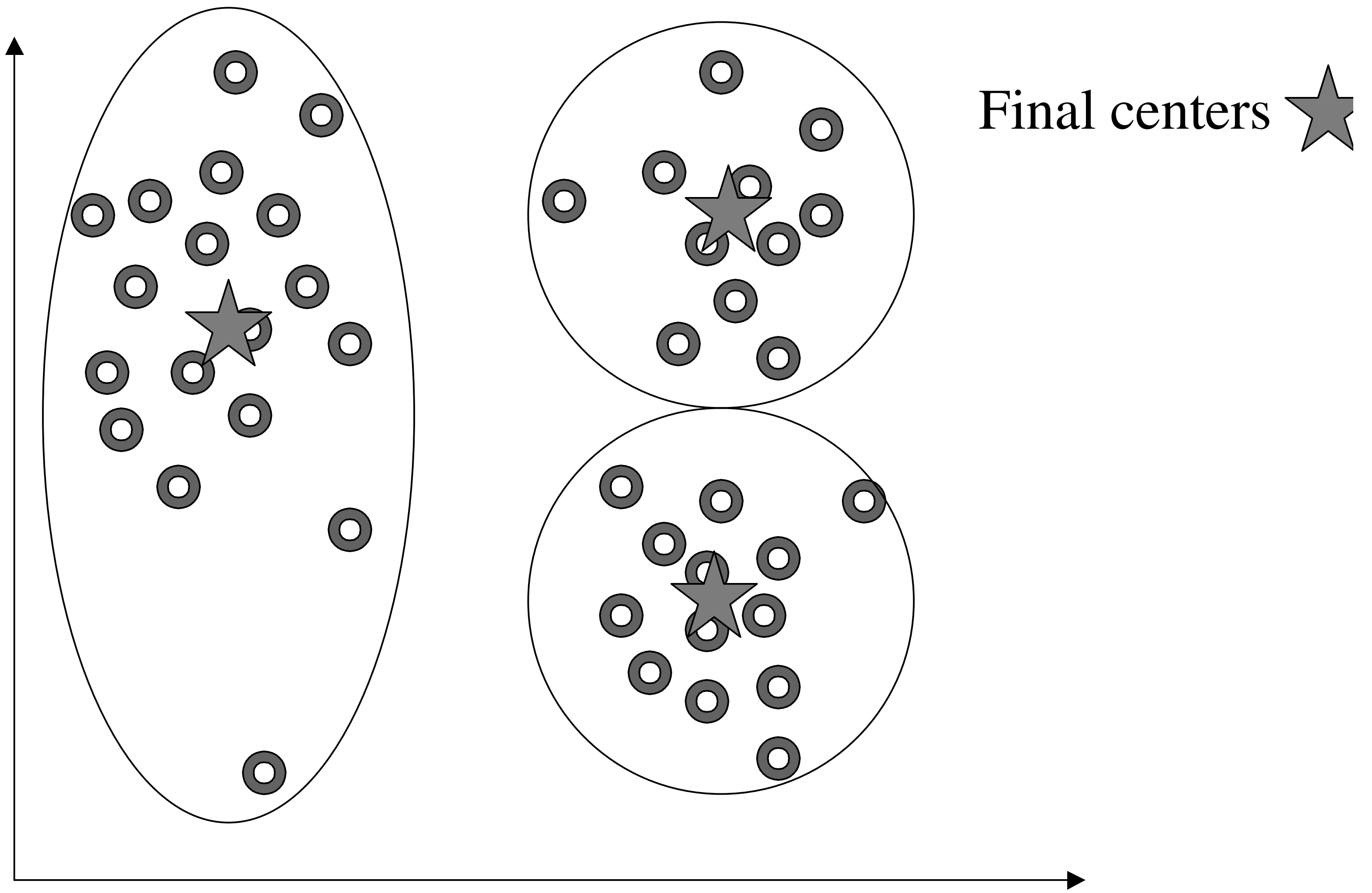
K-means algorithm



K-means algorithm



K-means algorithm



Gaussian mixture models

- Drawbacks of K-means
  - Doesn't do well with overlapping clusters
  - Clusters easliy pulled off center by outliers
  - Each records is either in or out of a cluster; no notion of some records begin more or less likely than others to really belong to cluster they have been assigned
- GMM: probabilistic variants of K-means

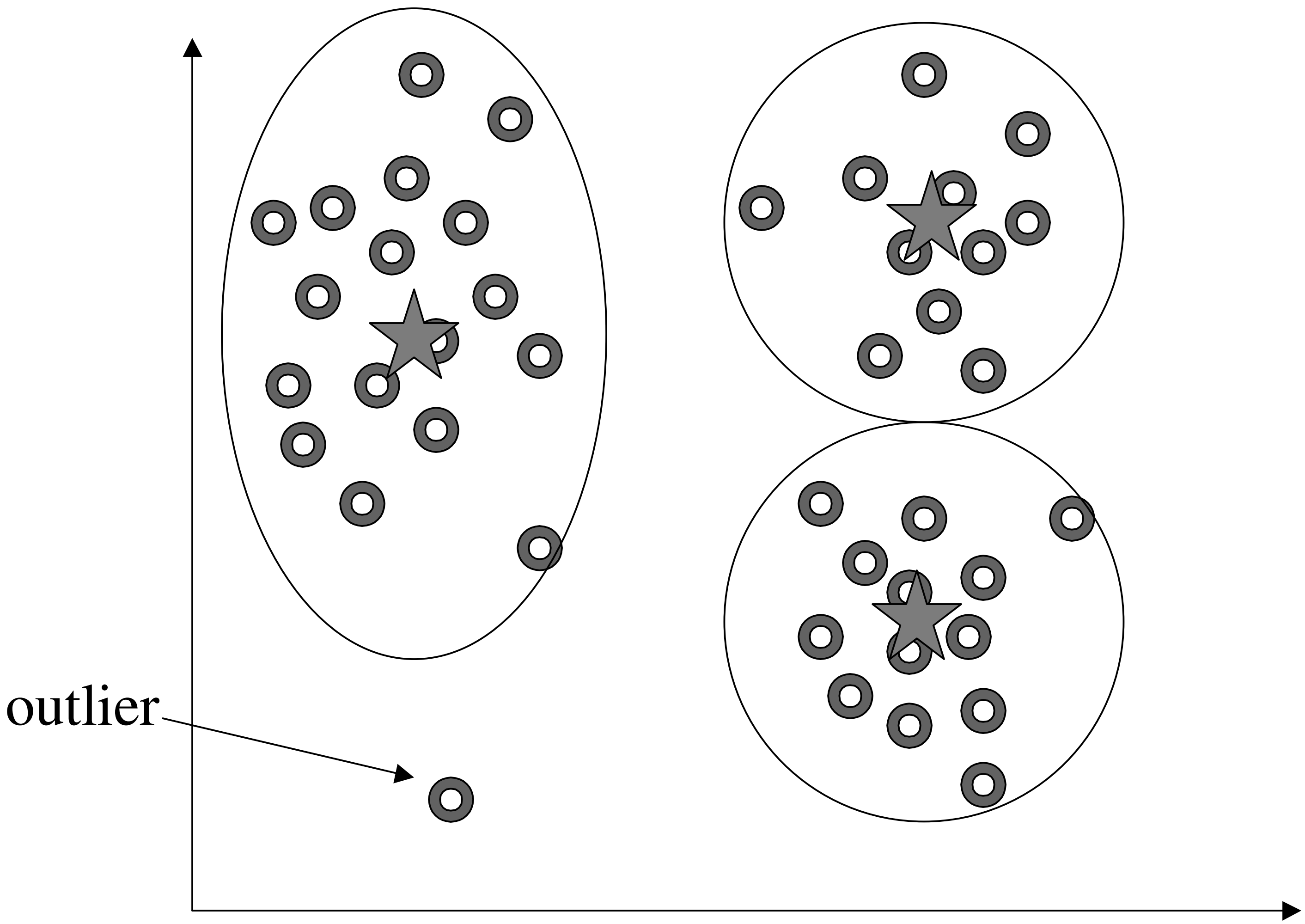
Estimation-maximization

- Choose K seeds: means of a gaussian distribution
- Estimation: calculate probability of belonging to a cluster based on distance
- Maximization: move mean of gaussian to centroid of data set, weighted by the contribution of each point
- Repeat till means don't move

Deviation/outlier detection

- Find points that are very different from the other points in the dataset
- Could be “noise”, that causes problems for classification or clustering
- Could be the really “interesting” points, for example, in fraud detection, we are mainly interested in finding the deviations from the norm

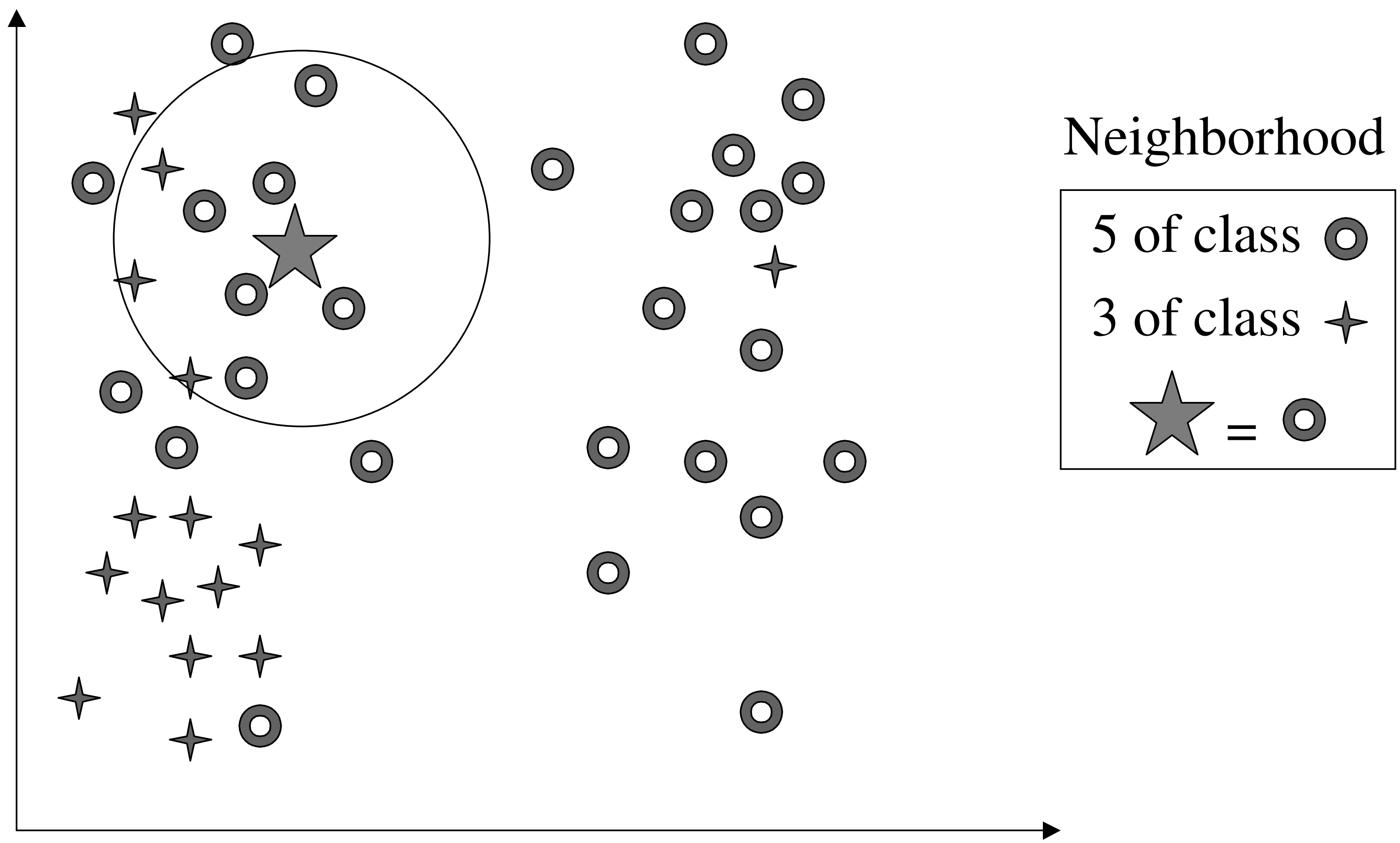
Deviation detection



K-nearest neighbors

- Classification technique to assign a class to a new example
- Find k-nearest neighbors, i.e., most similar points in the dataset (compare against all points!)
- Assign the new case to the same class to which most of its neighbors belong

K-nearest neighbors



Tutorial overview

- Overview of KDD and data mining
- Data mining techniques
- Data pre-processing
- Data mining in bioinformatics
- Summary

Data problems

- Noise
  - Recording errors and technology limitations
  - Uncertainty/probabilistic nature of attribute and class values
- Missing data
  - Conflicts with recorded data
  - Data not considered important at creation

Data problems

- Redundant data
  - Same characteristic has different names
  - Repeated records
  - Irrelevant or information-poor attributes
- Small and stale data
  - Rare events
  - Data not up to date

Data preprocessing

- Data cleaning (duplicate removal, remove inconsistencies, supply missing values)
- Selecting an appropriate data set and/or sampling strategy
- Dimensionality reduction and feature space selection
- Mapping of time-series (continuous) or sequence (categorical) data

Data/dimensionality reduction

- Why?
  - Time gains
  - Predictive power improves (larger data usually leads to more complex models)
- Issues:
  - Can data be discarded without sacrificing quality of results
  - Can this be done faster than running the main prediction algorithm itself

How to reduce data?

- Feature selection: remove irrelevant features
- Instance selection: remove examples
- Discretization: reduce the number of values of a feature
- Feature transformation: form new composite features (compression)

Dimensionality reduction

- Identify relevant attributes/features
- Reducing feature space is a search
  - Evaluation function for each feature
  - Enumerate possibilities
- In classification problems, can use estimated error rate using some fixed algorithm as evaluation function

Feature subset selection

- 2^d possible subsets of d features
- Best single features under feature independence assumption (significance)
- Best stepwise: single best, next conditional to the first, …
- Stepwise elimination: Start with all, eliminate worst, ….
- Optimal branch and bound

Principal component analysis

- Given N data records with k dimensions (works for numeric values)
- Find c << k N-dimensional vectors to best represent the data
  - Which c features are best
- Find n << N k-dimensional vectors
  - Used for data compression
  - Which n records are best

Handling time

- Most techniques cannot account for sequences in data
- Transforming to remove dependence between sequence cases
- Map times-series to static data; varies in complexity

Mapping time data

- Create attributes that code simple quantities on preceding n examples
- Transform from time to frequency domain, FFT
- Hidden Markov Models: infer “internal” state of process generating sequence data; state parameters become attribute values of “static” example

Tutorial overview

- Overview of KDD and data mining
- Data mining techniques
- Data pre-processing
- Data mining in bioinformatics
- Summary

Bioinformatics: Grand Challenges

- Protein structure prediction
  - primary structure (amino acid sequence)
  - secondary structure (alpha helix, beta sheets)
  - tertiary structure (3D conformation)
- Search space is Huge

Bioinformatics: Grand Challenges

- Homology search
  – Detect increasingly distant homologues, or proteins related by evolution from a common ancestor
  – Done via local alignment and similarity search algorithms
  – Challenges include developing faster methods (full local alignment is quadratic), and using more elaborate domain models

Bioinformatics: grand challenges

- Multiple alignment and phylogeny construction: mutual dependence
  – Phylogeny uses metrics of similarity derived from alignment of related multiple sequences
  – Multiple alignment are most effective when they account for degree of relatedness of sequences being aligned, i.e., when a phylogenetic tree is available

Bioinformatics: grand challenges

- Genomic sequence analysis and gene finding: Human Genome Project
  - Gene-finding and gene structure prediction from “raw” sequence data
  - Signals related to gene expression
  - Sheer volume of data poses the most serious challenge

The three problems

- There are no rules without exception
  - Exceptions to invariant dinucleotides found at intron boundaries
- Presence of non-local components
  - A syntactically correct gene is not a gene if it is not expressed (context dependent)
  - identical short protein stretches can be in different conformations
- Every problem is intertwined with others

Mining biosequences

- Group sequences thought to have common biological (structural, functional) properties -- semantic
- Study purely syntactic properties of these sequences -- patterns, clusters
- Test if discovered patterns make sense (back to semantic level)

Problem definitions

- Family characterization
  - Given example sequences from a family S of sequences (protein, DNA), find “interesting” properties characteristic to S
- Family classification
  - Given example sequences from two families S1 and S2, find properties distinguishing the two

DNA sequence mining

SID	Sequence
1	ACGTCACG
2	TCGA
3	GACTGCA
4	CAGTC
5	AGCT
6	TGCAGCTC
7	AGTCAG

- Maximal frequent sequences (sup>=4, size>=3)
  - ACT, CAG, GCA, TCA, TCG, AGTC
  - AGT $\Rightarrow$ C (4/5 = 80% confidence)

DNA sequence classification

- Mine features (frequent subsequences)
  - ACT, CAG, GCA, TCA, TCG, AGTC
- Convert data to boolean features

SID	Sequence	Class
1	ACGTCACG	C1
2	TCGA	C1
3	GACTGCA	C2
4	CAGTC	C2
5	AGCT	C1
6	TGCAGCTC	C2
7	AGTCAG	C1

SID	ACT	CAG	GCA	TCA	TCG	AGTC	Class
1	1	1	1	1	1	1	C1
2	0	0	0	1	1	0	C1
3	1	0	1	1	0	0	C2
4	0	1	0	0	0	1	C2
5	1	0	0	0	0	0	C1
6	1	1	1	1	1	1	C2
7	0	1	1	1	1	1	C1

Protein structure prediction

- Let each item denote a motif along with some attributes of that motif: A, B, C, ..
- Given 3D contact map among motifs
- Predict contact points for new sequence

	A	D	T	S	K	C	P
A		0	1	0	0	1	0
D			0	1	0	0	1
T				1	0	0	0
S					0	1	0
K						0	1
C							0
P							

M1	M2	P1		Pn
A	T	..		..
A	C	..		..
D	S	..		..
D	P	..		..
T	S	..		..
S	C	..		..
K	P	..		..

Tutorial overview

- Overview of KDD and data mining
- Data mining techniques
- Data pre-processing
- Data mining in bioinformatics
- Summary

Conclusions

- Scientific and economic need for KDD
- Made possible by recent advances in data collection, processing power, and sophisticated techniques from AI, databases and visualization
- KDD is a complex process
- Several techniques need to be used

Conclusions

- Need for rich knowledge representation
- Need to integrate domain knowledge, especially true in bioinformatics, where patterns are influenced by the three problems:
  - exceptions
  - non-local components
  - inter-dependence

KDD resources pointers

- ACM SIGKDD: www.acm.org/sigkdd
- KDD Nuggets: www.kdnuggets.com
- Book: Advances in KDD, MIT Press, ’96
- Journal: Data Mining and KDD,
  – research.microsoft.com/datamine
- Conference: ACM SIGKDD International Conference on KDD, 1999
  – research.microsoft.com/datamine/kdd99