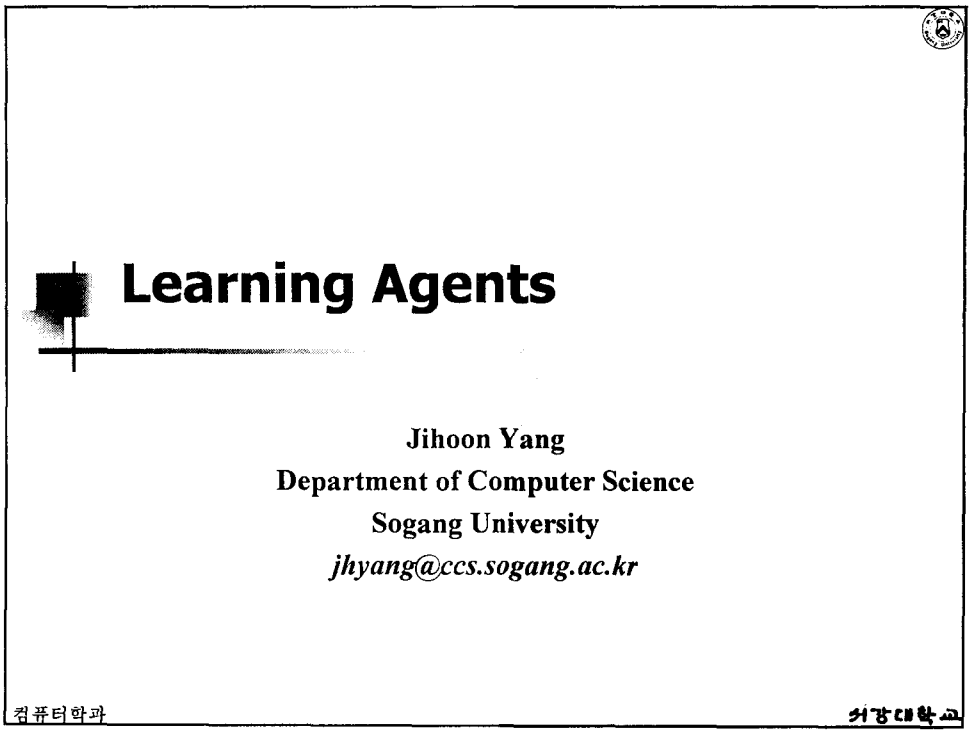


Learning Agents

Jihoon Yang
Department of Computer Science
Sogang University
jhyang@ccs.sogang.ac.kr

Overview

- **Introduction to Learning Agents**
  - Agents
  - Issues
  - Machine learning
- **Knowledge Discovery and Data Mining**
  - Necessity
  - Applications
  - Common algorithms
- **Interesting Applications**
  - Bioinformatics
  - Text mining
- **Summary**

Agents

- **An entity that perceives its environment and performs a set of tasks on behalf of a user with some degree of autonomy**
  - Simple reflex
  - Reflex agent with internal state
  - Goal-based
  - Utility-based
- **Learning provides autonomy**

Learning Agents

- **An agent that improves its performance over time**
  - Avoid repeating the same mistakes
  - Is able to change its behavior: dynamic rather than hard-coded
- **Often used for adaptation to**
  - Dynamically changing environment
  - New situations
  - Needs/interests of the user
  - Behavior of other agents

Questions

- **What should be learned?**
  - Definition of the learning task
    - Input features
    - Output values
  - Who defines the learning task?
    - Agent creator
    - The user
    - Agents by themselves
  - Refinement of learning tasks at run time?
    - Remove unnecessary features
    - Add new features

Questions

- **What learning techniques are used?**
  - Several choices
    - Rule induction
    - Reinforcement learning
    - Neural networks
    - ...
  - Time/space requirements
  - Reuse of existing code?
    - Ad hoc implementation?
    - Plug-in algorithms?

Questions

- **Where do the data come from?**
  - Given by the agent creator?
    - Off-line learning
    - Trained before use
  - Given by the user?
    - On-line learning
    - How long is the user willing to teach?
  - Collected by the agent?
    - How?
    - Performance problems?
    - Who gives the feedback?

Questions

- **Too little data?**
  - Unable to learn (cf. sample complexity in PAC framework)
  - Share data among similar agents?
    - User privacy
    - Data transferring costs
  - Active learning
- **Too much data?**
  - Most learning algorithms do not scale up
  - Distributed/cooperative learning agents

Questions

- **How much data can be stored?**
  - Garbage collection?
  - Is the same data duplicated in several agents?
- **When and where is the learning done?**
  - Mobile learning agents
  - Learning process may be heavy
  - Move elsewhere to learn

Machine Learning

- **Main interest in algorithms**
- **Usually monolithic systems**
  - Only one passive agent
  - Cooperative learning in a multi-agent system
- **DATA MINING**
  - Find useful information from huge databases

An Electronic Information Perspective

The amount of electronic information stored worldwide doubles every 18 months

- **Past 25 years: advances in hardware and software made it possible for large corporations to store transactions electronically**
- **On Line Transaction Processing (OLTP)**
  - Ordering, Billing, Tracking material, Customers
- **Development oriented to make databases easier and more efficient**
  - Languages (e.g., SQL)
  - Structural (e.g., indexing)
  - Models (e.g., Object Oriented, Deductive DB)

Databases, today

- **Corporations collect data at tremendous rate, basically any activity is stored electronically**
  - Business transactions
    - Bank transactions, Credit cards, Store purchases, Phone calls...
  - Scientific data
    - Satellite data, Atmospheric data, DNA chains, Medical data...
  - Factories
    - Product defects, Labor productivity, Customers' feedback, Inventory level, Customer service track...
  - Internet
    - Data available on the net, Logs...
  - Terabytes of data are now available within a corporation
- **Information is one of the most important assets for the upcoming future: Information age**

Data Driven Decisions World

- **Business people recognized the importance of data to support decisions**
- **Data are already available**
- **From a competitive perspective, cannot afford not doing it**
- **Type of user has changed; type of queries has changed; different needs**

  Old: *"How many units of x did we sell in Los Angeles in the last quarter of 1998?"*

  New: *"What is boosting the sales of x in Los Angeles?"*

  New: *"Who are the best prospects of x?"*

•Current approach: Hiring expensive human resources, time consuming

The Problem

"Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" - U.Fayyad et al.

- **A pattern is a statement that identifies a subset of the data and specifies some relationships among its constituents**
  - "if income > 70K and job=stable then Risk=Low"
  - "if beer is purchased then peanuts are purchased in the same transaction with 80% probability"
- **A pattern must be ultimately readable by the user**

Merging Expertise (finally!)

- **Machine learning (AI)**
  - Sophisticated knowledge extraction
  - Very small datasets; basically, no real world problems
  - Contribution: Algorithms
- **Database**
  - Never worked on this problem
  - Know how to deal with large databases and real world problems
  - Contribution: Scalability
- **Statistics**
  - Decades of experience
  - Numerical domain
  - Small datasets applied to real world problems
  - Contribution: Numerical analysis, methodologies
  - Reluctant to join the party

We got the knowledge, now what?

- **Get sense out of the original database (understand the database characteristics)**
  - *if* x is Italian ***and*** x goes out for dinner ***then*** x eats pizza ***with probability 80%***

·**Prediction and classification**

-What does an Italian eat when he/she goes out for dinner?

·**Support decision making**

-Where should we get a group of Italians out for dinner?

·**Support marketing activity**

-What kind of coupons should we send to an Italian?

Different Paradig

- **Database paradigm:** ***"Given a pattern p and a database d find all records in d that match p;"***
- **KDD/DM paradigm:** ***"Given a database d find all 'interesting' patterns p;"***
  - Fuzzier task
  - What does 'interesting' (or valid, novel, potentially useful) mean?
    - X can be interesting today and not tomorrow
    - X can be interesting for Y and not for Z
    - X can be interesting in Y and not in Z
- **Bottom line: To involve knowledge (external or embedded) to evaluate pattern interestingness**

Data Models

- **Classification**
  - "if income > 60K and job=stable then Risk=Low"
  - A set of *independent* variables to explain (i.e., classify) the *dependent* variable (i.e., the class)
  - Used mostly for prediction
  - Learning from known examples
  - Predict categorical values
  - Supervised learning
- **Regression/Estimation**
  - Like classification but oriented to predict continuous variables
  - Apply a regression formula
  - Natural fit in continuos domains
  - Well known and exploited in statistics

Data Models

- **Clustering**
  - Clusters set of records trying to:
    - **Maximize difference among groups of entities**
    - **Minimize difference among entities within the same group**
  - Unsupervised learning

- **Associations/Affinity**
  - Events which are likely to occur together
    - **"if beer is purchased then peanuts are purchased as well 70% of the times"**
    - **"if tires are bought then wheel service is required 95% of the times"**

Data Models

- **Temporal/Sequence pattern analysis**
  - Event relationships over time
    - **"Notebook batteries are bought within 6 months of notebook purchase"**
    - **"Customers who drop credit card usage by 25% for at least three consecutive months are likely to drop the service within the next year"**
    - **"If AT&T stock goes up for two days in a row and DEC stock does not fall during those two days then IBM stock goes up the following day 70% of the times"**

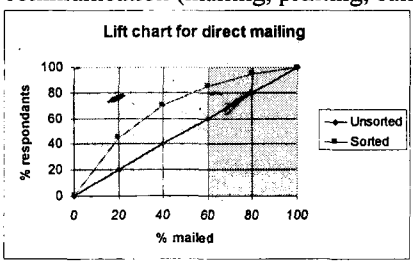
KDD Application Examples

- **Business applications are the widest type of KDD/DM applications**

- **Customer profiling**
  - The current DB is used to classify good customers
  - The discovered patterns (customer profiles) are then used to target new prospects
  - Benefits:
    - Reduces cost to find new customers
    - Keeping current customers
    - Support marketing activity: Direct mail, couponing

KDD Application Examples

- **Direct Marketing/Database Marketing**
  - One of the most successful applications for KDD
  - Many benefits
    - Improved return on investment (ROI)
    - Improved customer relationship and loyalty: they perceive an higher value in your communication
    - Save on communication (mailing, printing, calls)



KDD Application Examples

- **Market-basket analysis**
  - Understand the dynamics of customer purchases
    - Help to market products together
    - Combined promotions
    - Space allocation
- **Cross-Selling**
  - Market-basket joined with customer data
  - Usage of current database to market additional products/services
  - Database already available (no new data needs to be collected)

KDD Application Examples

- **Fraud detection**
  - Great importance for phone companies, credit card issuers, insurance companies, veterans administration
  - Enormous amount of money involved (e.g. procurement, annuity, money laundering)
  - DM helps to identify fraudulent patterns
- **Fault detection/prediction**
  - Quality control
  - Automobile industry
  - Two-valued classification with different weights (e.g., wrong prediction of a fire is more costly than wrong prediction of a non-fire)

KDD Symbolic Approach

- **The discovered knowledge is represented in a symbolic way, that is, understandable by human beings**
- **Support explanation. Most application environments need more than a simple classification, they need to know the reason why a certain classification occurs**
  - Automatic loan approval application: "An applicant does not qualify for a loan" why? ... "He does not have a stable job."
- **Algorithms**
  - Rule discovery (CN2)
  - Tree induction (C4.5, ID3)
  - Bayesian networks
- **Induced knowledge may be ad hoc changed and/or integrated with additional human expertise**
- **Successful approach in general**

KDD Non-symbolic Approach

- **The discovered knowledge is not readable in a clear form, it is scattered in the system itself**
- **"Black box" view**
- **No explanation support (not acceptable in same cases)**
  - Automatic loan approval application: "An applicant does not qualify for a loan" why? ... "I don't know"
- **Algorithms**
  - Neural networks
  - Statistics (linear regression)
- **Neural networks have however played a fundamental role in the overall 'learning from data' problem**
- **More tailored for fully automatic classification systems**

KDD Application Examples

- **Text mining**
  - Categorization
  - Summarization
  - Info extraction
  - Link analysis
  - Relation learning
- **Bioinformatics**
  - Genomics
  - Proteomics
  - Heterogeneous data mining

KDD Process Overview

Goal identification. State clearly what you expect to reach from the KDD investigation. Are you looking for relationships between products and services? Are you targeting new customers? Do you want to optimize your promotion redemption? Be sure the KDD outcome is measurable.

Data selection. Of course not all the data available are important for the KDD process. It is so important at this stage to understand what portion of the database is potentially useful to support the investigation.

DM tool selection. No universal tool is so far available for KDD. A combination of tools is most likely to satisfy the requirements.

Preprocessing. Data are retrieved from the target database and arranged in a convenient form. No standard has yet been defined, so data have to be transformed to ad hoc formats.

Cleansing. Noise in data is reduced. Useless segments of data are removed (both horizontally and vertically).

Transformation. Original variables are combined together. New variables are generated (i.e. by generalization). Data types are converted.

Data Mining. Here the core of the information discovery takes place. The chosen discovery algorithm(s) is invoked and information is produced. Models and patterns are the outcome of this step.

Evaluation. The output from the DM has to be then evaluated and enriched with

Classification Problem Overview

- **Training**
  - Definition:
    - k classes, $C = \{C_1, C_2, \dots, C_k\}$;
    - n (independent) variables, $V_1, V_2, \dots, V_n$, s.t., $\forall$ i, $V_i \in D_i$
    - Independent universe U being the Cartesian product $D_1 \times D_2 \times \dots \times D_n$;
  - Given a *Training Set*
    - A finite set T of *classified* tuples, $(r_1, r_2, \dots, r_n, y)$, s.t., $\forall$ i, $r_i \in D_i$ and $y \in C$;
  - Build:
    - A (classification) map $F: U \rightarrow C$
- **Classification (or Generalization)**
  - Classify a new (previously unseen) tuple $t'=(r_1, r_2, \dots, r_n)$, $\forall$ i, $r_i \in D_i$, as: $c'=F(t')$
- ***Different ways to represent F***

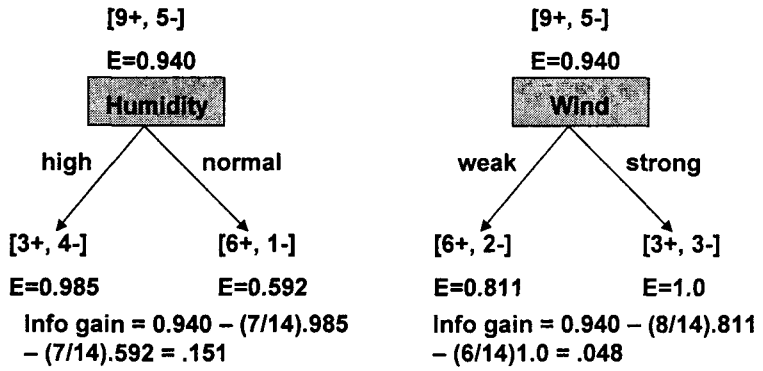
Algorithms -- Decision Tree (C4.5)

Day	Outlook	Temperatu	Humidit	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

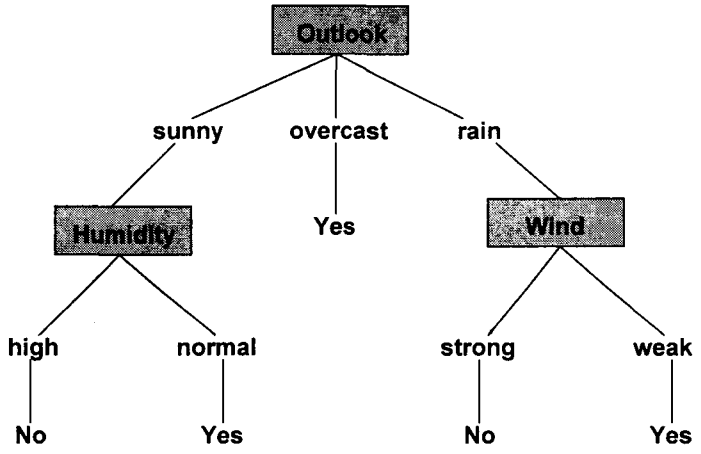
Algorithms -- Decision Tree (C4.5)

- **Feature selection for split**

$$Entropy = \sum_{i=1}^{c} - p_i \log_2 p_i$$



Algorithms -- Decision Tree (C4.5)



Criteria to build the tree

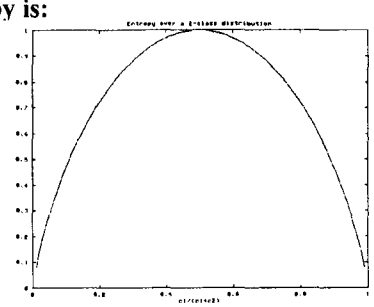
- **The tree structure depends upon:**
  - The split criterion (how to split)
  - The stop criterion (when to stop)
- **Different choices produce different trees**
- **A good tree should have**
  - Few levels. Better to classify with as less decisions as possible.
  - Large leaves population. Better leaves with as many cases as possible.
- **Generation of all possible trees to pick the best one is NP-complete: about $4x10^6$ possible different trees for our example**
- **Different tree induction algorithms vary according to the heuristics they adopt to build the tree**

Split Criterion - Entropy

- **Deciding how to split a set of events**
- **Which variable to split upon? Which function to apply?**
- **Based on the 'Entropy' (or impurity) measure. Intuitively, entropy is 0 (minimum) when all the cases in a set belong to the same class, whereas entropy is 1 (maximum) when each class is equally distributed in the given set.**
- **The general formula for the entropy is:**

$$Entropy = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

n = number of classes
p_i = probability of the class i



Stop Criterion

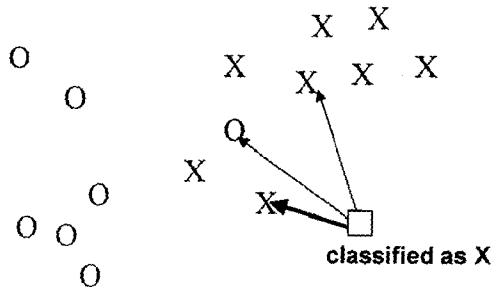
- **In the previous example we stopped when the entropy was 0 (same class for all cases): *Overfitting***
- **This can yield to very deep trees with few cases on the leaf nodes which is undesirable**
- **We can decide to stop the algorithm execution by, for instance:**
  - Define a minimum popularity allowed for the leaves
  - Define a certain entropy value to be reached
  - Or better, a combination of the two
- **Usually: Overfit first, then pruning**
- **Split and Stop criteria makes dramatic difference in terms of tree structure**

Decision Trees Summary

- **Decision Trees have been pretty successful so far**
- **Easy implementation. They can be found on the internet as public domain software**
- **Deal fairly with large databases**
- **Easy to be interpreted as knowledge validation process**
- **Reduced expressive power. Cannot represent interactions between variables**
- **Evaluation function may not be appropriate under certain circumpstances**
- **Cannot easily be integrated with external knowledge**
- **Can be converted to other knowledge representation forms (e.g. rules)**

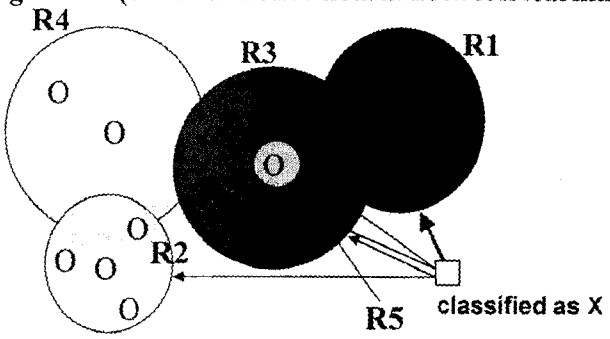
Algorithms -- Nearest Neighbor

- Simple classifier that chooses the class of nearest neighbor(s)
- No training is necessary, but testing is expensive

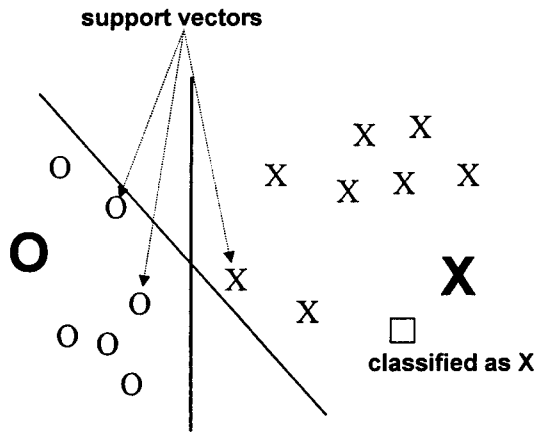


Algorithms -- DistAl

- Inter-pattern distance-based supervised clustering and nearest neighbor algorithm (or constructive neural network learning alg.)



Algorithms – Support Vector Machines



Algorithms -- TFIDF

- **Used a lot in information retrieval**
- **Based on**
  - Rocchio relevance feedback algorithm
  - Term frequency and inverse document frequency word weights
    TF(w) * log(|D|/DF(w))
- **Given set of classes *C* and a document *d***

$$\vec{c} = \sum_{d \in c} \vec{d}$$

$$\arg\max_{c \in C} \cos(\vec{d}, \vec{c}) = \arg\max_{c \in C} \frac{\vec{d} \cdot \vec{c}}{\| d \| \cdot \| c \|}$$

Algorithms -- Naïve Bayes

- **Find a class *c* with the highest conditional probability for data *d***

$$\arg\max_{c \in C} P(c \mid d)$$

- **Bayes rule**

$$P(c \mid d) = \frac{P(d \mid c) \cdot P(c)}{P(d)}$$

- **Independence assumption among features**

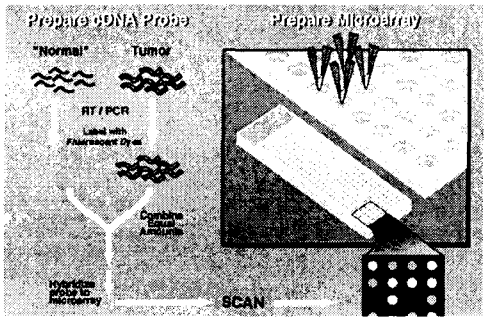
$$P(d \mid c) = \prod_{i=1}^{|d|} P(d_i \mid c)$$

Motivations for Bioinformatics

- **High throughput technologies in genomics, proteomics and drug screening are creating large, complex datasets**
- **Bioinformatics datasets are typically underdetermined**
  - very large number of features (complex domain)
  - small number of instances (high cost per data point)
- **Multi-relational nature of data**
  - reflect complex interactions between molecules, pathways and systems
  - hierarchical organization of interacting layers
- **Current tools and approaches do not adequately address the challenge**

Genomics

- **Data preparation: gene expression by microarray**



- **Mining**
  - sequence analysis
  - clustering
  - supervised learning for drugs, diseases, toxic agents, etc.

Genomics

- **Static**
- **Simple compound of little complexity**
- **Easier to identify**
- **Easier to manipulate**
- **Relation to function not established**
- **Can serve as drug target for particular cases**

Proteomics

- **Data preparation: protein expression**
  - 2-D gels
  - protein arrays
  - mass spectrometry
- **Similar mining tasks as genomics but with protein expression after protein identification**
- **More challenging but worthwhile**
  - dynamic
  - very complex due to post translational modification
  - identification and manipulation more difficult
  - good correlation with function
  - target for drug action for a wide variety of compounds

Sample Task 1: Description

- **Drugs are small molecules that achieve desired activity by binding to a target site on a receptor**
- **Drug discovery process:**
  - identify and isolate receptor to which it should bind
  - test many molecules for their activity to bind to the target site
- **Task: determine what separates active (binding) compounds from inactive (non-binding) ones**

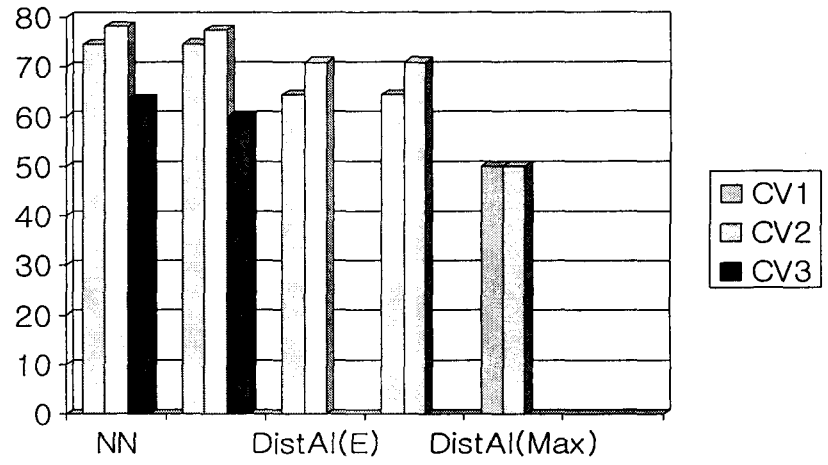
Sample Task 1: Dataset

- **From DuPont Pharmaceuticals**
- **Compounds for binding to a target site on Thrombin (a key receptor in blood clotting)**
- **139351 binary features (describing 3-D properties of molecule)**
- **1909 compounds (number of patterns)**
  - 42 actives
  - 1867 inactives
  - 593 (2 actives, 591 inactives) all zero patterns
- **Evaluation on 634 test patterns**

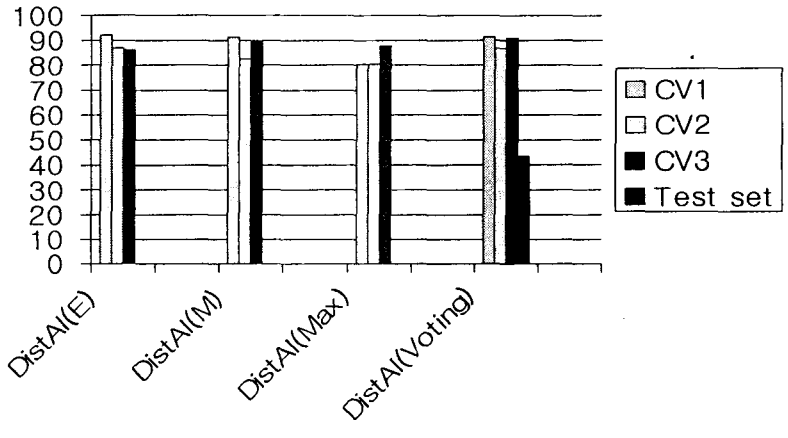
Sample Task 1: Dataset

- **Challenges**
  - large number of features
  - unbalanced
  - cost-sensitive learning
- **Feature subset selection**
  - domain knowledge: agglomeration of 1000 bits generating a reduced feature set of 140 features
  - *genetic algorithm: maximizing classification accuracy of a TFIDF classifier*

Sample Task 1: Results -- various algorithms with original data (CV3 not experimented with DistAl)



Sample Task 1: Results -- DistAl with no zero rows, balanced, real-values



Sample Task 1: Summary

- **Challenging**
  - unbalanced
  - cost-sensitive
  - mislabeled (or multiple labeled) patterns
- **NN and C4.5 did not exhibit difference between**
  - original vs. balanced
  - with vs without mislabeled
  - binary vs. real
- **DistAl**
  - comparable or slightly worse than NN and C4.5 with original patterns
  - outperformed NN and C4.5 with modified data

Sample Task 2: Description

- **Genes code for proteins, which tend to localize in various parts of cells and interact with one another to perform crucial functions**
- **Yeast genome dataset**
  - data on protein-protein interactions from MIPS database (Munich Information Centre for Protein Sequences)
  - expression profiles: DeRise *et al.* (1997), Science 278:680
- **Relational dataset**
  - gene information
  - interaction information
- **Tasks**
  - predict localizations (unique) [15: cell wall, nucleus, mitochondria, etc.]
  - *predict functions (multiple) [14: metabolism, cell growth/division and DNA synthesis, etc.]*

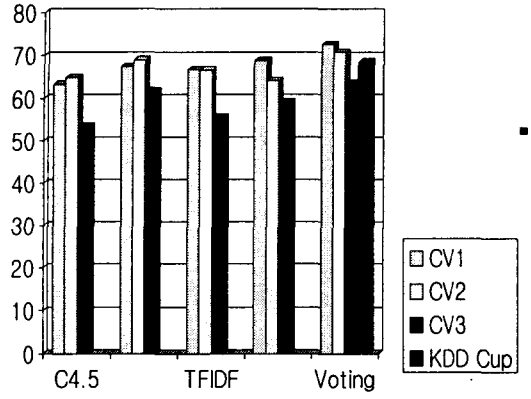
Sample Task 2: Dataset

- **458 original features**
  - essential (1: essential, non-essential, ambiguous)
  - class (24: Actins, ATPases, Cyclins, etc)
  - complex (56: Acetolactate synthase, etc.)
  - phenotype (11: nucleic acid metabolism defects, etc.)
  - motif (351: PS#)
  - chromosome (1: [1..16])
  - number of interacting functions (14)
  - interacting gene type and correlation are ignored (1243)
- **18 additional features**
  - number of interacting localizations (15)
  - number of interacting types (3: physical, genetic, genetic-physical)

Sample Task 2: Dataset

- **Feature subset selection**
  - Different combinations of features were tested
  - Use of all features was the best
- ***Challenges***
  - *Relational learning*
  - *Multiple classification*

Sample Task 2: Results

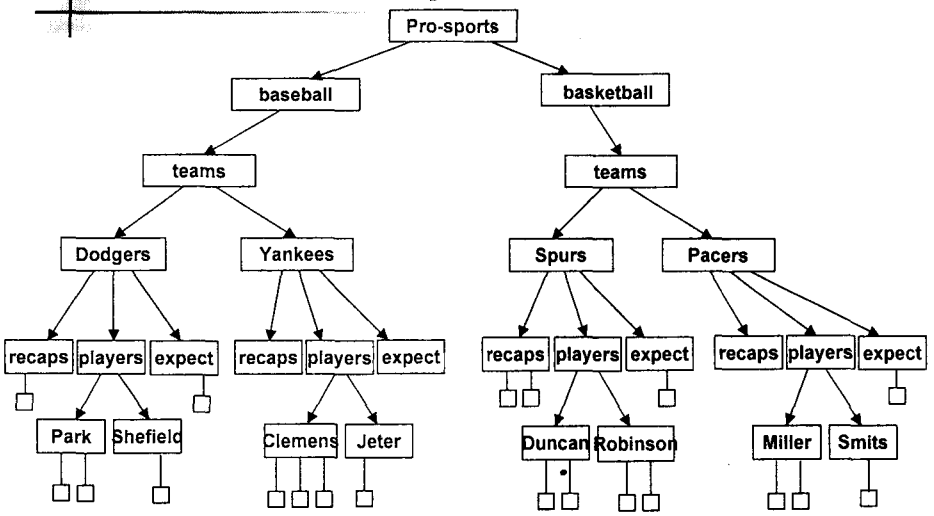


- **Voting**
  - algorithms (in Rainbow)
    - NB
    - TFIDF
    - SVM
  - preference is given to NB

Sample Task 2: Summary

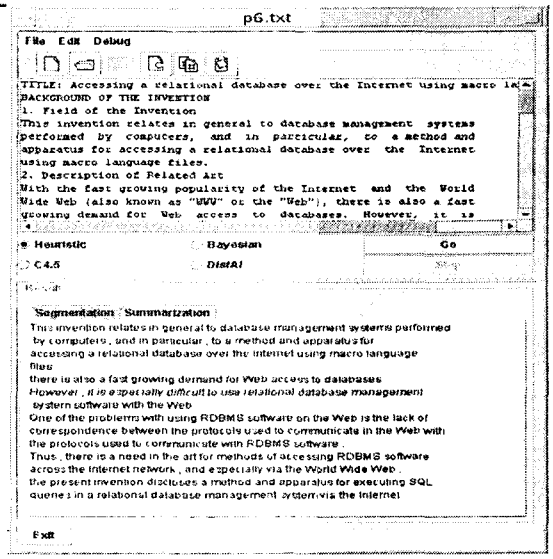
- **Naïve bayes yielded good performance**
- **Voting improved performance**
- **Voting with preference to Naïve bayes was the best**
- **Relation learning can be applied beyond simple inductive learning**
  - probabilistic relational modeling
  - inductive logic programming
  - link analysis
  - …

Sample Concept Hierarchy in Professional sports



cf. non-hierarchical email classification

Text Summarization



cf. email summarization with entire sentences

Intelligent Metadata Extraction

- **Idea: automatic extraction of metadata from a paper/patent prior to mining**
- **approach:**
  - papers in Postscript
  - rule-based system in GCLIPS
  - successful extraction with high accuracy for 100 scientific papers

Field	Accuracy (%)
Title	92
Authors	87
Affiliations	75
Author-Affiliation match	71
Table of contents	76

Text Classification Using a Hierarchical Classifier

Asok Tiyyagura and Jihoon Yang
Information Sciences Lab, HRL Laboratories, LLC

1. Motivation and System Architecture

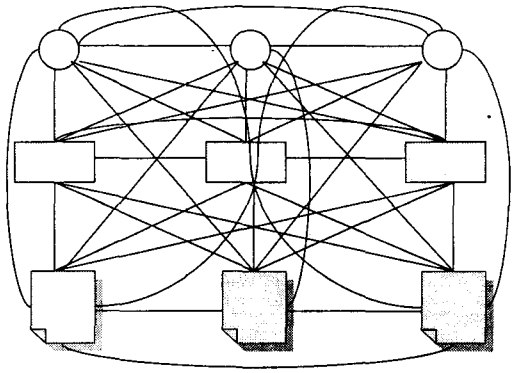
As the amount of on-line data increases astronomically, the design of an efficient algorithm or an approach to access the data has become of great interest and importance. A hierarchical classifier is a good candidate

FILE: hier99
TITLE: Text Classification Using a Hierarchical Classifier
Association Rules
AUTHOR: Asok Tiyyagura (7)
AUTHOR: Jihoon Yang (7)
AFFILIATION 7: Information Sciences Lab, HRL Laboratories,LLC
---Table of Contents---
1 Motivation and System Architecture
2 Concept Hierarchy
3 Machine Learning for Text Classification in a Concept Hierarchy
5 Experimental Results
6 Future Work

Sequence Learning

- **Find frequently occurring sequence (e.g. click stream analysis, ID)**
- **Intrusion detection system (snort alerts)**
  - source IP
  - destination IP
  - source port
  - destination port
  - protocol
  - message (alert) type
- **Previous work**
  - Frequent episode algorithm (Mannila *et al.*, 1997)
  - GSP (Srikant and Agrawal, 1996)
  - SPADE (Zaki, 2000)
- **A variant of SPADE algorithm has been implemented and being experimented**

Bio Digital Library



- Get global knowledge by relation learning
- Additional entities can be added (e.g. drugs, diseases, toxic agents, etc.)

Conclusions

- **KDD/DM as a mean to the design of learning agents, IT IS NOT the solution by itself (just another powerful tool!)**
- **Still incomplete, no standards, no languages, mixed terminology**
- **Human expertise plays a fundamental role for:**
  - Understanding the problem
  - Conceive a data mining solution
  - Choose the right data
  - Choose the right tools/algorithms
- **Just at the beginning of a long lasting problem**
- **Basis for solving lots of significant problems**