Data Mining in Bioinformatics

Mohammed J. Zaki

Computer Science Department

Rensselaer Polytechnic Institute

Tutorial overview

- Overview of data mining
- Data mining techniques
- Data mining in bioinformatics
- Summary

Overview of data mining

- What is data mining?
- Why is it necessary
- The mining process
- Mining operations and methods
- Core issues in Mining

What is data mining?

or Knowledge Discovery in Databases (KDD)

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

Massive databases

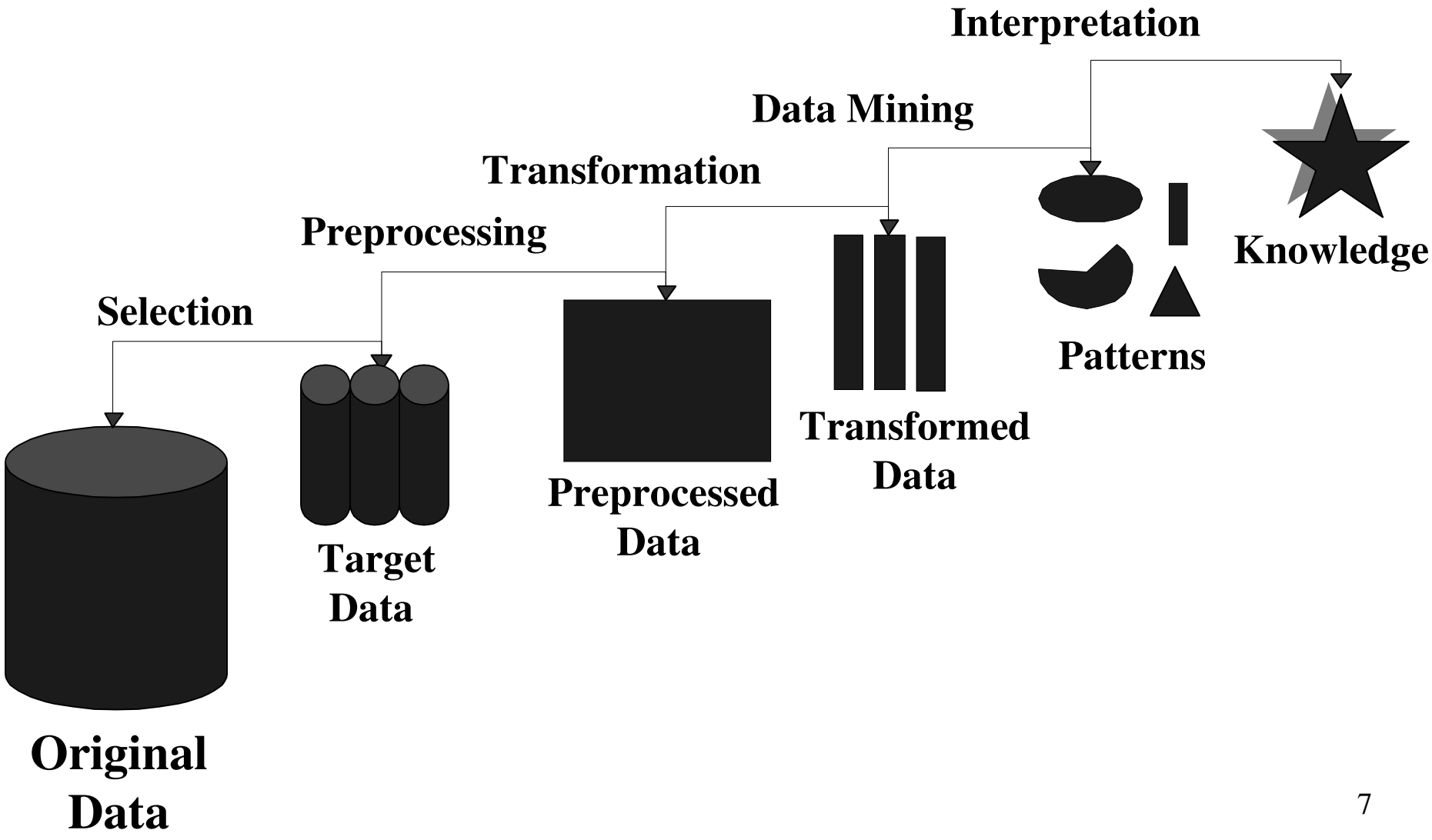
What is data mining?

- 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 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

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

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 in Mining

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

Tutorial overview

- Overview of data mining
- Data mining techniques
- 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)	C
83% (5)	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

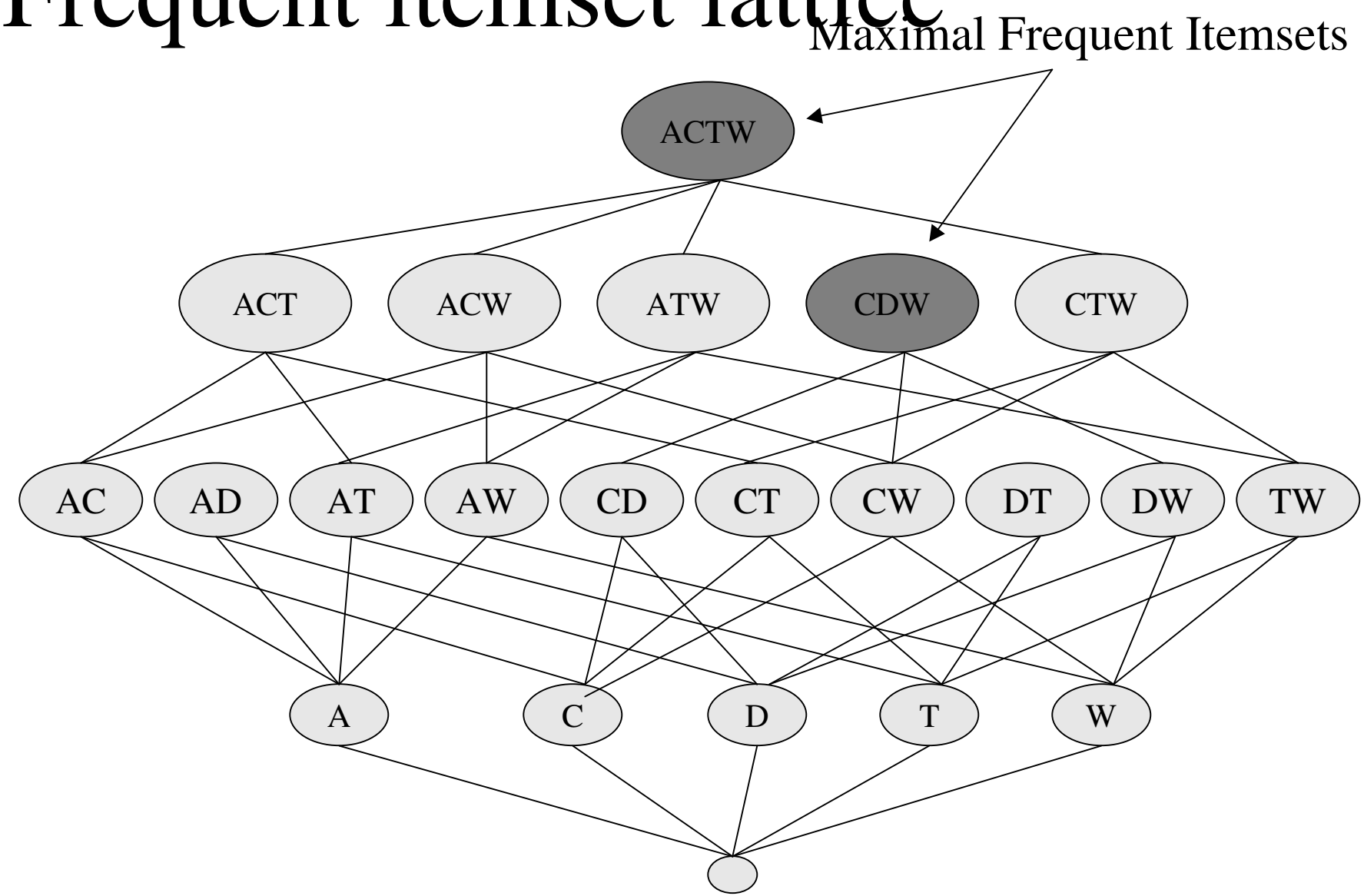
Size	Itemsets
1	A C D T W
2	A C A D A T A W C D C T C W D T D W T W
3	A C T A C W A T W C D W C T W
4	A C T W

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

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

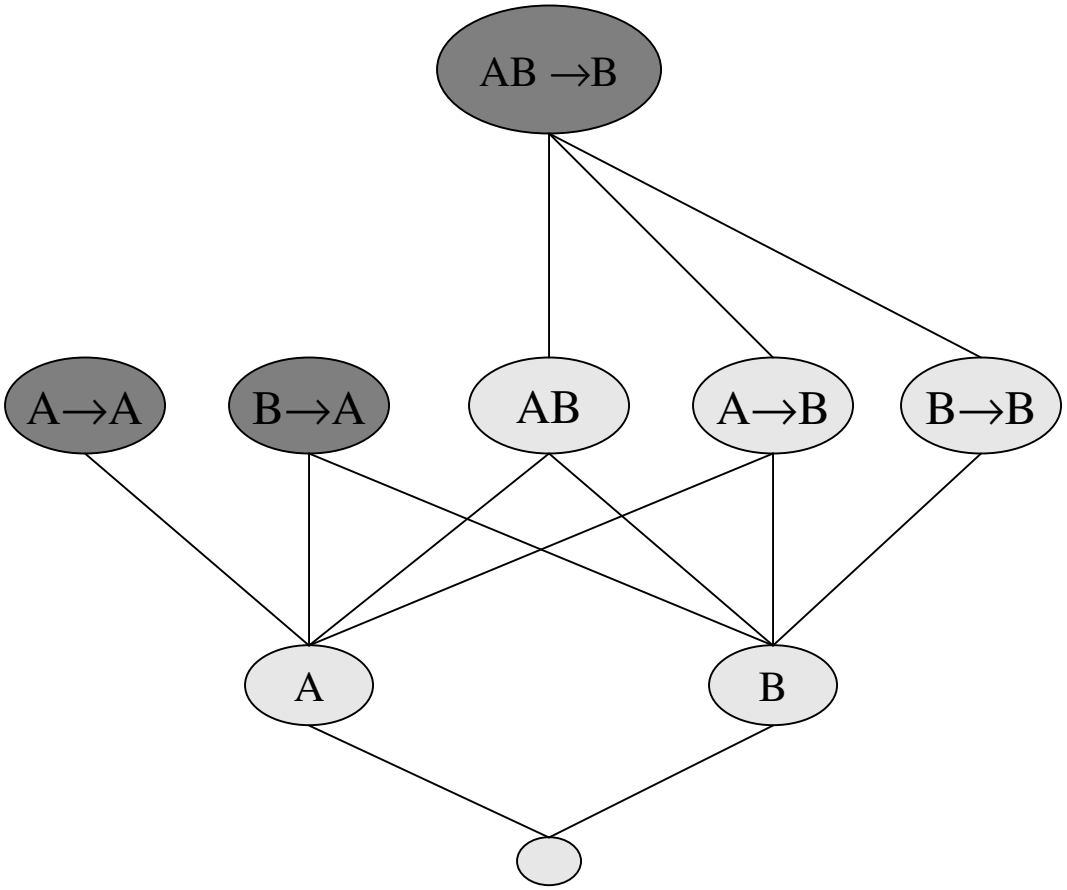
CID	Time	Items
1	10	A B
1	20	B
1	30	A B
2	20	A C
2	30	A B C
2	50	B
3	10	A
3	30	B
3	50	A
4	30	A B
4	40	A
4	50	B

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



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

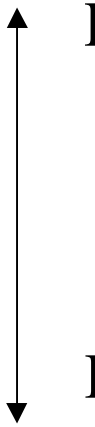
- Supervised learning (labels known)
- Example described in terms of attributes
  - Categorical (unordered symbolic values)
  - Numeric (integers, reals)
- Class (output/predicted attribute): categorical for classification, numeric for regression

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.

Models

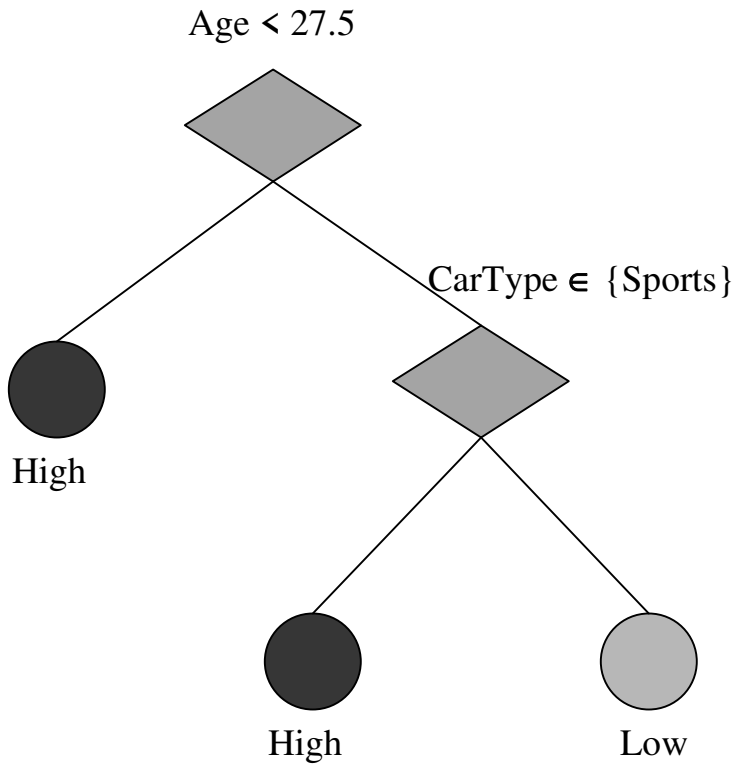
- Some models are better than others
  - Accuracy
  - Understandability
- Models range from easy to understand to incomprehensible
  - Decision trees
  - Rule induction
  - Regression models
  - Neural networks



Decision-tree classification

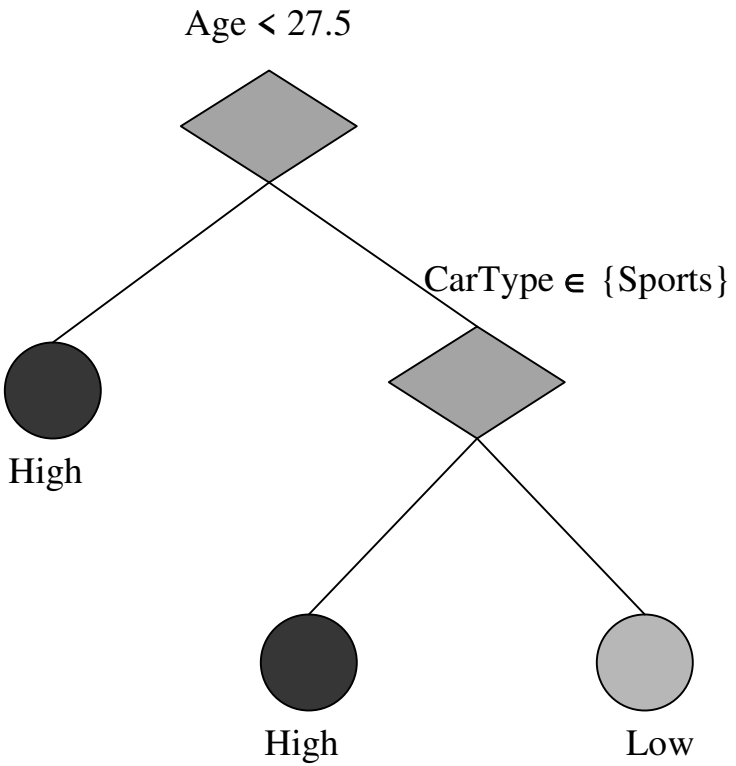
Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

Numeric Categorical



Age=40, CarType=Family ⇒ Class=Low

From tree to rules



1) Age < 27.5 ⇒ High

2) Age >= 27.5 and
   CarType = Sports ⇒ High

3) Age >= 27.5 and
   CarType ≠ Sports ⇒ 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

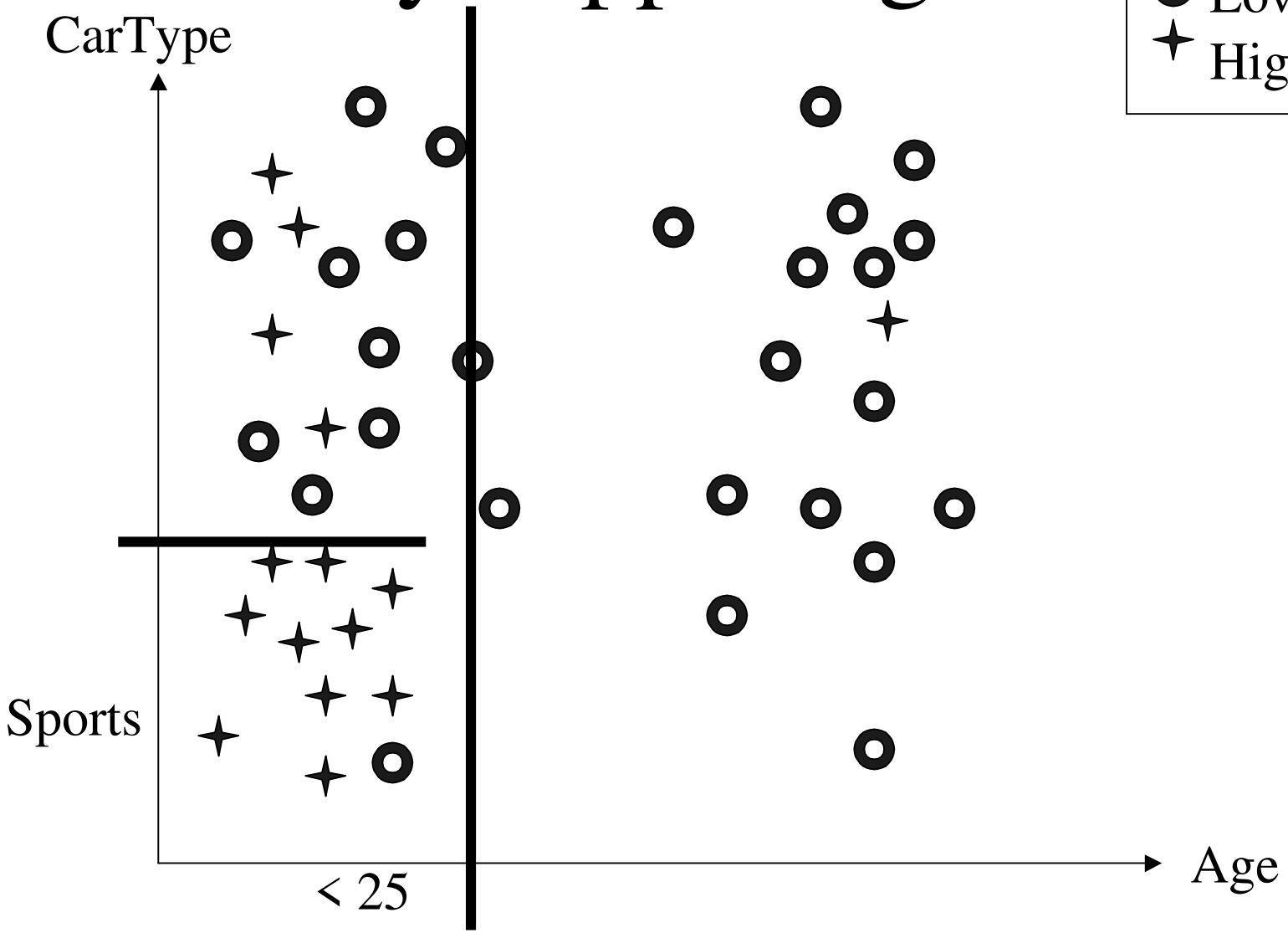
❧ Prune tree

- 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

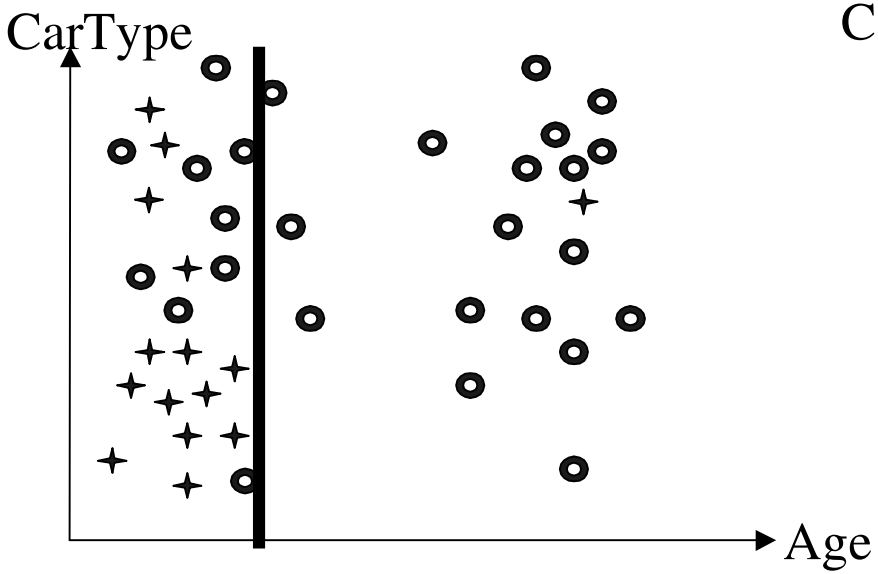
- Use Gini index for partition purity

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

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

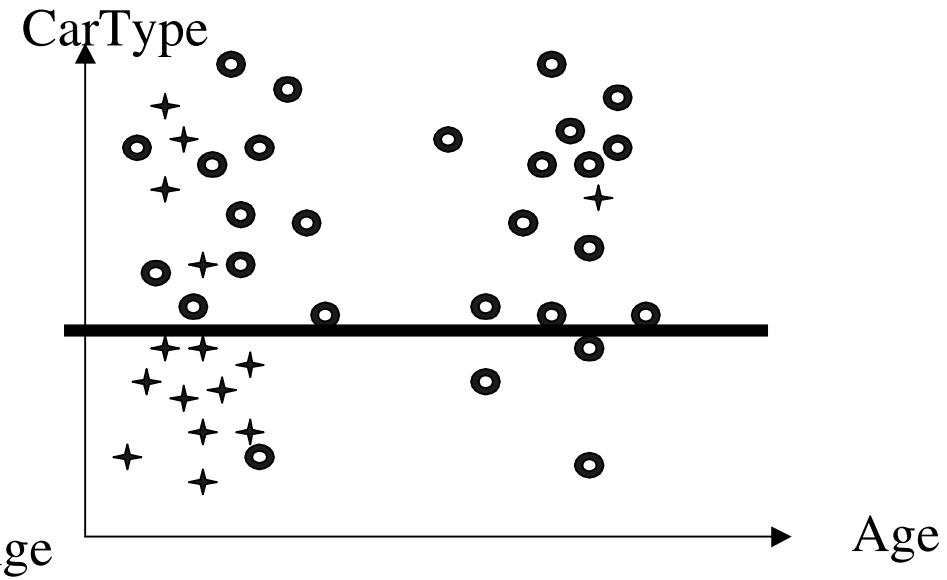
- If S is pure, Gini(S) = 0
- Find split-point with minimum Gini
- Only need class distributions

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

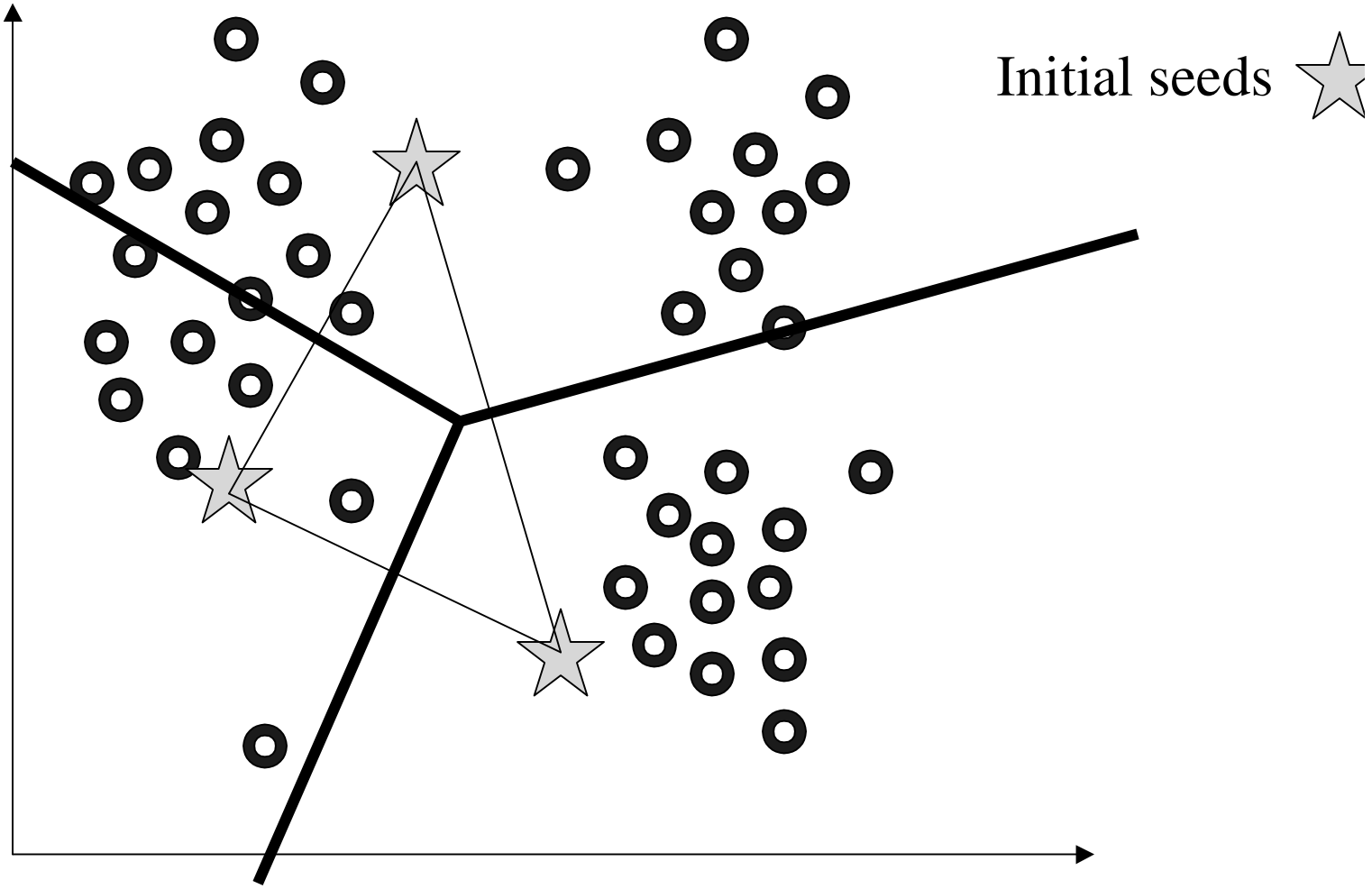
Before clustering

- 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

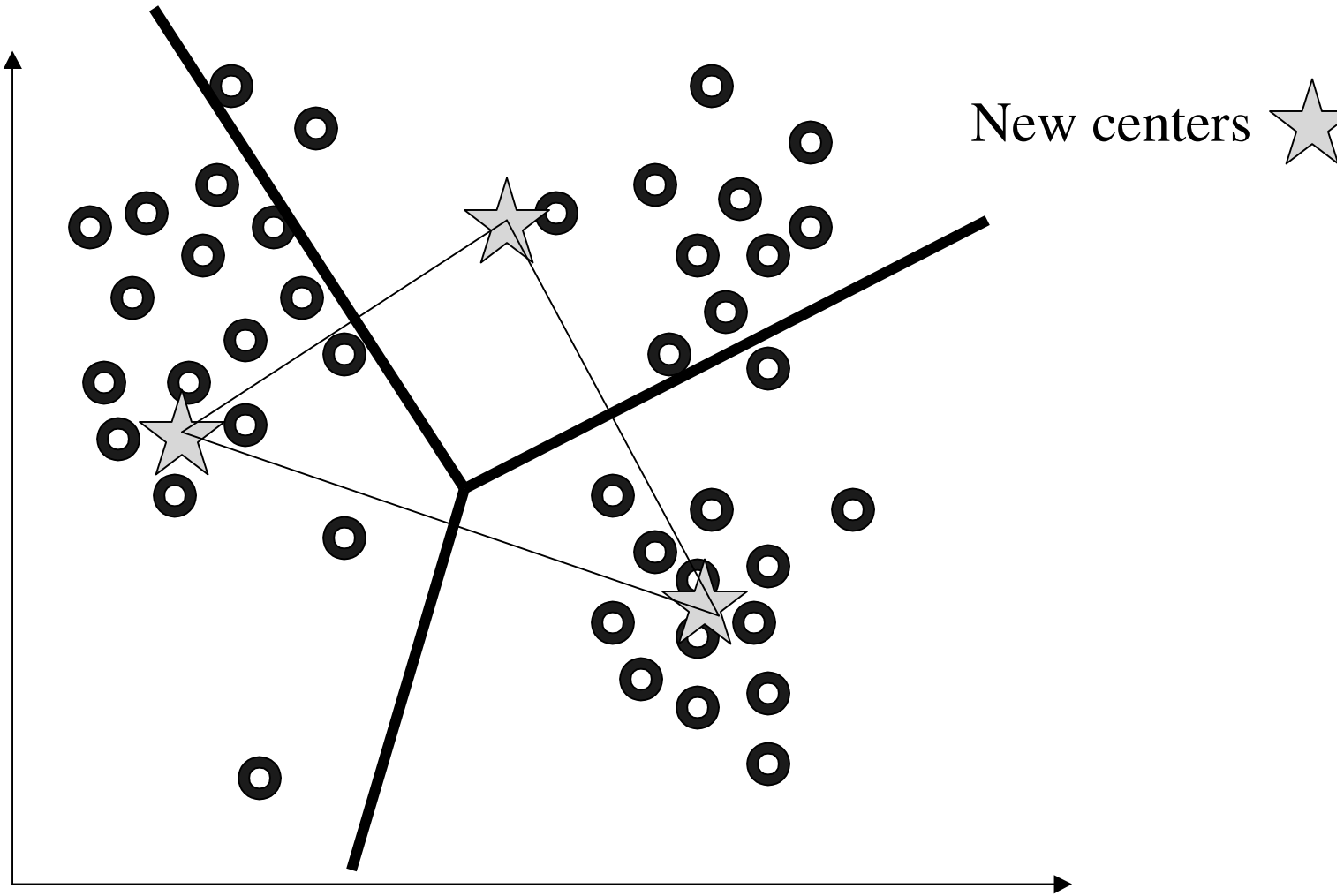
The k-means algorithm

- 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

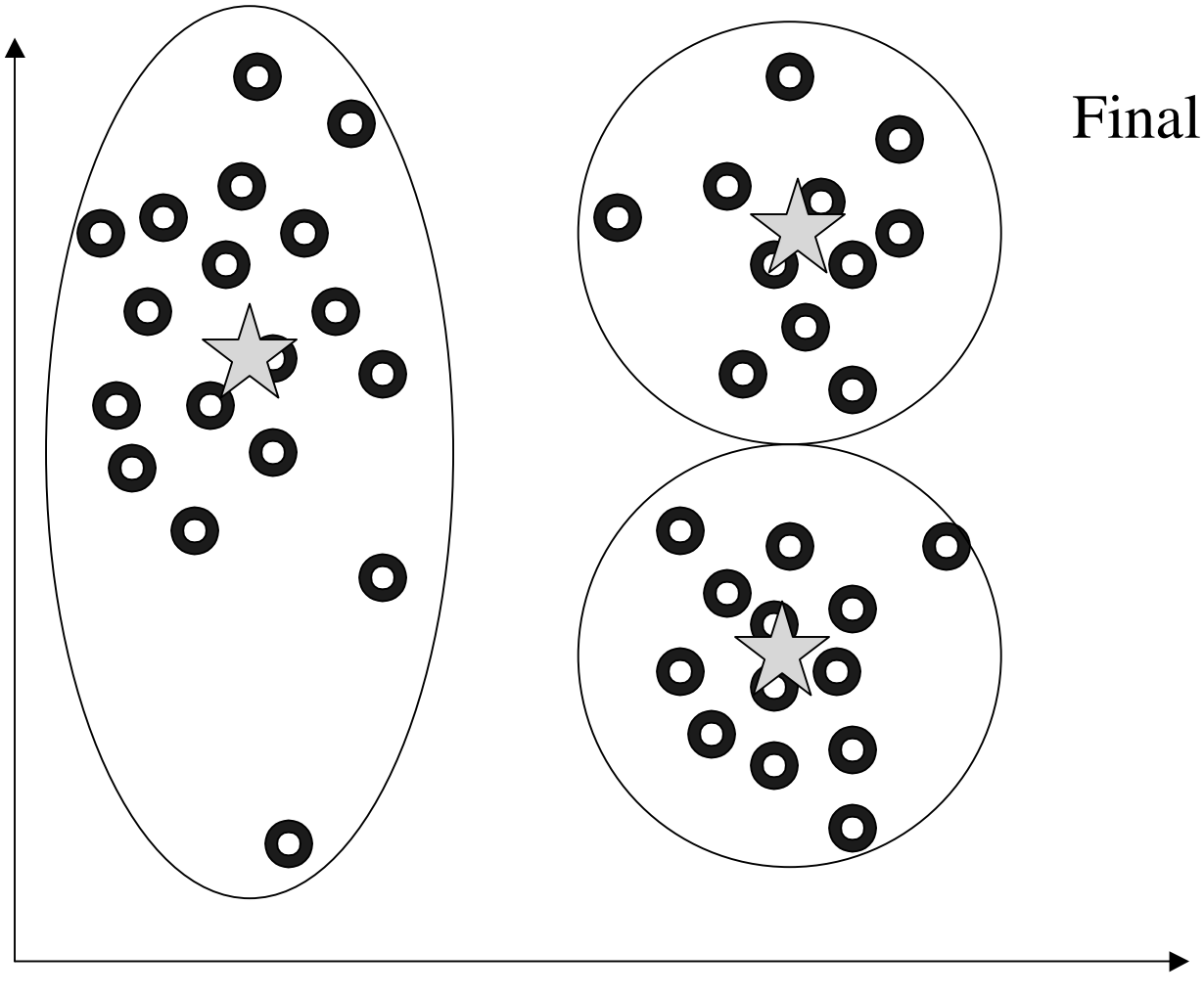
K-means algorithm



K-means algorithm



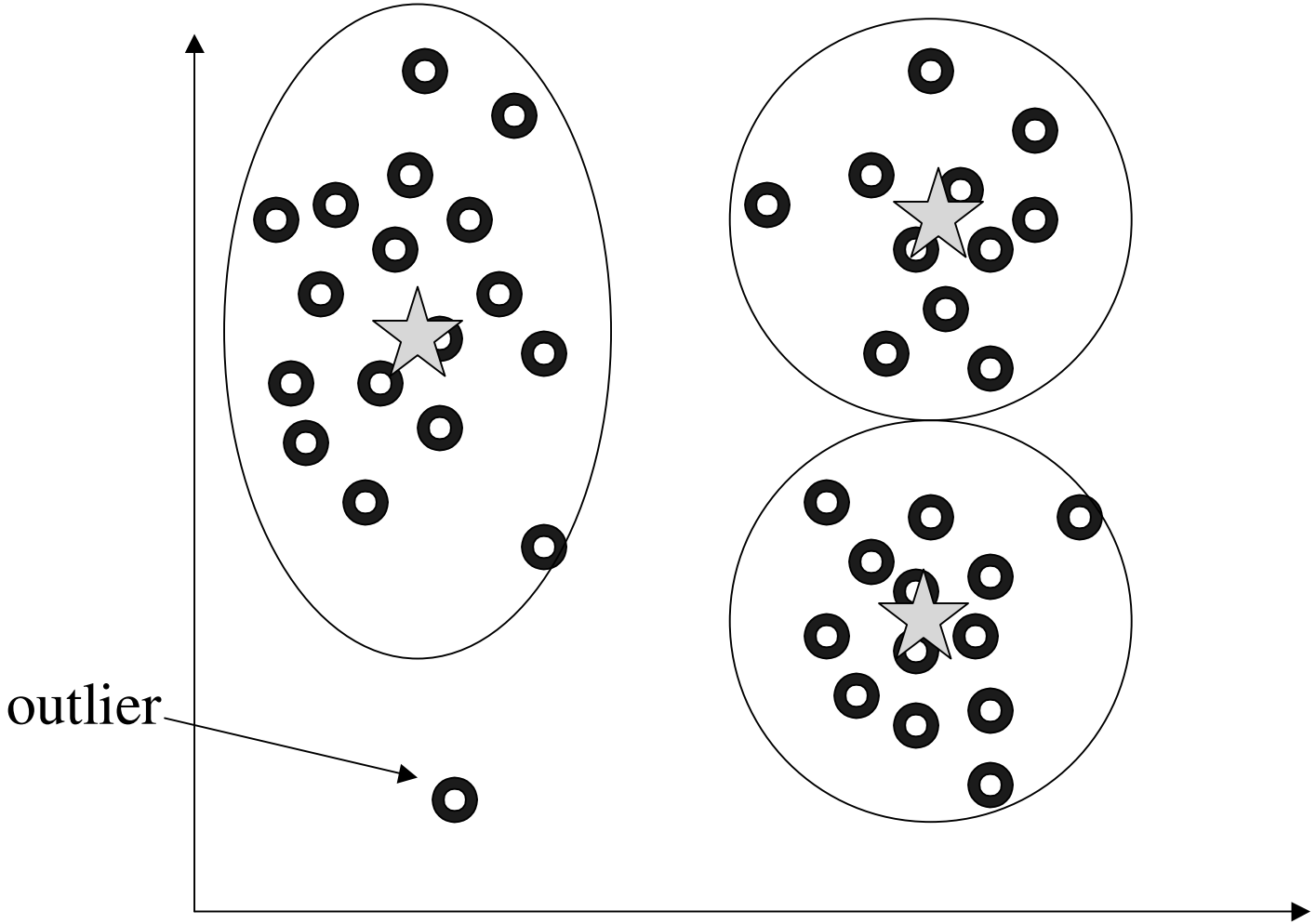
K-means algorithm



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

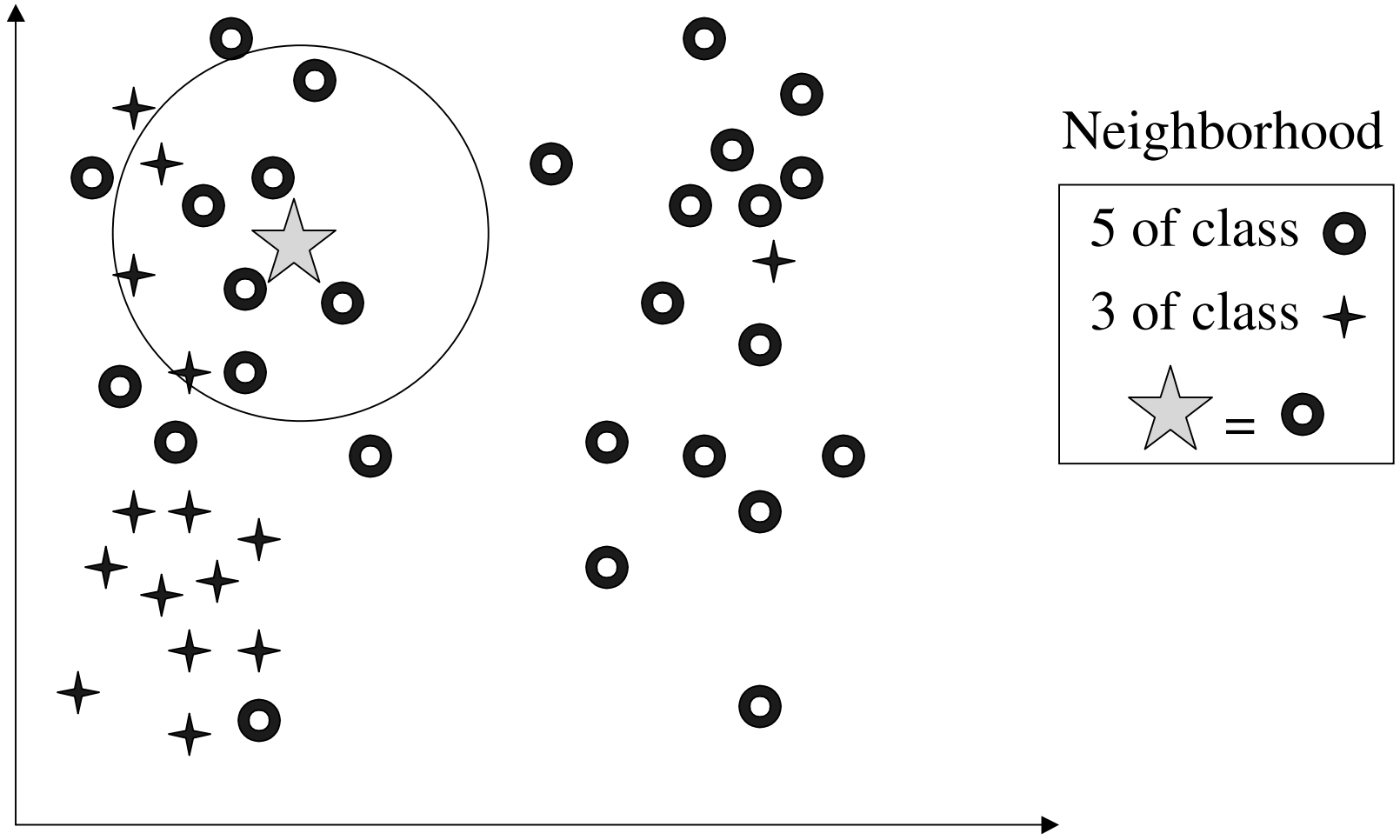
Deviation detection



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 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)

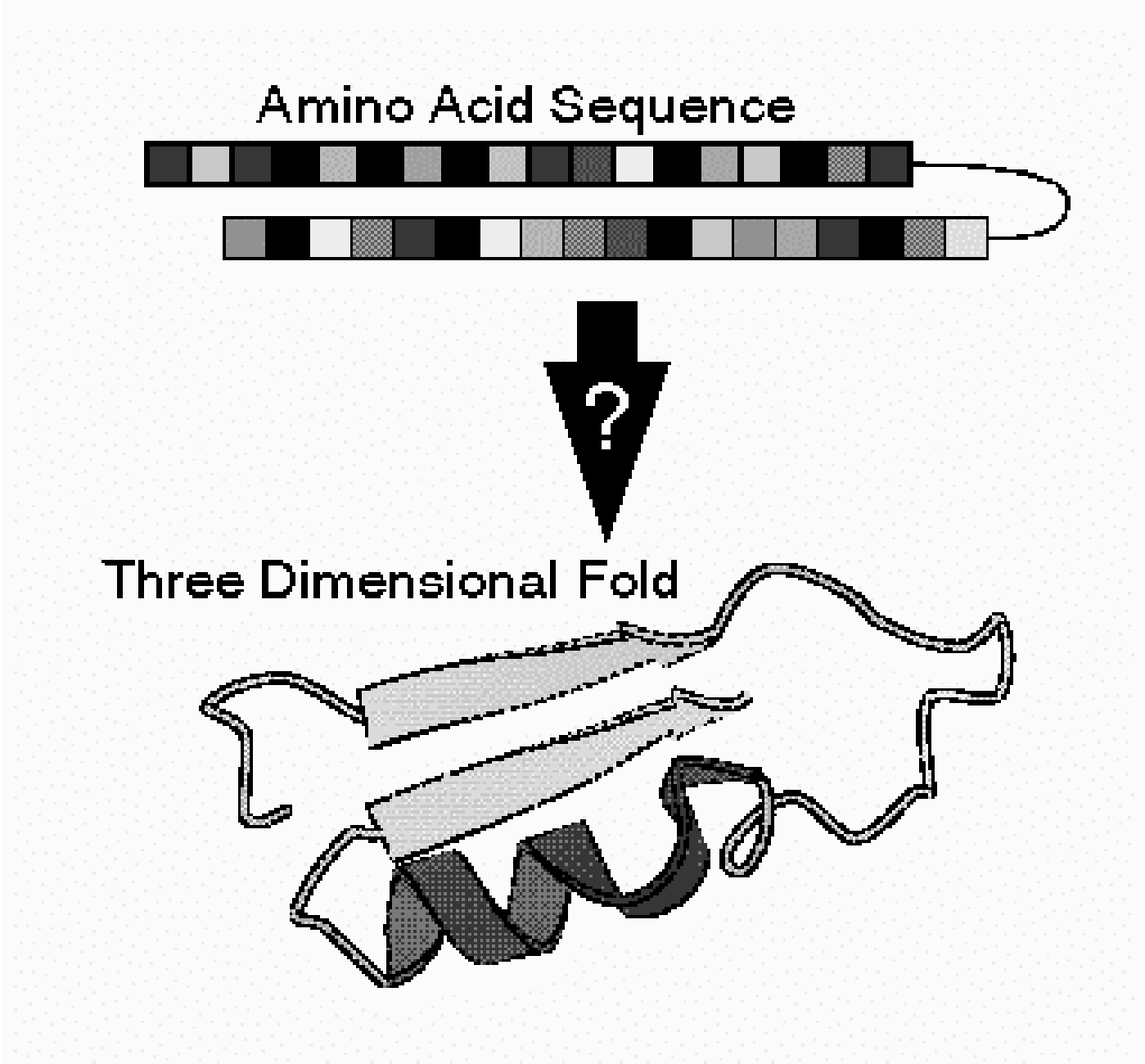
DNA sequence classification

- 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

The Protein Folding Problem



An HMM for Local Predictions

- HMMSTR (Chris Bystroff, Biology, RPI)
- Build a library of short sequences that tens to fold uniquely across protein families: the I-Sites Library
- Treat each motif as a Markov chain
- Merge the motifs into a global HMM for local structure prediction

Training the HMM

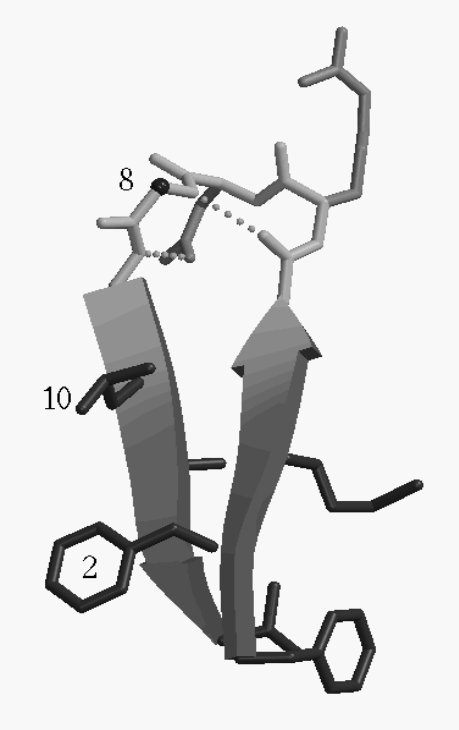
- Build I-sites Library
  - Short sequence motifs (3 to 19)
  - Exhaustive clustering of sequences
  - Non-redundant PDB dataset (< 25% similarity)
- Build an HMM
  - Each of 262 motifs is a chain of Markov states
  - Each state has seq and structure for one position
  - Merge I-sites motifs hierarchically to get one global HMM for all the motifs

HMM Output

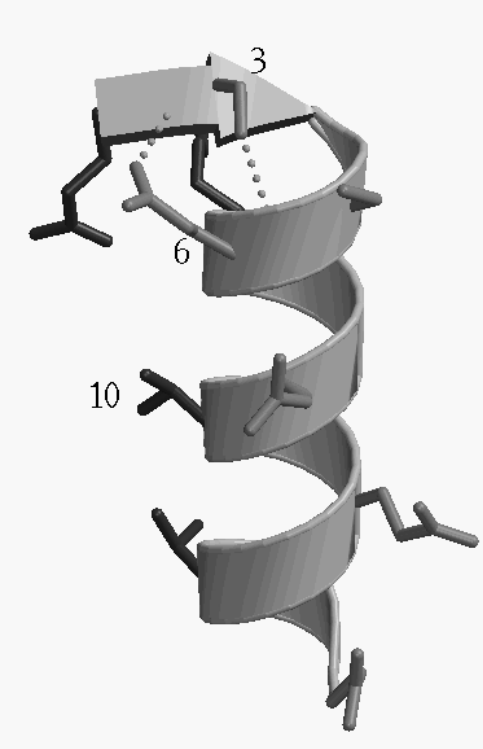
- Total of 282 States in the HMM
- Each state produces or “emits”:
  - Amino acid profile (20 probability values)
  - Secondary structure (D) (helix, strand or loop)
  - Backbone angles (R) (11 dihedral angle symbols)
  - Finer structural context (C) (10 context symbols)

I-Sites Motifs (Initiation Sites)

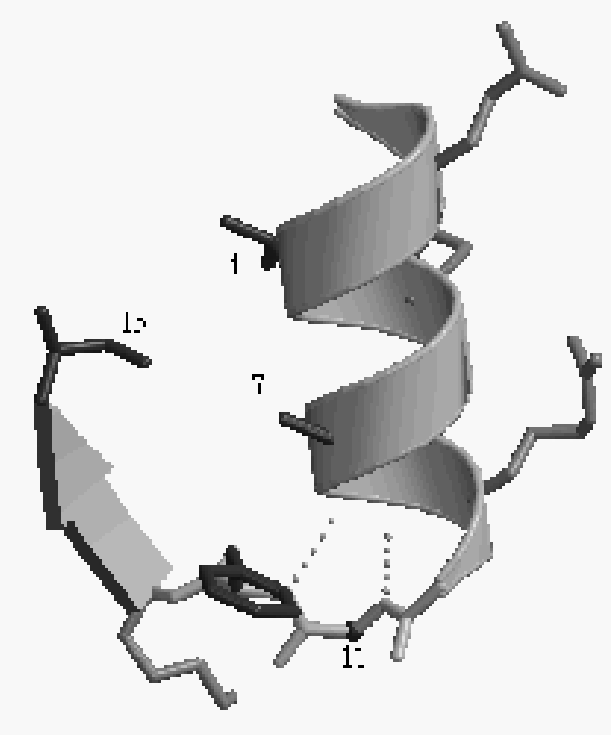
Beta Hairpin

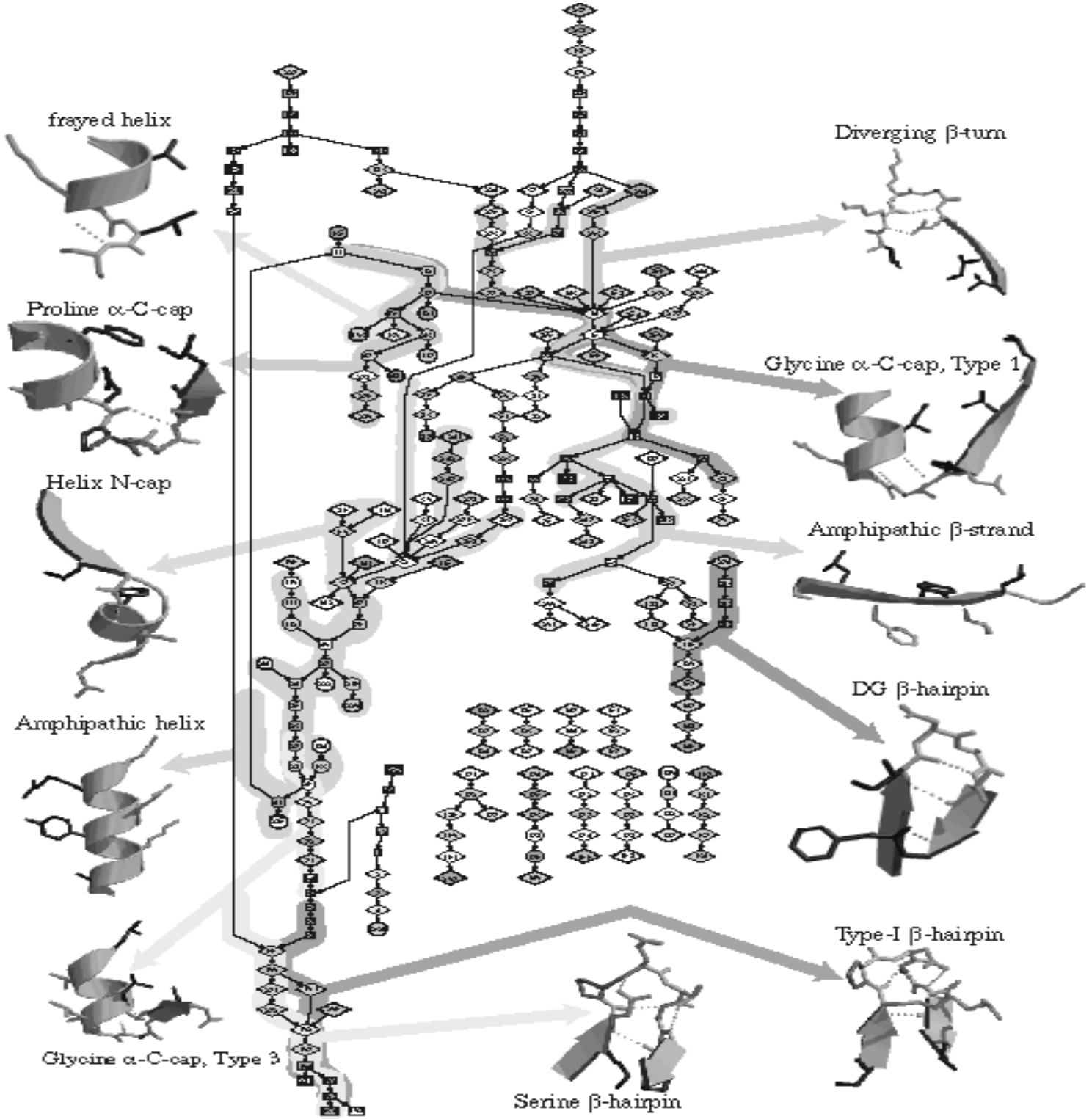


Beta to Alpha



Helix C-Cap

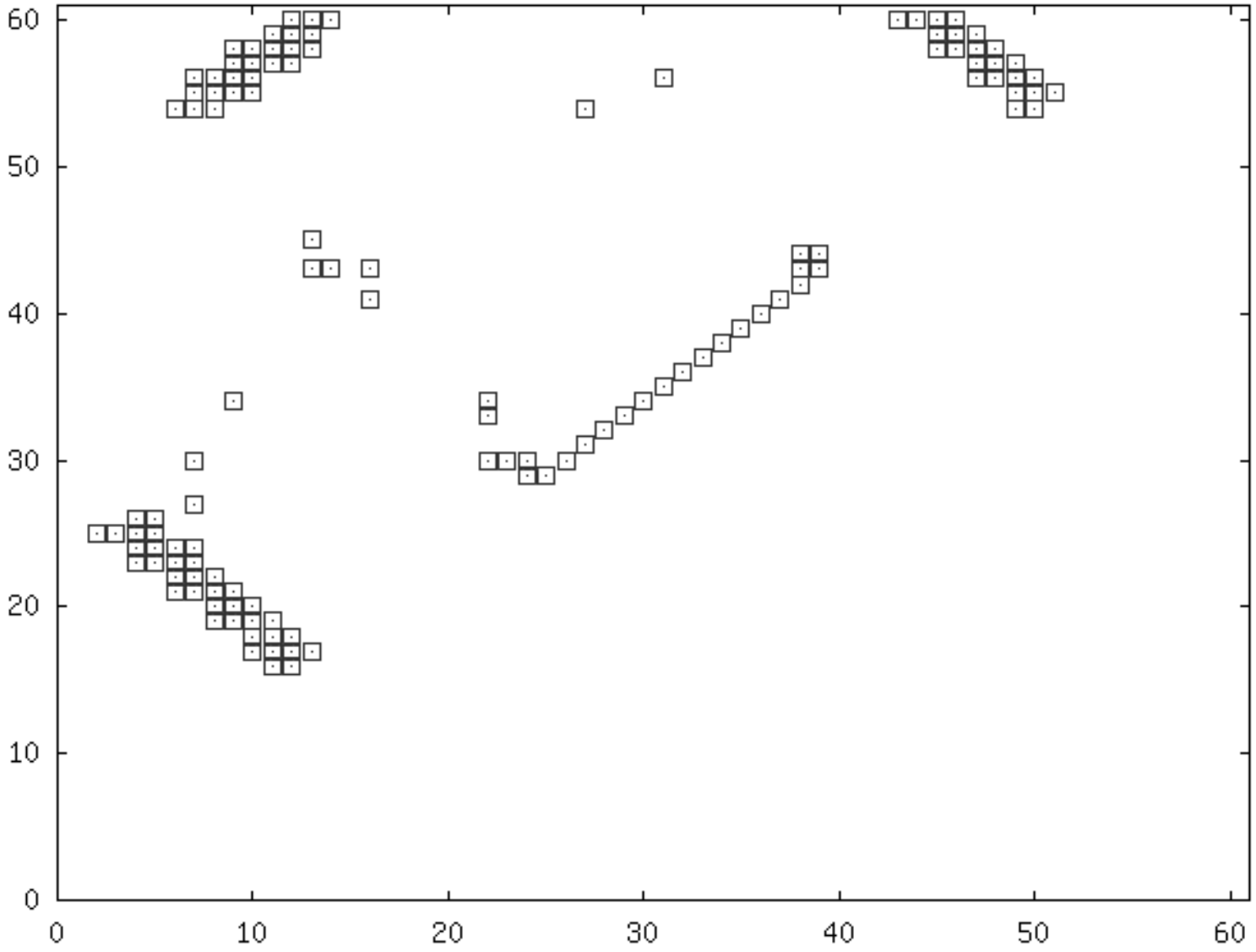



frayed helix
Diverging β-turn
Proline α-C-cap
Glycine α-C-cap, Type 1
Helix N-cap
Amphipathic β-strand
DG β-hairpin
Amphipathic helix
Type-I β-hairpin
Glycine α-C-cap, Type 3
Serine β-hairpin

Contact Map

- Amino acids Ai and Aj are in contact if their 3D distance is less than threshold (7å)
- Sequence separation is given as |i-j|
- Contact map C is an N x N matrix, where
  - C(i,j) = 1 if Ai and Aj are in contact
  - C(i,j) = 0 otherwise
- Consider all pairs with |i-j| >= 4

Contact Map (2igd PDB)



Data Format and Preparation

- Take the 794 PDB proteins
- Compute optimal alignment to HMM
  - Find best state sequence for the observed acids
  - Output probability distribution of a residue over all the 282 HMM states
- Integrate the 3 datasets
  - Alignment probability distribution (Nx282)
  - Amino acid and context information (D, R, C)
  - Contact map (NxN)

HMMSTR Output (per Protein)

PDB Name: 153l_
Sequence Length: 185

Position: 1
Residue: R
Coordinates: 0.0, -73.2, 177.6
AA Profile (20 values): 0.0 1.0 0.0
HMMSTR State Probabilities (282 values):
0.0 0.7 0.3 0.0
Distances (185 values): 0 3 5 18 15 13
...
Position: 185
Residue: Y
Coordinates: -88.7 , 0.0, 0.0
AA Profile (20 values): 0.0 0.4 0.6 0.0
HMMSTR State Probabilities (282 values):
0.0 0.2 0.5 0.3 0.0
Distances (185 values): 15 13 10 5 3 0

Classification Problem

- A pair of amino acids (Ai,Aj) is an instance
- The class: C or NC (contact or non-contact)
- Highly skewed class distribution
  - (1.7% C and 98.3% NC; 300K C vs 17,3M NC)
- Features for each instance
  - Ai Aj Di Dj Ri Rj Ci Cj
  - Profile: pi1 pi2 … p120 pj1 pj2 … pj20
  - HMM States: qi1 qi2 .. qi282 qj1 qj2 .. qj282
  - Class: C or NC

Predicting Protein Contacts

- Predict contacts 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							

Ai	Aj	F1		Fn
A	T	..		..
A	C	..		..
D	S	..		..
D	P	..		..
T	S	..		..
S	C	..		..
K	P	..		..

Association Mining

- Mine Frequent Itemsets in C data
- Compute Frequency in NC dataset
- Prune mined results
  - compute probability of occurrence in C and NC
  - prune pattern if ratio of contact to non-contact probability is less then some threshold
  - i.e., keep only the patterns highly predictive of contacts

Testing Phase

- 90-10 split into training and testing
- 2.4 million pairs, with 36K contacts (1.5%)
- Evidence calculation:
  - Compute matching patterns for each instance
  - Compute cumulative frequency in C and NC
  - Compute evidence: ratio of freq in C over NC
- Prediction:
  - Sort all test examples based on evidence
  - Predict the top PR fraction of examples as contacts

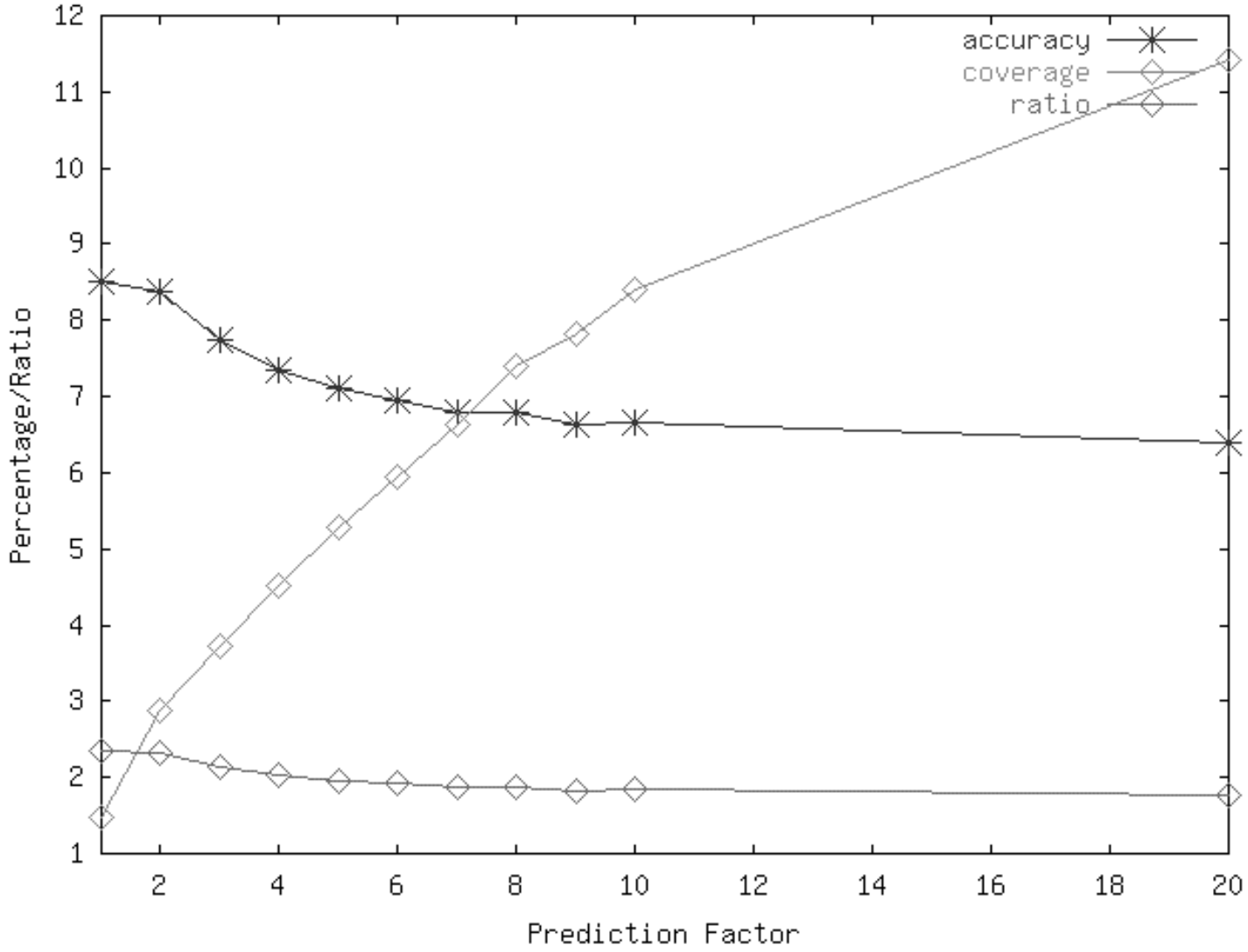
Experiments

- 794 Proteins from Protein Data Bank
- Distinct structures (< 25% similarity)
- Longest: 907, Smallest: 35 amino acids
- 90-10 split for training-testing
- Total pairs: 20 million (> 2.5 GB)
- Contacts: 330 thousand (1.6%)
- Highly uneven class distribution

Evaluation Metrics

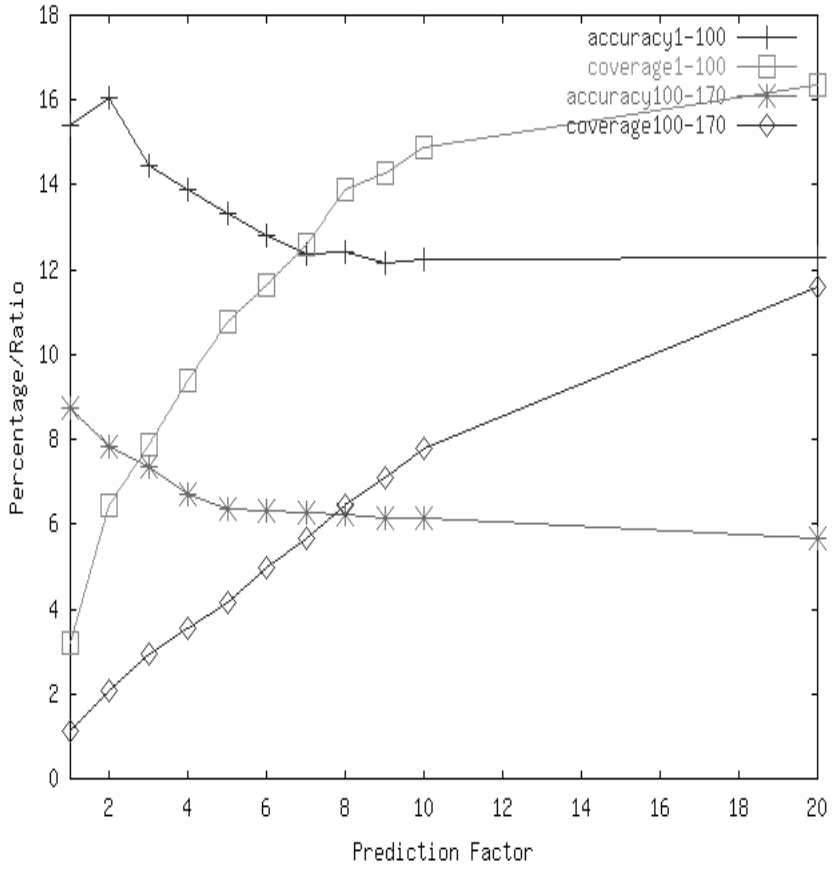
- Na: set of all pairs
- Na*: all pairs with positive evidence
- Ntc: true contacts in test data
- Ntc*: true contacts with positive evidence
- Npc: predicted contacts
- Ntpc: correctly predicted contacts
- Accuracy = Ntpc / Npc
- Coverage = Ntpc / Ntc
- Prediction Ratio (PR): Ntc*/Na*
- Random Predictor Accuracy: Ntc/Na

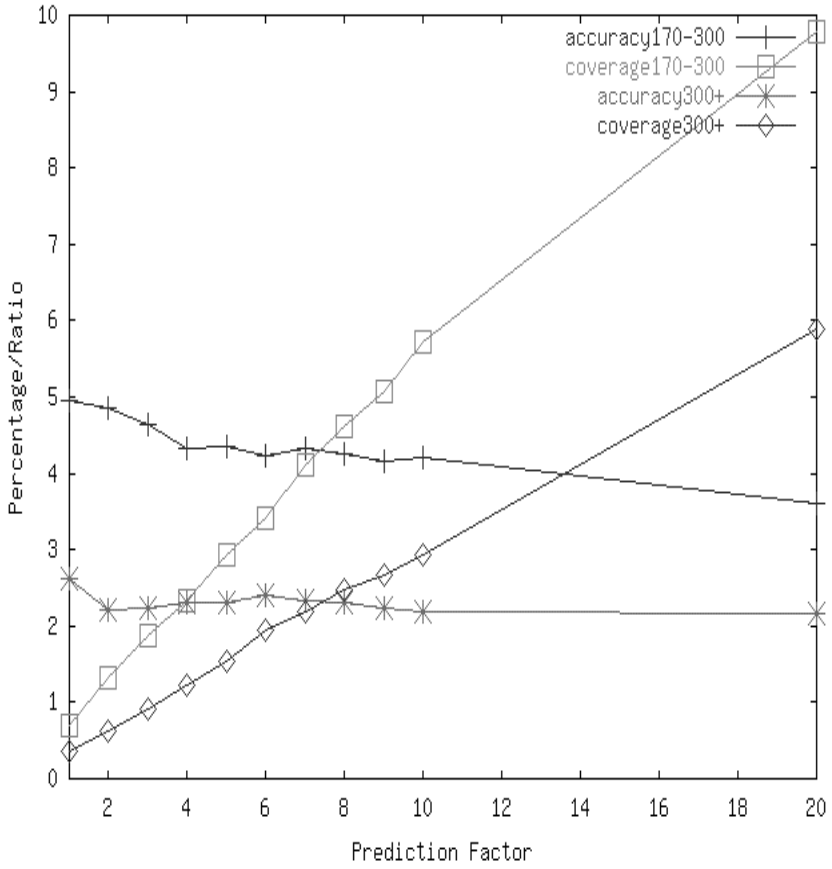
Results (Amino Acids; All)



Crossover: 7% accuracy and 7% coverage; 2 times over Random

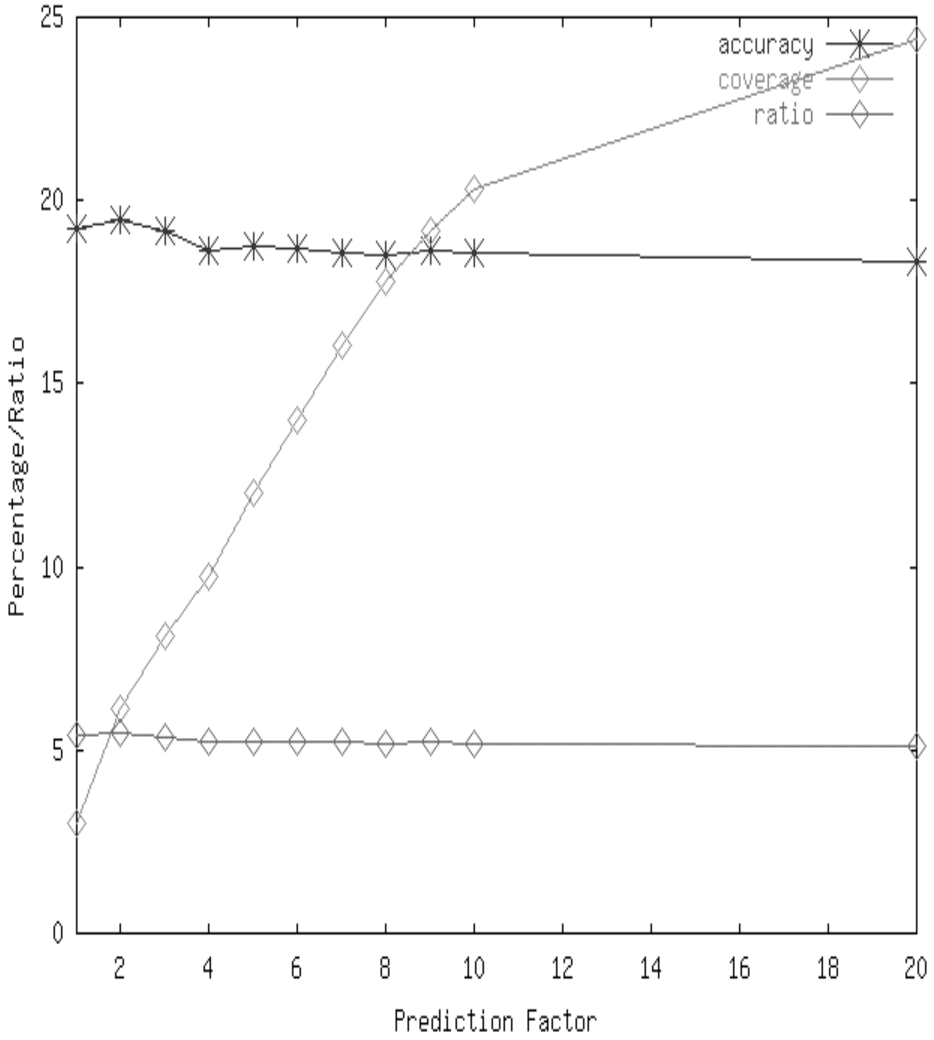
Results (Amino Acids; by len)

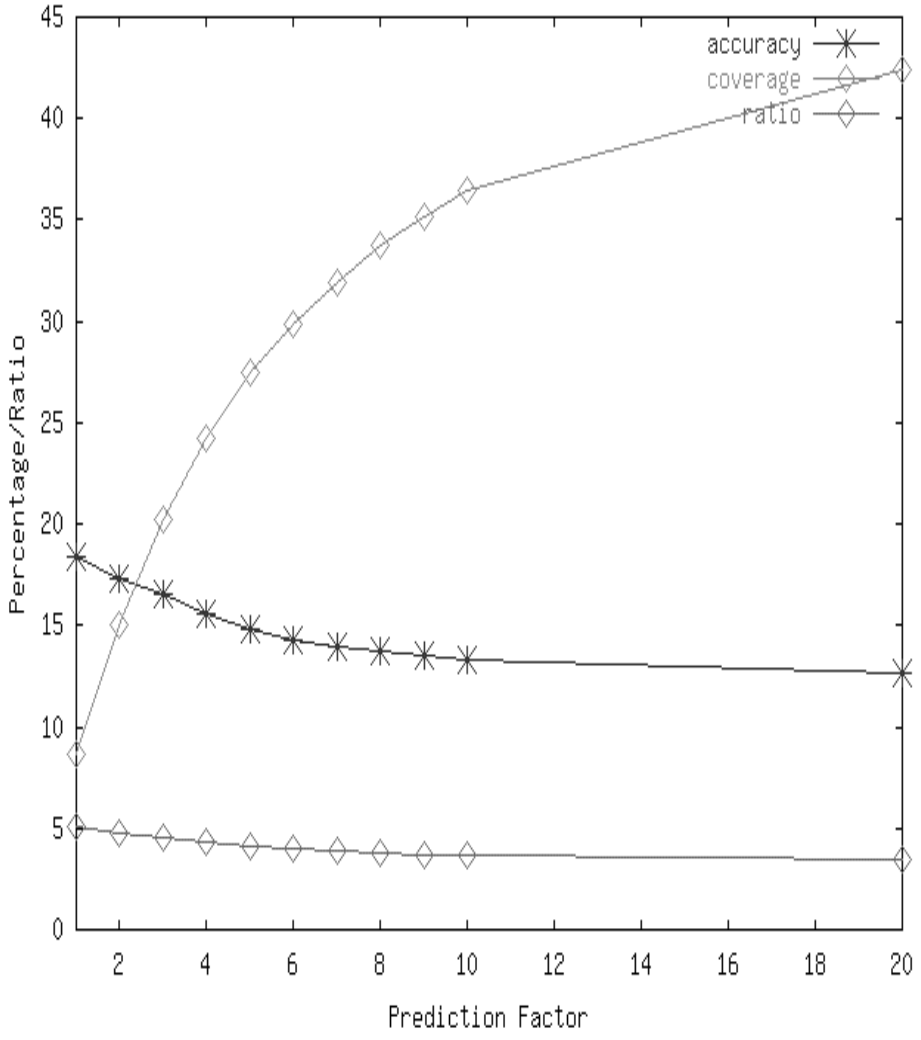




1-100: 12% accuracy(A) and coverage (C); 100-170: 6% A and C
170-300: 4.5% A and C; 300+: 2% A and C

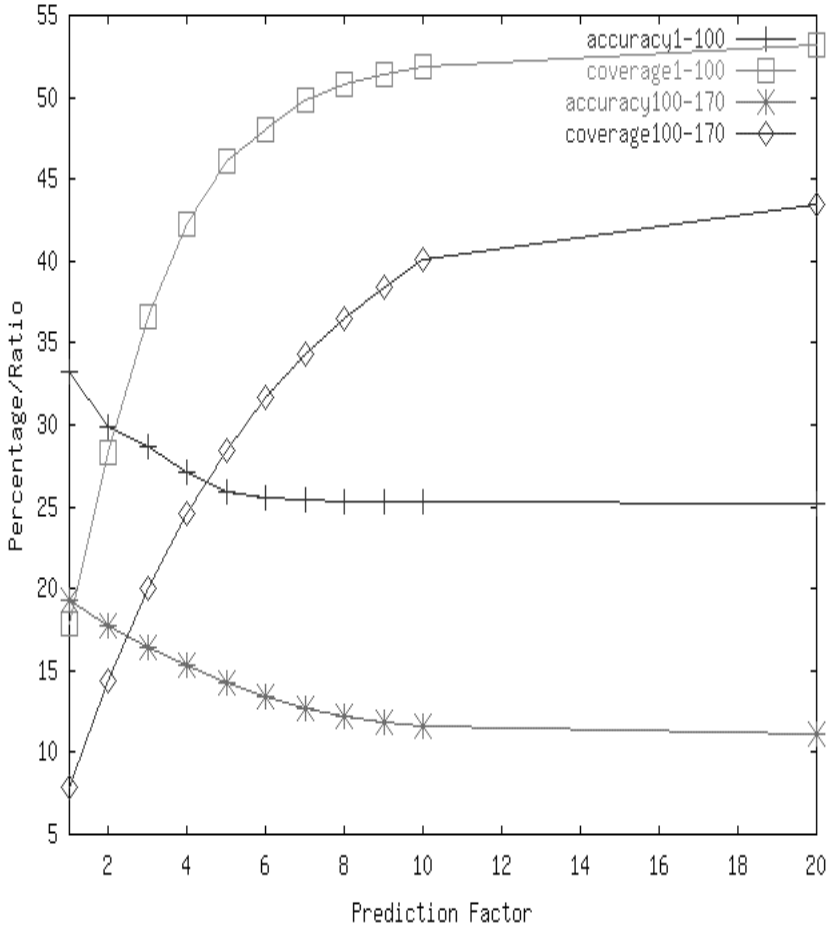
HMM + AA + (R,D,C) ; All

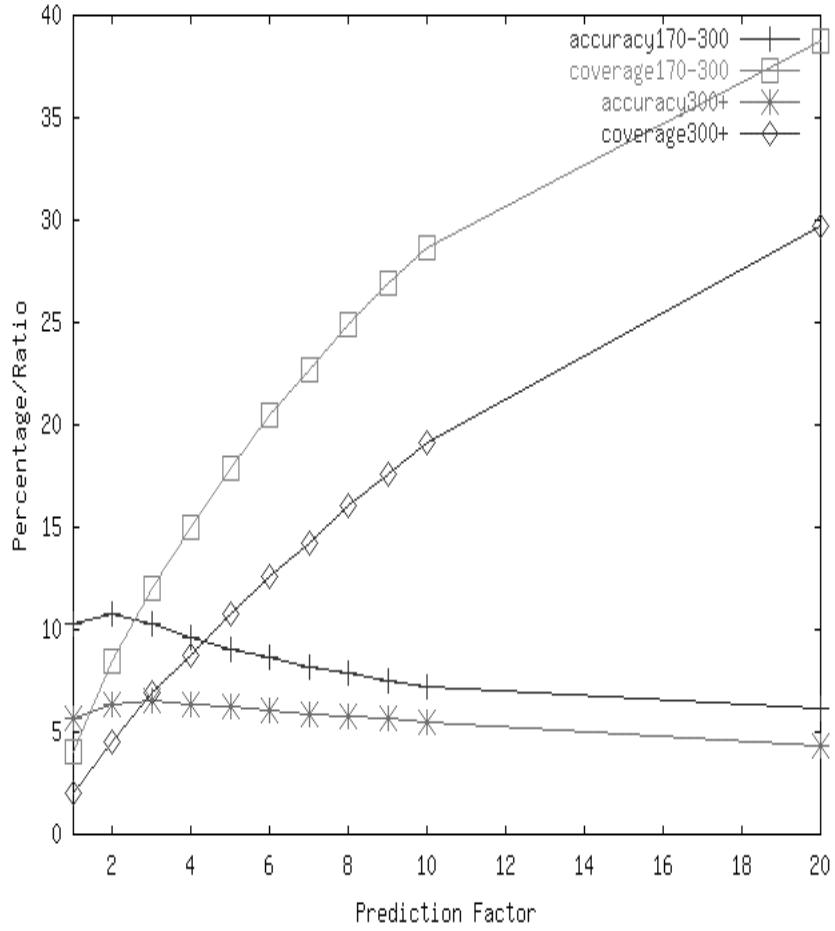




Crossover: 19% accuracy and coverage; 5.3 times over Random
Crossover (+RDC): 17% accuracy and coverage; 5 times over Random

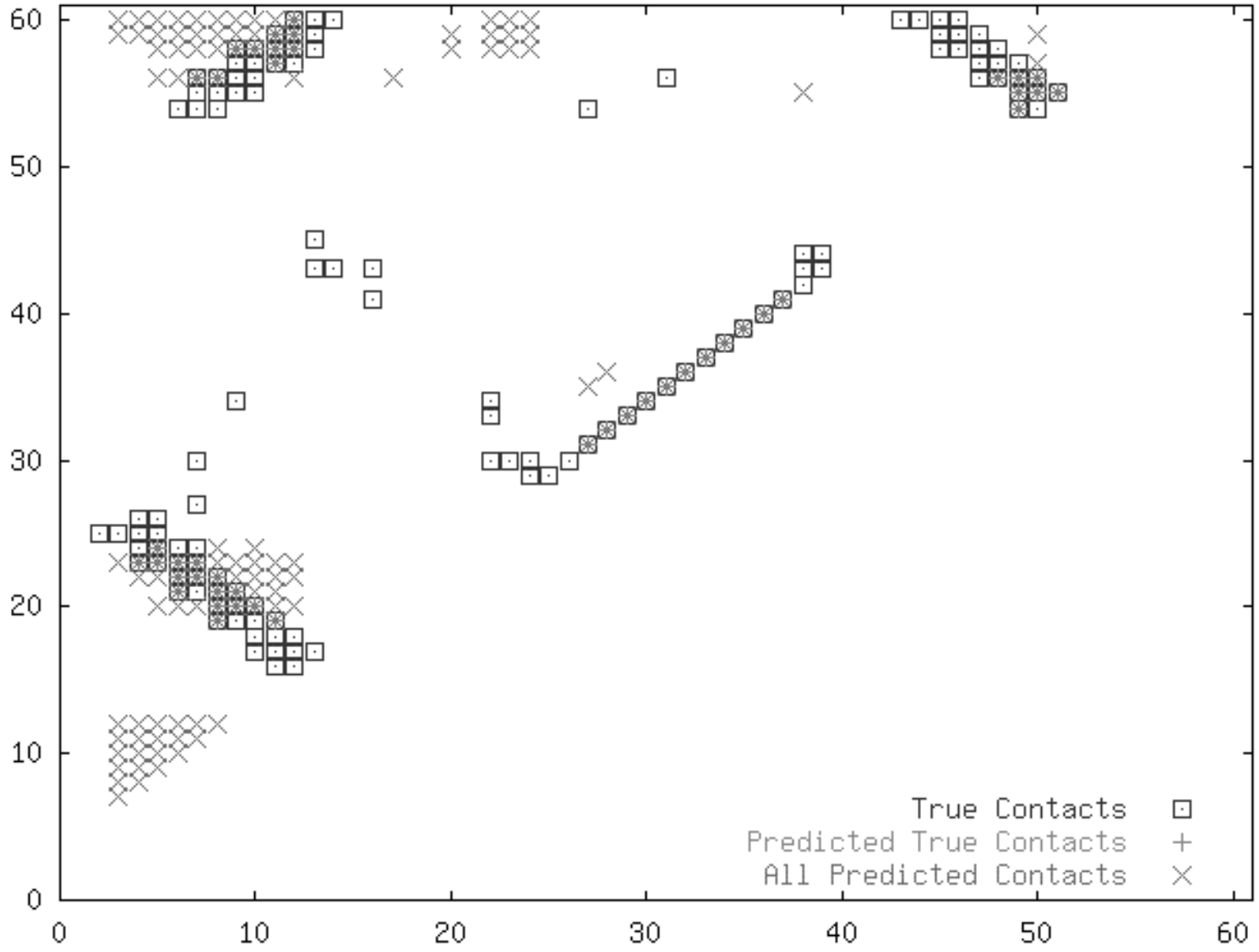
HMM + AA + R,D,C (by length)





1-100: 30% accuracy(A) and coverage (C); 100-170: 17% A and C
170-300: 10% A and C; 300+: 6% A and C

Predicted Contact Map (2igd)



Summary

- Challenging prediction problem
- In essence, we have to predict a contact matrix for a new protein
- Hybrid HMM/Associations approach
- Best results to-date: 19% A and C
  - 14.4% Accuracy (Fariselli, Casadio ‘99; NN)
  - 13% Accuracy (Thomas et al ‘96)
  - Short proteins: 26% (vs. our 30%) (Olmea, Valencia, ‘97)

Tutorial overview

- Overview of data mining
- Data mining techniques
- Data mining in bioinformatics
- Summary

Conclusions

- Scientific and economic need for data mining
- Made possible by recent advances in data collection, processing power, and sophisticated techniques from AI, databases and visualization
- Mining 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

Data mining 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, 2000
  - www.acm.org/sigkdd/kdd2000