Application of Natural Induction and Conceptual Clustering to Pattern Discovery in Data

Examples of Application to Medicine, Bioinformatics, Agriculture, Volcanology, Manufacturing, Demographics, User Modeling and Intrusion Detection, and Tax Fraud Detection

Machine Learning and Inference Laboratory
George Mason University
(www.mli.gmu.edu)

Support: National Science Foundation and NSA

Research Team: R.S. Michalski (PI), K. Kaufman,
J. Wojtusiak, J. Pietrzykowski, S. Mitchell, W. Seeman

Major Research Projects in the Machine Learning and Inference Laboratory

(www.mli.gmu.edu)

- Natural Induction (AQ21)
- Conceptual Clustering (CLUSTER 3)
- Learnable Evolutionary Computation (LEM3)
- Learnable Data Bases and Knowledge Scouts (VINLEN)
- Plausible Reasoning and Dynamic Recognition
- Areas of Applications: Medicine, Bioinformatics, Agriculture, Volcanology, Manufacturing, Demographics, User Modeling and Intrusion Detection, World Demographics, Tax Fraud Detection, Heat Exchanger Optimization

This presentation focuses on selected examples of application of natural induction and conceptual clustering

Natural Induction

- Natural induction hypothesizes general concept descriptions from concept examples and discovers strong patterns in data, expressing them in the forms easy to understand and interpret by people, such as natural language-like descriptions and graphical visualizations. Such forms are natural to people because they closely follow forms in which people represent knowledge

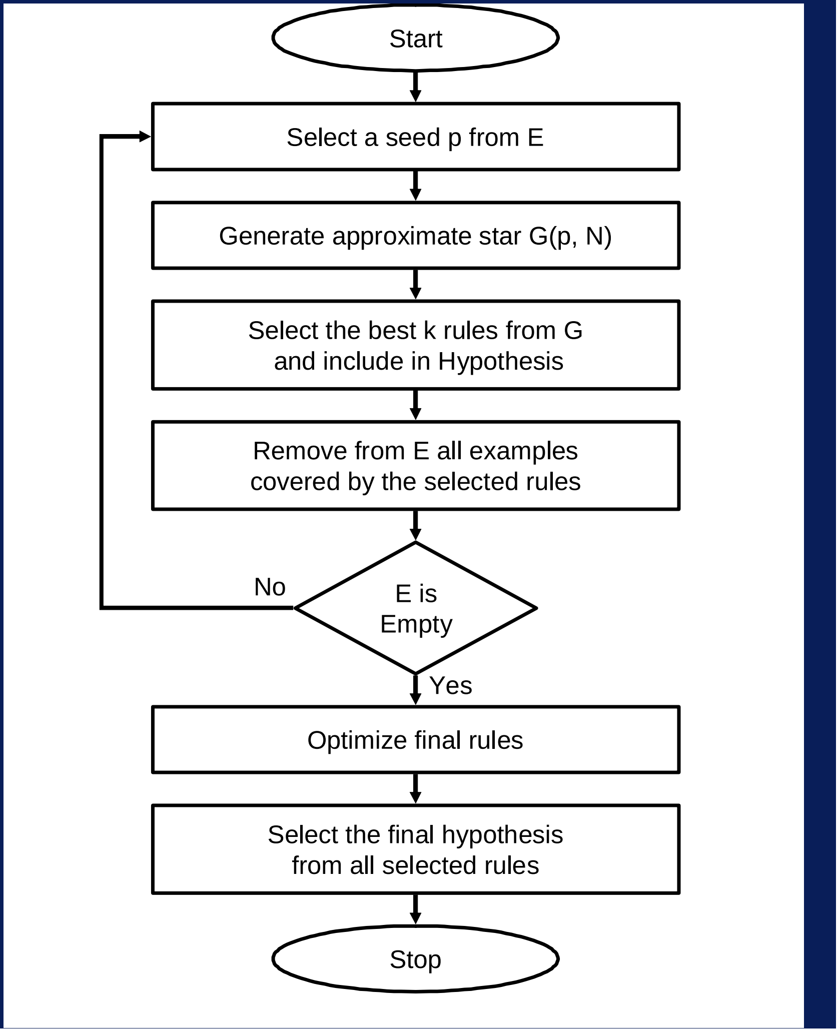
- Natural induction puts an equal emphasis on predictive accuracy and the understandability of computer-generated knowledge, in contrast to conventional methods of machine learning that are primarily concerned with predictive accuracy.

AQ21: A Laboratory for Natural Induction

Major features of AQ21:

- Learns concepts from examples or discovers patterns in data
- Operates in three modes:
  Theory Formation (TF), Approximate Theory Formation (ATF), and Pattern Discovery (PD)
- Learned rules may include exception clauses
- Learns alternative hypotheses
- Handles meta-values (?, NA, *)
- Able to improve the representation space
- Contains a module for rule testing and application that works in two modes- strict and flexible, and can be apply to both static data and temporal data streams.

Simple version of AQ21 in PD mode



Start
Select a seed p from E
Generate approximate star G(p, N)
Select the best k rules from G and include in Hypothesis
Remove from E all examples covered by the selected rules
E is Empty
No
Yes
Optimize final rules
Select the final hypothesis from all selected rules
Stop

AQ21 Learns Rules Expressed in Attributional Calculus

- AQ21 represents concepts or patterns in the form of sets of *attributional rules* with the same consequent:

$A <= B_1$ (Rule 1)

$<= B_2$ (Rule 2)

……… ……….

$<= B_k$ (Rule k)

.

where A, $B_{1,}$ B_2 , .., B_k are conjunctions of *attributional conditions,* such as:

[Blood-type = 0], [Weight > 200 Lb]
[Color = red or blue or green], [X = 2..8]
[X1 > X2], [Size & Width ≤ 1m],
[Count{X1,X2,X4,X6 EQ 2} ≥ 3]
[Weather: warm & sunny]

- Due to the ability to employ of such conditions, attributional rules are more expressive than conventional decision rules that use only *<attribute-rel-value>* conditions. As a consequence, AQ21 may be able to learn rules or patterns that other programs can not.

Examples of Attributional Rules

The simplest form:

[Activity=running_experiments]
<= [Day = weekend] & [Clock_speed >= 2GHz] &
[Location = lab1 v lab3] & [Weather: quiet & warm]

Rules with an exception clause and annotations:

[Activity=play]
<= [Condition=cloudy v sunny: 7,8] & [Temp=med v high: 7,7]
|_ [Wind] & [Condition=cloudy] & [Temp=high]: p=7,n=0, Q=1

where

|_ is an exception operator

a pair of numbers in each condition denotes numbers of positive and negative examples covered by the condition

p and n are numbers of positive and negative training examples covered, respectively

Q is a measure rule quality defined in the next slide.

A Measure of Rule Quality

The rule (or pattern) quality measure, Q(w), is defined by:

$$Q(w) = Cov^{w} * Consig^{1-w}$$

where

$Cov = p/P$ ("Coverage")

$Consig = ((p / (p + n)) – (P /(P + N))) * (P +N) /N$ ("Consistency gain")

where p and n are numbers of positive and negative examples covered by the rule, and P and N are total numbers of positive and negative examples in the data, respectively.

An Example of Application to Medicine

Determining Relationships between Lifestyles and Diseases

- Database from American Cancer Society contains records of responses to surveys of non-smoking males, aged 50-65, regarding lifestyle and disease history
  - Each patient is described in terms of 32 attributes: 7 lifestyle attributes (2 Boolean, 2 numeric, and 3 rank) and 25 Boolean attributes denoting diseases
  - Dataset contains 73,553 records
- VINLEN was applied to discover attributional rules characterizing dependency of 25 diseases on lifestyles and other diseases
- Discovered rules were used to generate concept association graphs (CAGs) to represent dependencies visually and more abstractly
- In a CAG, the thickness of a link reflects some characteristic of the condition, e.g., its relative support or confidence, and the link annotation (+, –, v, ^) indicate the type of the relationship between condition and consequent.

Lifestyle Attributes in the Data

- Rotundity (very_low .. very_high)
- Exercise (none, slight, moderate, heavy)
- Sleep (nightly, in hours)
- Y_i_n (years living in the same neighborhood)
- Education (8th grade or less, some hs, hs grad, vocational, some college, college grad, grad degree)
- Mouthwash (yes/no)
- Veteran (yes/no)

Examples of Strong Patterns Discovered
(for Arthritis and Colon Polyps)

[Arthritis = Present]

<= [HBP=present: 432, 1765] &
[Education<=college_grad: 940, 4529] &
[Rotundity>=low: 1070, 5578] &
[Y_i_n>0: 1109, neg:5910]: p = 325, n = 1156;
P = 1171, N = 6240

[Colon Polyps = Present]

<= [Prostate Disease=present: 34, 967] &
[Sleep=5,9: 16, 515] &
[Rotundity=average: 58, 2693] &
[Education<=some_college: 81,4146]: p = 5, n = 0 ;
P = 147, N = 7383

Explanation:

The two numbers within each condition denote the number of positive and negative examples covered, respectively
p, n, -- numbers of positive and negative examples covered by the rule
P, N – numbers of positive and negative examples in the training data for that class (concept)

Examples of Strong Patterns Discovered
(for Diverticulosis and Rectal Polyps)

[Diverticulosis = present]
<= [Arthritis=present: 70,1033] &
[Rotundity>=average: pos:170, neg:4202] &
[Education>=some_college: 176, 4412]&
[Stroke=Absent: 257, 7037] &
[Sleep=7..9: 205, 574] &
[Y_i_n>10: 134, 3836]: p:24, n:115;
P = 262, N = 7117

[Rectal Polyps = present]
<= [Prostate Disease=present: 73, neg:893] &
[mouthwash=yes: 194, 3509] &
[education>=some_hs: 252, 5246] &
[Y_i_n=2..63: 296, 6173] &
[rotundity ≠ high: 275, 5967]: p:38, n:271;
P = 334, N = 6951

Examples of Strong Patterns Discovered
(for Stomach Ulcer, Astma and Hay Fever)

[Stomach Ulcer=Present]

<= [Arthritis=Present:107,1041] &
[Education<=college_grad: 305, 5276] &
[Exercise>=medium: 298, 5606]: p =79, n =668;

P = 367, N = 7108

[Asthma=Present]

<= [Hay Fever=Present: 170, 787]: p = 170, n = 787

P = 331, N = 7047

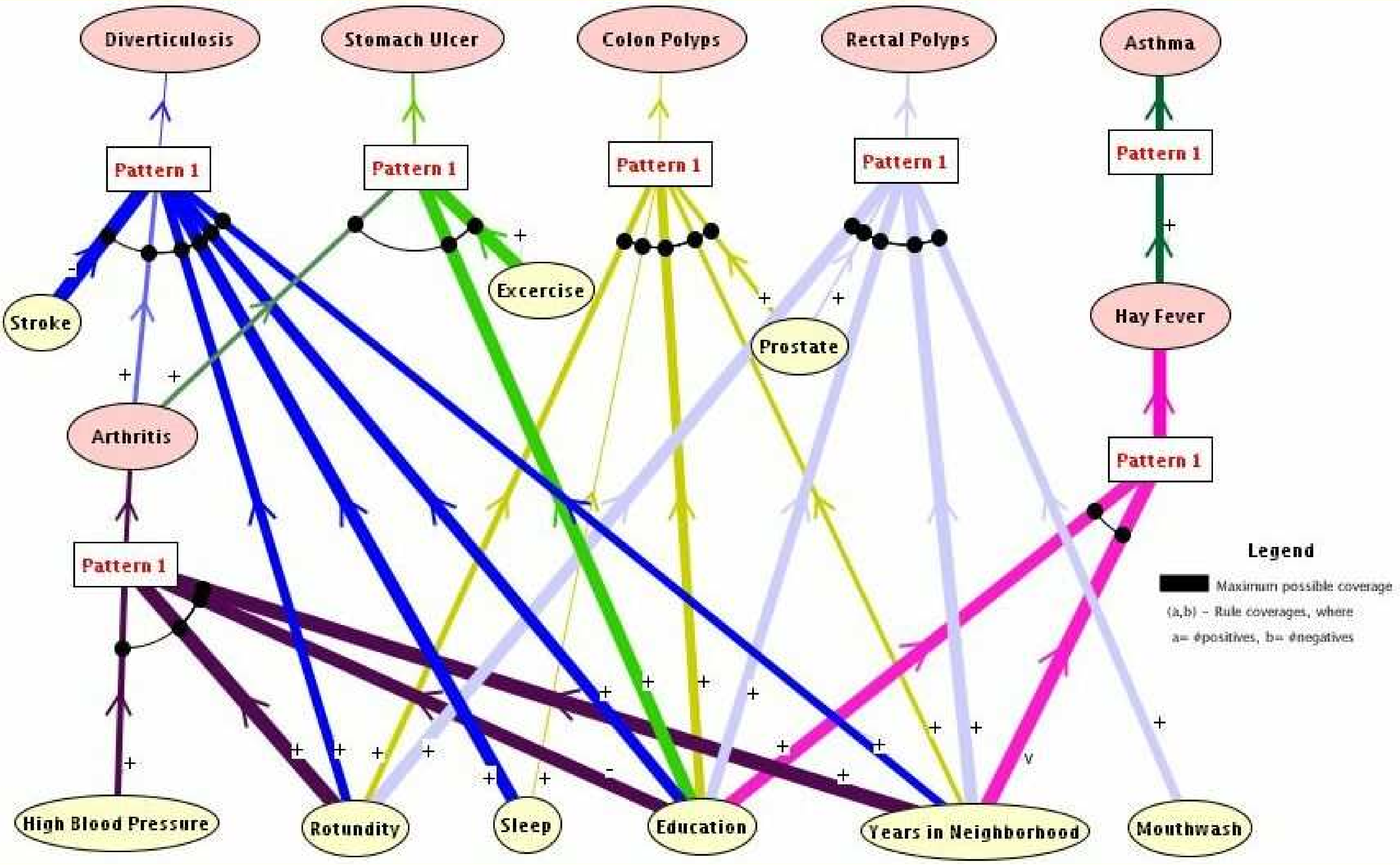
[Hay Fever=Present]

<= [education>=vocational: 772, 4231]
[y_i_n>0: 939, 6073]: p=763, n=4141;

P = 965, N = 6304

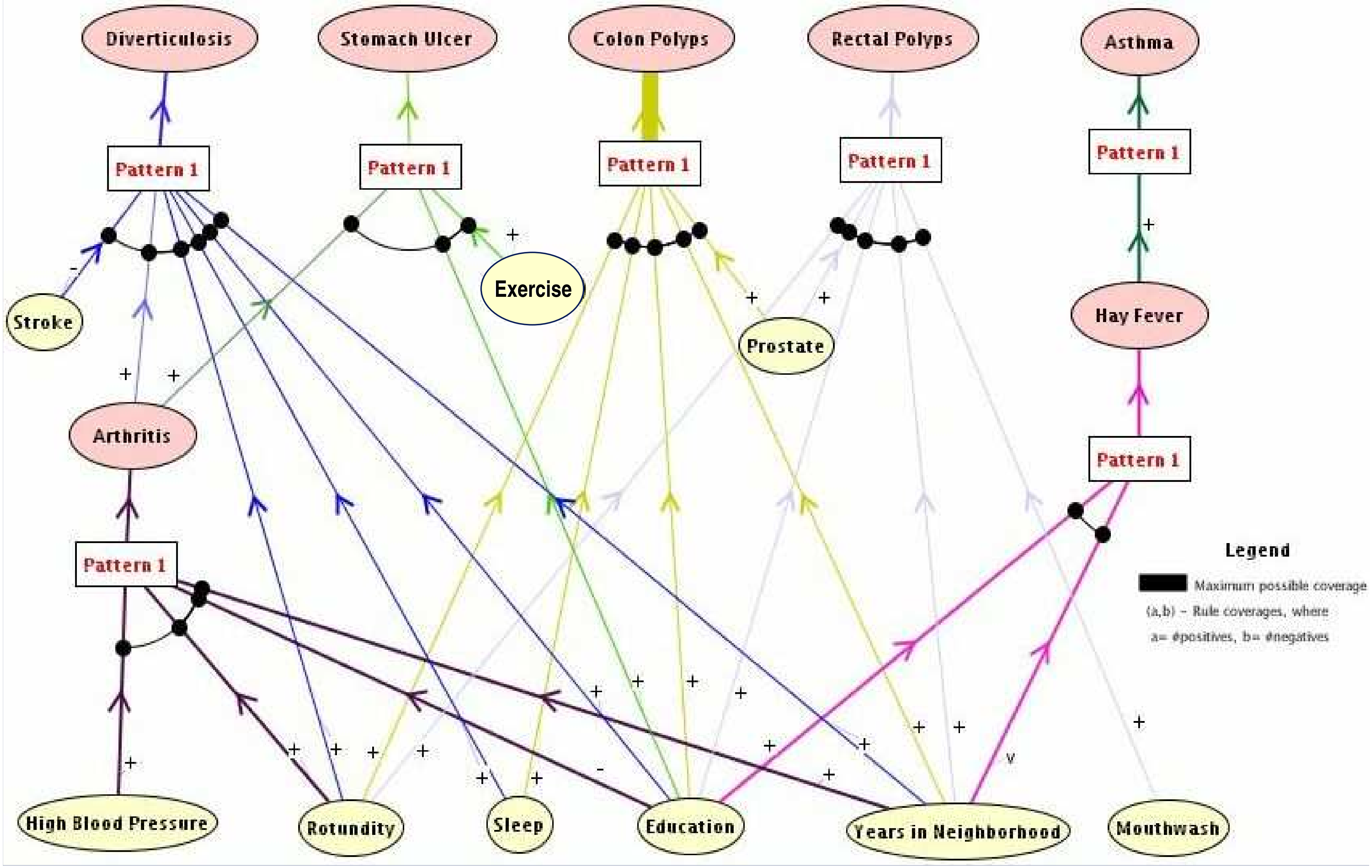
Concept Association Graph Representing Discovered Patterns

Link’s Thickness Represents Relative Support (p/P)



Concept Association Graph Representing Discovered Patterns

Link’s Thickness Represents Consistency p/(p+n)



An Example of Application to Bioinformatics

Discovering Rules For Diagnosing Metastatic MedulloblastomanTumors from Gene Arrays

Medulloblastoma is a highly invasive primitive neuroectodermal tumor of the cerebellum

It is the most common malignant brain tumor of childhood.

The research was based on the paper:

T. J. MacDonald , K. Brown, B. LaFleur, K. Paterson, C. Lawlor, Y. Chen, R. Packer, P. Cogen, & D. Stephan, “Expression profiling of medulloblastoma: PDGFRA and the RAS/MAPK pathway as therapeutic targets for metastatic disease,” *Nature Genetics*, Vol. 29, pp. 143-152, October 2001.

Data

Gene Expression Omnibus (GEO), NCBI NLM NIH

http://www.ncbi.nlm.nih.gov/geo

Nucleic Acids Res. 2002 Jan 1;30(1):207-10

Last updated 06/26/2003

GDS232

# attributes (probes): 2059

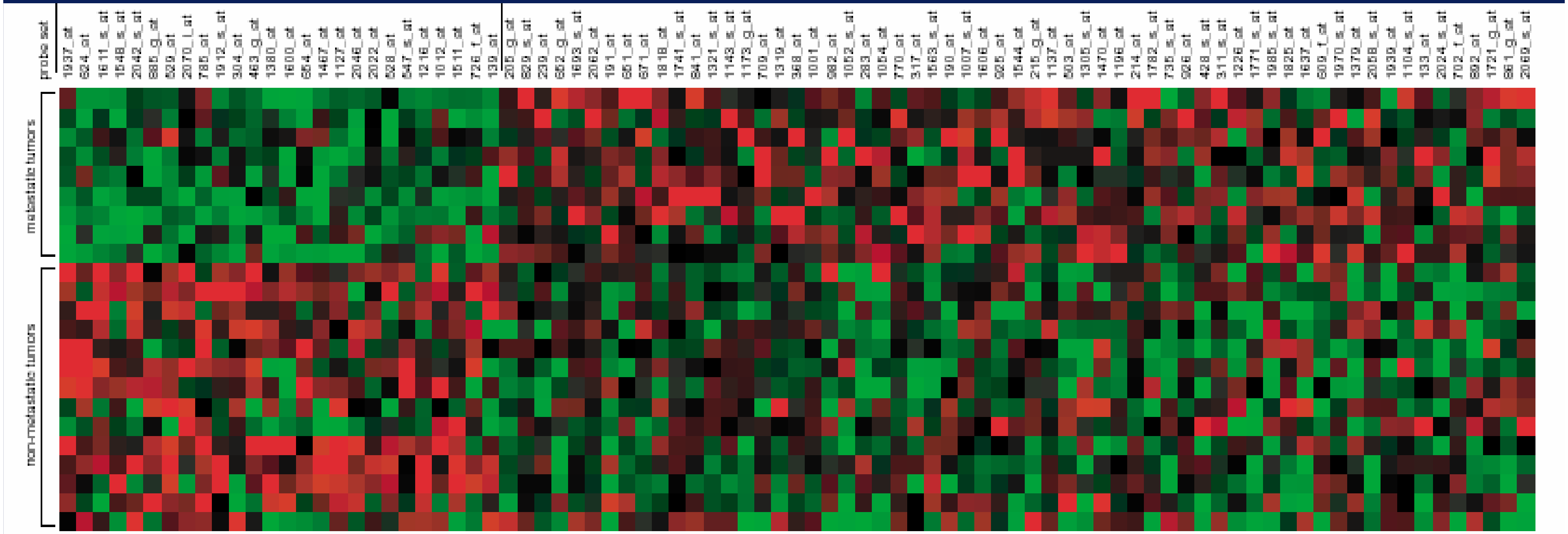
- real values
- maximum = 184262
- minimum = -13888.4

# events (patients): 46

- 20 metastatic (M1)
- 26 non-metastatic (M0)

Gene Array with Initially Selected 87 Attributes

Based on Tobey MacDonald, op. cit.



Attributes were selected using Prediction Strength Correlation which is a ratio of difference between mean value in class 1 and in 2 over the difference between standard deviation in class 1 and in 2. McDonald et al. selected 87 attributes shown below in columns (subset of samples in rows). The first 9 samples are metastatic, the next 14 non-metastatic. Bright red color means high expression levels, bright green – the opposite.

Attribute Selected by the PROMISE Criterion

The 10 attributes with highest PROMISE value selected from among 2059:

Attribute name:	Promise value:
Gene-1343_s_at:	1
Gene-1723_g_at:	1
Gene-1036_at:	1
Gene-532_at:	1
Gene-481_at:	1
Gene-1216_at:	0.98419
Gene-1611_s_at:	0.983871
Gene-914_g_at:	0.983539
Gene-1509_at:	0.983193
Gene-1783_at:	0.979167

Data Projected on 10 Selected Real-Valued Attributes

Class M0 (non-Metastatic): 16 unique events (26 total)

```
Attributes: Disease,  G1343_s_at, G1783_at, G1723_g_at, G1611_s_at, G1509_at, G1036_at, G914_g_at, G532_at, G481_at, G1216_at, Frequency
M0,   7.588,   63.23,   -90.78,   110.2,   348.5,   -5.514,   -4.387,   -13.52,   -1.799,   153.1,   2
M0,   20.79,   -52.27,   -18.56,   158.3,   -74.73,   -103.4,   107.2,   -22.78,   -13.78,   181,   1
M0,   22.92,   -161.4,   -58.53,   199,   100.8,   19.77,   46.86,   -56.88,   26.59,   -67.03,   2
M0,   24.27,   -61,   -21.66,   184.7,   -87.22,   -120.6,   125.1,   -26.58,   -16.07,   211.3,   1
M0,   27,   83.22,   54.02,   117.6,   124.3,   59.93,   -37.98,   65.02,   35.28,   -7.383,   2
M0,   36.53,   -172.8,   97.77,   41.17,   10.59,   75.58,   125.2,   -24.87,   -15.98,   -23.04,   2
M0,   45.71,   -81.69,   -144.2,   139.4,   -21.39,   -37.48,   46.66,   145.7,   80.6,   -97.5,   2
M0,   47.57,   -133.7,   -131.6,   78.31,   69.25,   233.9,   78.73,   6.702,   27.34,   54.23,   2
M0,   98.22,   -8.824,   -1.473,   170.8,   348.2,   84.24,   160.9,   119.8,   96.59,   751.9,   2
M0,   -10.14,   54.16,   -30.22,   122.6,   225.3,   21.75,   124,   23.14,   25.65,   115.5,   2
M0,   -12.98,   -167.3,   36.52,   -38.15,   95.94,   96.82,   163.7,   -43.17,   -62.3,   257,   1
M0,   -13.59,   -175.2,   38.23,   -39.94,   100.4,   101.4,   171.4,   -45.2,   -65.23,   269,   1
M0,   -68.06,   90.42,   -207.1,   38.61,   -49.67,   -50.58,   81.99,   -89.47,   -159.9,   428.8,   2
M0,   111.1,   60.52,   -85.43,   181.3,   -4.413,   65.86,   -3.385,   50.36,   112.5,   184.7,   2
M0,   317.2,   -536.1,   -2692,   118.3,   482.2,   108,   599.7,   2392,   839.9,   95.52,   1
M0,   354.9,   -599.8,   -3011,   132.4,   539.4,   120.8,   670.9,   2676,   939.7,   106.9,   1
```

Class M1 (Metastatic): 12 unique events (20 total)

```
M1,   11.87,   -28.51,   79.68,   22.11,   -196.2,   -32.2,   -113.2,   -66.83,   -96.39,   -59.77,   1
M1,   12.14,   -29.17,   81.53,   22.63,   -200.8,   -32.95,   -115.8,   -68.39,   -98.64,   -61.16,   1
M1,   31.29,   -16.33,   -48.11,   43.53,   74.94,   -6.838,   30.69,   -26.16,   -28.54,   -7.781,   2
M1,   35.53,   -40.8,   36.1,   43.66,   -81.86,   55.72,   64.9,   -6.184,   60.84,   -17.54,   2
M1,   41.15,   -29.18,   -24.94,   91.61,   20.24,   73.39,   117.9,   37.85,   37.04,   -3.72,   2
M1,   41.62,   -35.32,   -14.15,   45.8,   72.12,   13.74,   34.59,   68.18,   95.99,   -43.16,   2
M1,   42.51,   102.8,   198,   72.33,   -23.29,   87.75,   37.5,   -36.8,   -40,   -43.55,   2
M1,   51.63,   147.2,   -82.33,   -31.76,   27.5,   -52.63,   -40.02,   -8.407,   7.981,   154.9,   1
M1,   51.86,   147.8,   -82.69,   -31.9,   27.62,   -52.85,   -40.19,   -8.444,   8.015,   155.6,   1
M1,   78.17,   113.6,   0.628,   163,   232.2,   -25.65,   86.23,   60.46,   93.59,   52.87,   2
M1,   -38.98,   4.672,   -33.52,   -9.136,   37.1,   63.07,   84.88,   -95.57,   36.13,   -147.5,   2
M1,   -78.46,   -251.4,   162.4,   -45.9,   -40.36,   25.93,   96.31,   68.2,   -39.62,   65.81,   2

Min values:  -78.46,   -599.8,   -3011,   -45.9,   -200.8,   -120.6,   -115.8,   -95.57,   -159.9,   -147.5
Max values:    354.9,   147.8,      198,      199,    539.4,     233.9,     670.9,     2676,      939.7,    751.9
Range:          433.36,   747.6,    3209,   244.9,   740.2,     354.5,     786.7,     2771.57, 1099.6,  899.4
```

Rules Discovered by AQ21 for Detecting the Metastatic Cancer

(Without discretizing the original continuous attributes)

[Cancer = metastatic]
<= [Gene-1611_s_at <= 100.9: 18, 8, 69%, 18, 8, 69%] &
[Gene-1036_at = -41.76..160.8: 18, 20, 47%, 16, 4, 80%] &
[Gene-914_g_at <= 121.5: 20, 15, 57%, 16, 0, 100%]
#positives = 16, #new positives = 16, #unique = 14, complexity = 17 (Rule 1)

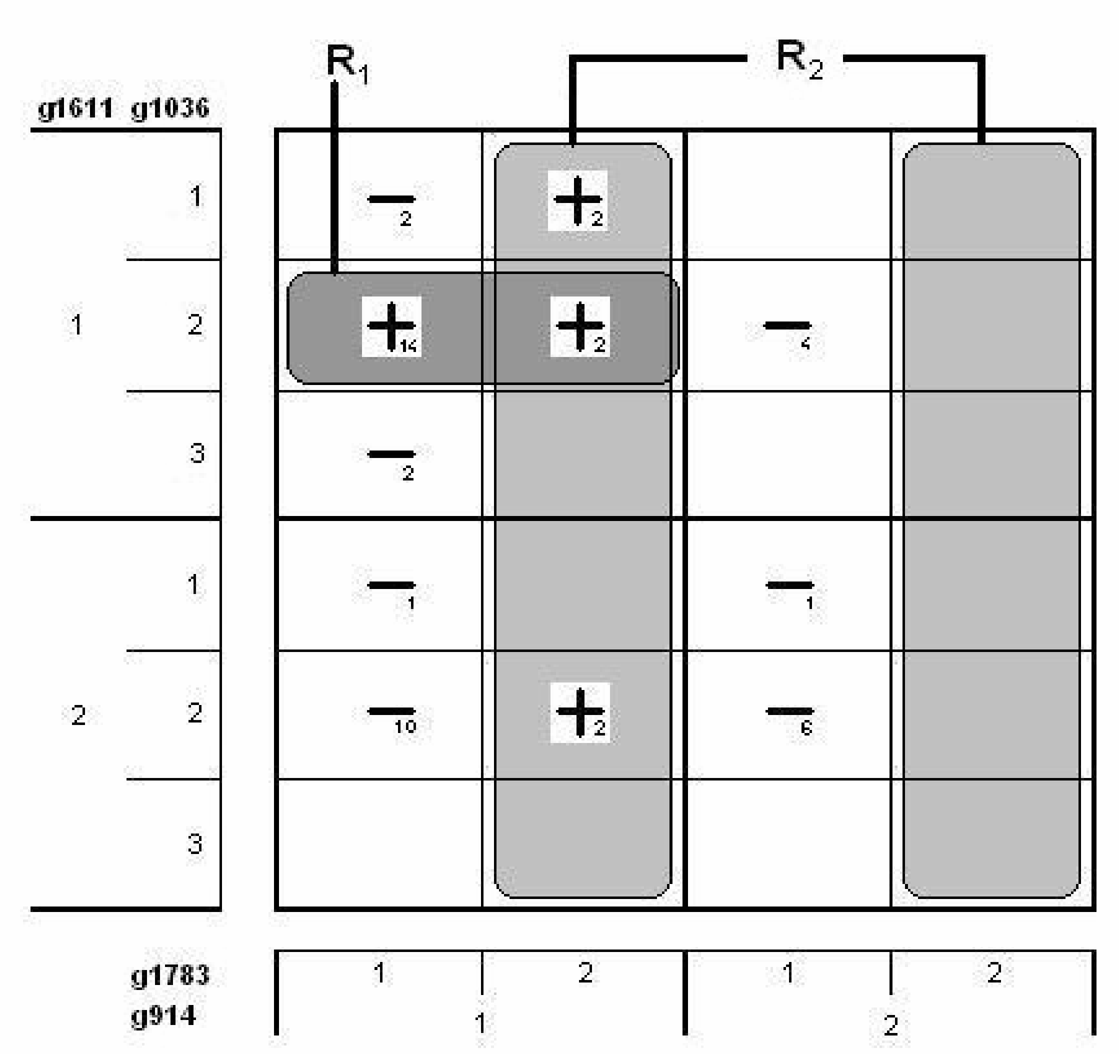
<= [Gene-1783_at >= 96.6: 6,0,100%,6,0,100%]
#positives = 6, #new positives = 4, #unique = 4, complexity = 5 (Rule 2)

Explanation:

The third number each triple in every condition indicates p/p+n%. The second triple indicates p, n and p/p+n% in the context of previous conditions.

Application of these rules to the learning and to the testing data gave 100% accuracy
(this is a highly unusual case; the rules need to be tested on more data to confirm or disconfirm their validity).

Visualization of Discretized Training Data and Learned Rules in a General Logic Diagram



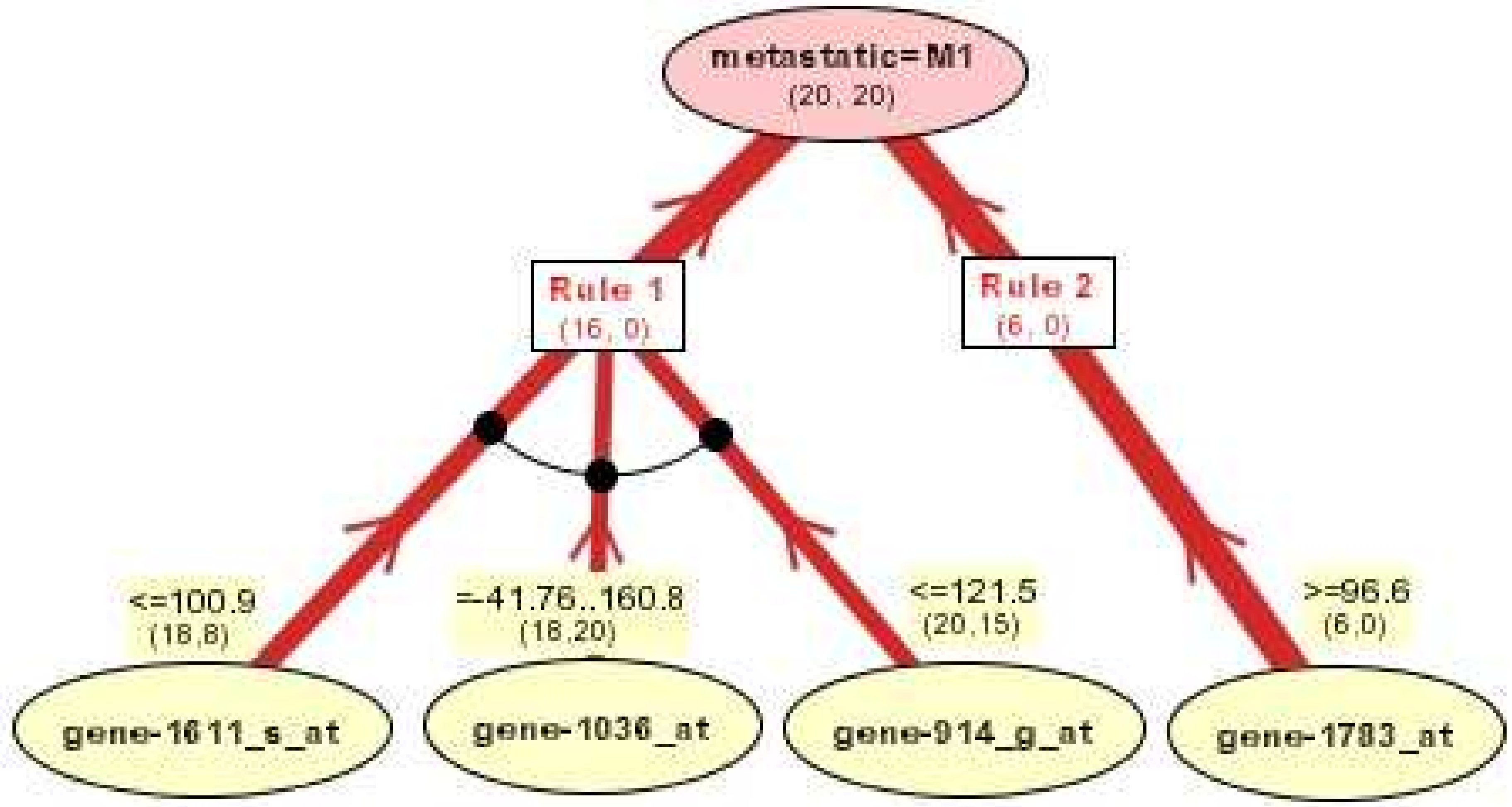
"+" = metastatic examples (M1)

"-" = non-metastatic examples (M0)

[Cancer = metastatic]
<= [*Gene-1611_s_at <= 1] &
[*Gene-1036_at = 2] &
[*Gene-914_g_at = 1] (Rule R1)
<= [*Gene-1783_at = 2] (Rule R2)

where * indicates that this is a discretized version of the corresponding gene.

Visualization of Rules for Metastatic Cancer as a Concept Association Graph (CAG)



Comments on the Obtained Results

- The rules learned by AQ21 for distinguishing metastatic from non-metastatic tumors in gene arrays have a very high predictive accuracy (~ 95% in a 5-fold cross-validation experiment), but that result was obtained on a very small amount of data. We are searching for additional data to confirm the obtained results
- Rules involve only a few genes (selected from over 2000) and are easy to interpret
- It may be noteworthy that the neural net developed by T. J. MacDonald et al. for the same task involved about 80 genes, and its prediction accuracy was about 72%.

Application to Agriculture

Learning a Rule-Tree Representation for Diagnosing Soybean Diseases

This example illustrates an application of a version of natural induction that outputs a *rule-tree*, a simple structure oriented toward classifying entities into many related classes.

PROBLEM

Learn rules for diagnosing most common soybean diseases (19 diseases)

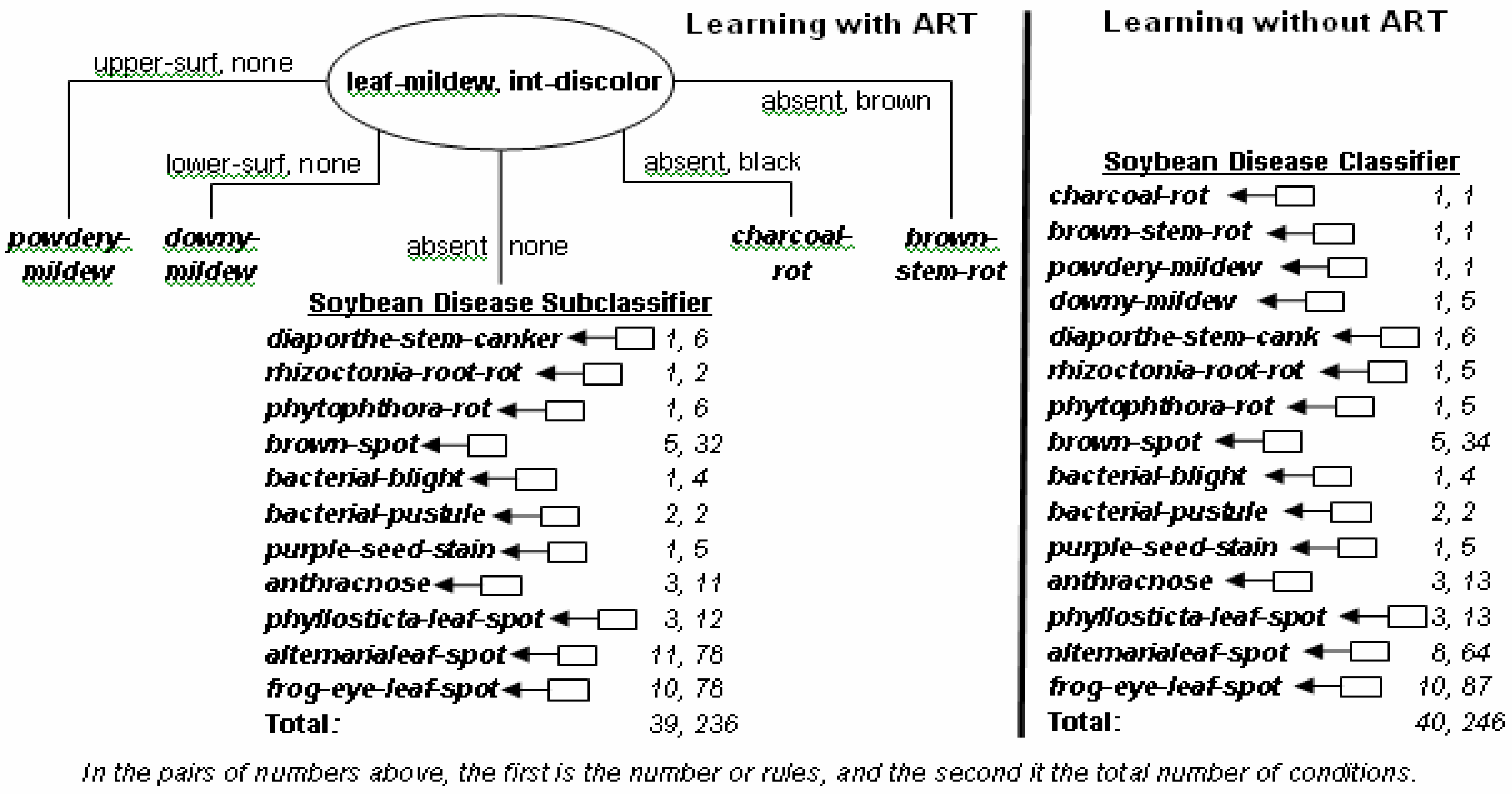
Each case of a disease is described by 35 multi-valued attributes

The training data consists of 266 cases provided by an agricultural expert.

An Application of the ART Program

- ART works in two steps: the first step seeks *partitioning attributes*, and the second step learns rules distinguishing classes within groups determined by the partitioning attributes
- For the soybean disease diagnosis problem, ART found two partitioning attributes, leaf-mildew and int-discolor (internal discoloration), that are assigned to the root node of the Rule-tree and whose combinations of values split all nineteen classes into five logically disjoint subsets
- Different combinations of their values are assigned to the branches stemming from the root
- Four combinations identify individual classes: powdery--mildew, downy-mildew, charcoal-rot and brown-stem-rot, and one combination leads to a subclassifier for the other eleven diseases.
- The next slide presents such a rule-tree and, for comparison, also an equivalent flat ruleset for the nineteen diseases.
- As one can see, the rule-tree representation is simpler and easier to understand.
- It may be worth-noting that the overall learning time of the rule-tree was shorter than learning the flat ruleset representation.

Attributional Rule-Tree v. Flat Ruleset



Soybean Disease Subclassifier

diaporthe-stem-canker ◄─□	1, 6
rhizoctonia-root-rot ◄─□	1, 2
phytophthora-rot ◄─□	1, 6
brown-spot ◄─□	5, 32
bacterial-blight ◄─□	1, 4
bacterial-pustule ◄─□	2, 2
purple-seed-stain ◄─□	1, 5
anthracnose ◄─□	3, 11
phyllosticta-leaf-spot ◄─□	3, 12
alternarialeaf-spot ◄─□	11, 78
frog-eye-leaf-spot ◄─□	10, 78
Total:	39, 236

Learning without ART

Soybean Disease Classifier

charcoal-rot ◄─□	1, 1
brown-stem-rot ◄─□	1, 1
powdery-mildew ◄─□	1, 1
downy-mildew ◄─□	1, 5
diaporthe-stem-cank ◄─□	1, 6
rhizoctonia-root-rot ◄─□	1, 5
phytophthora-rot ◄─□	1, 5
brown-spot ◄─□	5, 34
bacterial-blight ◄─□	1, 4
bacterial-pustule ◄─□	2, 2
purple-seed-stain ◄─□	1, 5
anthracnose ◄─□	3, 13
phyllosticta-leaf-spot ◄─□	3, 13
alternarialeaf-spot ◄─□	8, 64
frog-eye-leaf-spot ◄─□	10, 87
Total:	40, 246

In the pairs of numbers above, the first is the number or rules, and the second it the total number of conditions.

Rules Generated by ART Can Be Interpreted as Rules with "Provided-that" Operator

For example, a rule for rhizoctonia-root-rot learned by ART can be expressed as:

<= [Plant_growth = abnorm] & [Leaves = norm]

Γ [Leaf_mildew = absent] & [Int_discolor = discolor = none]

where Γ denotes the "provided that" operator*.

The rule states that "*The soybean disease is rhizoctonia-root-rot, if the plant's growth is abnormal and its leaves are normal, provided that there is no leaf mildew and no internal discoloration*".

* The "provided that" operator has been introduced in attributional calculus and is useful for representing rules with preconditions (Michalski, 2004; www.mli.gmu.edu/papers)

An Example of Application to Volcanology

Discovering Patterns in the Smithsonian Volcanic Activity Data Base

- Given a database with records of volcanic eruptions if the past 10,000 years, AQ21 sought patterns in those eruptions
- Data consisted of approximately 20,000 records; in each experiment half were randomly selected for training, and the other half for testing
- Each case of eruption is described by 78 attributes of different types (nominal, linear, integer, real, Boolean and structured)
- One experiment was to differentiate between eruptions in which fatalities were known to have occurred and eruptions without fatalities
- Learning utilized both *theory formation* and *pattern discovery* modes
- This research has been being conducted in collaboration with the Smithsonian Institution in Washington, D.C.

Some of the Attributes in the Volcano Database

(Volcano attributes are in white, eruption attributes in yellow)

Name	Type	Description
Subregion	structured	Part of the world in which the volcano is located
Latx, Longx	real	Latitude and longitude of the volcano
Upper	integer	Elevation of the peak (meters)
Upper1	integer	Height of the volcano (meters)
Type	structured	Type of volcano
TC	structured	Tectonic setting of the volcano
MapStatus2	integer	Indicates how long since last erupt (lower=more recent)
Year, Stop_year	integer	Years of eruption start and end, respectively
Radial_fissure	Boolean	Whether there was a radial fissure eruption
Regional_fissure	Boolean	Whether there was a regional fissure eruption
Island_forming	Boolean	Whether the eruption resulted in the creation of an island
Subglacial	Boolean	Whether there was a subglacial eruption
Crater_lake_erupt	Boolean	Whether there was a crater lake eruption
Explosive	Boolean	Whether the eruption was explosive in nature
Pyroclastic	Boolean	Whether the eruption included pyroclastic materials
Lava_lake	Boolean	Whether a lava lake was formed
Damage	Boolean	Whether there was damage to human structures
Lahars	structured	Whether lahars were formed
Tsunami	Boolean	Whether the eruption resulted in a tsunami
Evacuation	Boolean	Whether there were evacuations

Examples of Strong Patterns Discovered by AQ21

[Fatalities = present]

<= [Radial_fissure=present: 72,773] &
[Tsunami=present: 61,29] &
[Latx<=33.99: 343,5782] &
[Stop_year<=1889: 148,934]: p=13, n=0; Q=0.701

[Fatalities = absent]

<= [Pyroclastic=absent: 8244,221]: p=8244,n=228,q=0.443

Example of Consistent Rules Discovered by AQ21

```
[Fatalities = present]
<= [regional_fissure=absent: 431,8913] &
   [explosive=present: 443,7447] &
   [pyroclastic=present: 220,1069] &
   [damage=present: 382,769] &
   [subregion=Mediterranean-WAsia,Java,Alaska-SW,US-Washington,Mexico,Guatemala : 126,1568] &
   [tc=CC : 321,6017] &
   [latx=-23.76..62.97: 427,8031] &
   [longx=-152.8..112.9: 209,4414] &
   [upper=647..4841: 384,8162] &
   [upper1>=825: 335,7017] &
   [type=Lava_Class,Cinder_cones,Explosion_crater,Pyroclastic_cones,Stratovolcano,Tuff_cones: 270,4987] &
   [map_status2<=1: 369,6744] &
   [year=1920..1992: 147,2583] &
   [stop_year>=1982: 81,739]: p=29,n=0
```

Example of Consistent Rules Determined by AQ21

[Fatalities = absent]

```
<= [radial_fissure=absent : 8575,377] &
   [island_forming=absent : 9247,428] &
   [subglacial=absent : 9155,434] &
   [crater_lake_erupt=absent : 8889,388] &
   [lava_lake=absent : 9195,432] &
   [damage=absent : 8564,67] &
   [lahars=absent : 8679,232] &
   [tsunami=absent : 9327,388] &
   [evacuation=absent : 9110,256] &
   [subregion=Germany,France,Greece,Turkey,Georgia,Armenia,Red_Sea,Ethiopia,Africa-C,Africa-W,Africa-N,Middle_East-Indian_Ocean,New_Zealand_to_Fiji,Melanesia-Australia,Andaman_Is,Sumatra,Java,Lesser_Sunda_Is,Sangihe_Is,Phillipines-SE_Asia,N-of-Taiwan,Ryukyu,Volcano_Is,Mariana_Is,Kuril_Is,Kamchatka-Mainland_Asia,Alaska,Colombia,Ecuador,Chile-N,Argentina,Chile-C,Chile-S,W-Indies,Iceland-Arctic,Atlantic-N,Cape_Verde_Is,Atlantic-C,Atlantic-S,Antarctica : 5105,229] &
   [latx<=63.96 : 9034,439] &
   [longx>=-78.52 : 7259,398] &
   [type=Caldera_Class,Shield_Class,Lava_Class,Volcanic_Class,Submarine_Class,Cinder_cone,Cinder_cones,Complex_volcanoes,Compound_volcano,Cones,Crater_rows,Explosion_craters,Lava_cone,Maar,Maars,Pumice_cone,Pumice_cones,Pyroclastic_cone,Pyroclastic_cones,Pyroclastic_shield,Stratovolcano,Stratovolcano_maybe,Stratovolcanoes,Tuff_cone,Tuff_ring,Tuff_rings,Twin_volcano : 8770,402]: p=3389,n=0
```

Note that this single rule covers 3389 positive examples and no negative examples

Summary of Experimental Results from Analyzing Volcano Database

- In all experiments with the volcano database, predictive accuracy was greater than 90% on separate testing data
- One surprising result was that the simpler strong patterns performed comparably to the detailed complete and consistent rulesets even though they involved far fewer rules and conditions
- The rules were understandable by the collaborating scientists from the Smithsonian Institution, who could modify them or use them as a basis for their own interpretation and understanding

An Example of Application to Demographics

Discovering Patterns in the World Factbook

- **Goal:** Discover interesting patterns in the demographic characteristics of different countries described in the World Factbook
- **Method:** Conduct a *grand tour* of the data in search of interesting patterns (in a grand tour, each attribute is consecutively used as an output one, and the rest as input attributes)
- **Input data** involved the 1993 dataset, which had records describing 190 countries
- Attributes used to describe countries include population growth rate, birth rate, death rate, net migration rate, fertility rate, infant mortality rate, literacy, life expectancy, and religion (treated as a structured attribute whose domain is a hierarchy).

Example of the Data

Country	Religion	NetMigRate	DeathRate	BirthRate	InfMortRate	LifeExp	FertRate	Literacy	PopGrRate
Afghanistan	Sunni_Muslim	neg10_to_0	15_to_20	40_to_50	GT100	40_to_50	6_to_7	LT30%	2_to_3%
Albania	Muslim	neg10_to_0	5_to_10	20_to_30	25_to_40	70_to_80	2_to_3	70_to_90%	1_to_2%
Algeria	Sunni_Muslim	neg10_to_0	5_to_10	30_to_40	40_to_55	60_to_70	3_to_4	50_to_70%	2_to_3%
Andorra	Roman_Cathol	GT20	5_to_10	10_to_20	LT10	70_to_80	1_to_2	?	3_to_4%
Angola	Mixed	neg10_to_0	15_to_20	40_to_50	GT100	40_to_50	6_to_7	30_to_50%	2_to_3%
Antigua_and	Anglican	neg10_to_0	5_to_10	10_to_20	10_to_25	70_to_80	1_to_2	70_to_90%	LT1%
Argentina	Roman_Cathol	0_to_10	5_to_10	10_to_20	25_to_40	70_to_80	2_to_3	90_to_95%	1_to_2%
Armenia	Armenian_Ort	neg10_to_0	5_to_10	20_to_30	25_to_40	70_to_80	3_to_4	100%	1_to_2%
Australia	Mixed	0_to_10	5_to_10	10_to_20	LT10	70_to_80	1_to_2	100%	1_to_2%
Austria	Roman_Cathol	0_to_10	10_to_15	10_to_20	LT10	70_to_80	1_to_2	95_to_99%	LT1%
Azerbaijan	Muslim	neg10_to_0	5_to_10	20_to_30	25_to_40	70_to_80	2_to_3	100%	1_to_2%
The_Bahamas	Mixed	0_to_10	5_to_10	10_to_20	25_to_40	70_to_80	1_to_2	70_to_90%	1_to_2%
Bahrain	Shi'a_Muslim	0_to_10	0_to_5	20_to_30	10_to_25	70_to_80	3_to_4	70_to_90%	3_to_4%
Bangladesh	Muslim	neg10_to_0	10_to_15	30_to_40	GT100	50_to_60	4_to_5	30_to_50%	2_to_3%
Barbados	Protestant	neg10_to_0	5_to_10	10_to_20	10_to_25	70_to_80	1_to_2	95_to_99%	LT1%
Belarus	Eastern_Orth	0_to_10	10_to_15	10_to_20	10_to_25	70_to_80	1_to_2	100%	LT1%
Belgium	Roman_Cathol	0_to_10	10_to_15	10_to_20	LT10	70_to_80	1_to_2	95_to_99%	LT1%
Belize	Roman_Cathol	neg10_to_0	5_to_10	30_to_40	25_to_40	60_to_70	4_to_5	90_to_95%	2_to_3%
Benin	indigenous	neg10_to_0	10_to_15	40_to_50	GT100	50_to_60	6_to_7	LT30%	3_to_4%
Bhutan	Lamaistic	neg10_to_0	15_to_20	30_to_40	GT100	50_to_60	5_to_6	?	2_to_3%

A Learned Rule that Helped to Discover Anomaly

- Among rules learned was a rule characterizing 25 of the 55 countries with low (<1%) population growth:

  [PopGrRate < 1%]

  <= [BirthRate = 10..20 or 50..60: 46, 20] &

  [FertRate = 1..2 or >7: 32, 17] &

  [Religion is Protestant or Catholic or Orthodox or Shinto: 38, 32] &

  [NetMigRate < +10: 54, 123]

- The first and strongest condition is surprising. The condition points to a low birth rate, which satisfies our intuitive, but also to a very high one
- Looking at the 25 countries that satisfy this rule, 24 have birth rates less than or equal 20. Only one, Malawi, has a birth rate above 50
- Investigating Malawi against the rest of the countries quickly revealed an explanation: the country has an outward net migration rate that dwarfs those of all other countries.

An Example of Application to Manufacturing

Generating Rules for Gearbox Design

- **Goal:** Assist manufacturer in designing gearboxes meeting customer specifications
- **Method:** Create a database describing previously filled orders and learn inductively the conditions that suggest that a given component should be used in a gearbox
- **Input attributes** characterize the individual user specifications (line, model, size, mount, motor, dH7, flange, disco, gear ratio), and **output attributes** indicate types and of components to be used (e.g., housing, shaft, flange, lubricant)
- This work was conducted in collaboration with Lenze GmbH & Co KG, Germany.

Examples of a Ruleset Learned

Schneckenwelle=00650943 ← [Ratio = 5] & [Model=104,105,113,145]
00650944 ← [Ratio = 7] & [Model=104,105,113,145]
00650945 ← [Ratio = 10] & [Model=104,105,113,145]
00650946 ← [Ratio = 13] & [Model=104,105,113,145]
00650947 ← [Ratio = 15] & [Model=104,105,113,145]
00650948 ← [Ratio = 20] & [Model=104,105,113,145]
00650949 ← [Ratio = 26] & [Model=104,105,113,145]
00652155 ← [Ratio = 5] & [motor size = 80]
00652156 ← [Ratio = 7] & [motor size = 80]
00652157 ← [Ratio = 10] & [motor size = 80]
00652158 ← [Ratio = 13] & [motor size = 80]
00652159 ← [Ratio = 15] & [motor size = 80]
00652160 ← [Ratio = 20] & [motor size = 80]
00652161 ← [Ratio = 26] & [motor size = 80]
00652143 ← [Ratio = 5] & [motor size = 90]
00652144 ← [Ratio = 7] & [motor size = 90]
00652145 ← [Ratio = 10] & [motor size = 90]
00652146 ← [Ratio = 13] & [motor size = 90]
00652147 ← [Ratio = 15] & [motor size = 90]
00652148 ← [Ratio = 20] & [motor size = 90]
00652149 ← [Ratio = 26] & [motor size = 90]

The first rule is interpreted: If the user requests a gearbox with “gear ratio=5 and model 105, 105, 113 or 145,” then use the part number 00650943 as the “Schneckenwelle“ (worm shaft) component of the gearbox. Other rules are interpresed similarly.

Results of Application to Gearbox Design

- Some of the discovered rules were straightforward -- for requested size or gear ratio, certain parts were dictated. Others rules were more complex
- Learned rules provided insights into the relationships and constraints of the gearbox manufacturing domain, and even exposed some errors in the data
- Classifier was able to select components with 100% accuracy (such a result was possible because training set contained all practical cases and they were captured by the rules).

An Example of Application to User Modeling and Intrusion Detection

- For this task, we developed a new methodology, called LUS (Learning User Signatures) that employs symbolic learning to derive patterns in the datastreams characterizing individual users
- LUS has several versions, depending on the type user model employed
- The following models have been developed: Multistate Templates (MT), Prediction-based (P), Rule-Bayesian (RB), and Activity-based (A)
- Here we present briefly results using MT model (LUS-MT project)
- One important feature of the MT model is that the user signatures are relatively easy to interpret and can potentially be edited by a user.

Brief Overview of LUS-MT

- User models are created automatically through symbolic learning from training data streams characterizing users' interaction with computers
- Learning of user models is a multi-step process that consists of a determination of the most relevant attributes and the most relevant events in training target dataset for each user, and an application of a learning method, or a combination of learning methods under appropriate parameter settings
- Knowledge representation for user models is based on *Attributional Calculus*, a logic and representation system that combines elements of propositional logic, predicate logic, and multiple-valued logic (Michalski, 2004; www.mli.gmu.edu/papers)
- The methodology strives to develop user models that can be easily interpreted by human experts and/or modified manually, and reliably detect illegitimate user behavior from user data streams that are as short as possible
- This research has developed a wide range of new ideas, concepts, methods, and computer programs for learning and testing user models.

Cognitive Aspects of LUS Methodology

LUS strives to emulate several important aspects of human learning and recognition processes:

- Idiosyncracy: It searches for patterns that are most characteristic of a given user, so that recognition is possible from short episodes that contain such features
- Satisfiability: If, at some point the observed behavior strongly matches one user model, and only weakly matches other models, the observation of the users' data stream stops, and a decision is reported
- Understandability: It strives for creating user models that are easy to interpret and understand by humans
- Incrementability: User models can be updated incrementally over time, without re-learning them from scratch.

Raw Experimental Data

- The raw data stream (from the NJIT archive) comprises three sets of data consisting of records in process table
- Each set contains 1282 sessions from 26 users
- The following slide shows the number of sessions recorded for each of 26 users in each dataset. The number of sessions recorded for each user is not constant. For five users only one session was recorded. For User 1, 287 sessions were recorded
- From the available data we selected 10 Users that have the highest number of sessions and the first 10 sessions of each of the Users were selected for training and the following 5 for testing
- These data are used to create different training and testing target datasets.

Raw Datastreams

User	Number of Sessions	User	Number of Sessions	User	Number of Sessions	User	Number of Sessions
1	287	8	167	15	1	22	1
2	54	9	21	16	7	23	1
3	37	10	17	17	15	24	4
4	134	11	6	18	5	25	99
5	34	12	35	19	134	26	9
6	1	13	14	20	10		
7	193	14	1	21	5		

The ten users with the most recorded sessions (shaded) were selected for the Phase 1 experiments.

Examples of Original Attributes Used for Characterizing User Behavior

The data were extracted from the raw data, such that only processes corresponding to the main task of the active window were logged.

Process names: over 100 names

Records were of two types:

- Records corresponding to changes in the active window's title (W type)
- Records corresponding to activity in the active window (A type)

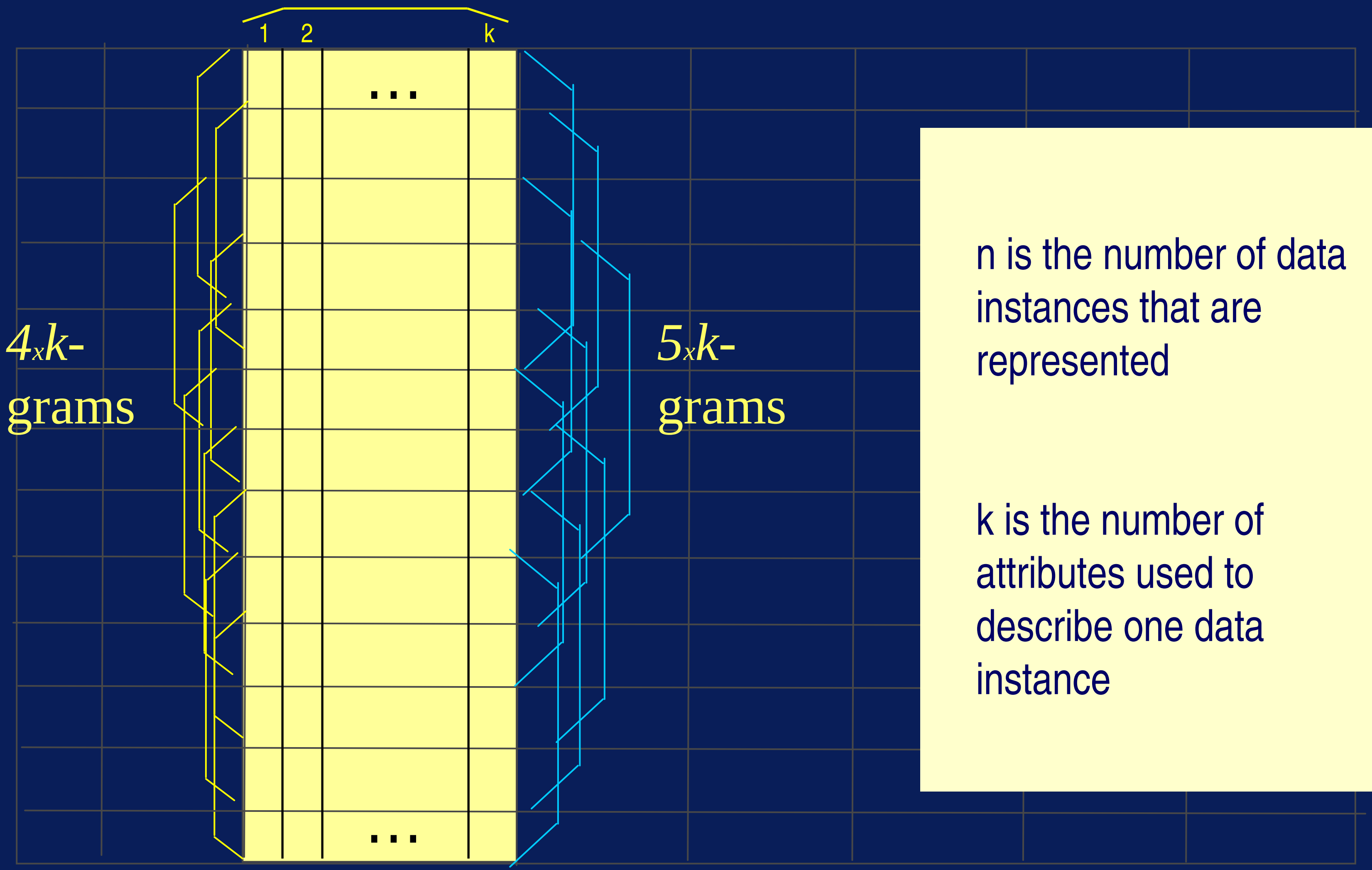
Attributes are explained in the table to the right.

Field Name	Field Type	Record Type	Field Description
LineNo	Parenthesized Integer	W, A	Line number in the raw process table data corresponding to this record
Delta_t	Real	W, A	Number of seconds since start of session when this record was generated
Process name	String	W, A	General description of the user's current activity or program
PID	Integer	W	Process ID of the active window
Status	Character	A	Indicates whether process is new, continuing, ending, or running in the background
CPU	Real	A	Total amount of CPU time used by the process when this record was generated
WinName	String	W	Name of the active window (sanitized)
Process lineage	String	A	Lineage of the process from its base window process

Multistate Template User Model

- User signatures are learned from sets of multigrams obtained from the training user data stream
- The learning process employs the AQ-type learning program, AQ21, for generating attributional rules. The rules define templates in the form of Cartesian products of multistate conditions
- Program EPIC-MT is used for matching user data streams with user models, and computing a matching score for a given testing episode
- EPIC-MT employs the ATEST program for matching individual events with multistate templates.

Extracting *nxk*-grams from Data Streams



n is the number of data instances that are represented

k is the number of attributes used to describe one data instance

An Example of Training Data Consisting of *n*x*k*-grams

```
(#
The time of creation is 2004-02-19 13:24
Input data file name is /home/shared/data/lus-njit/user1-host19-12_12_01-09_35_45.1s
This output data filename is njit-top10ts5-all-lb4.lus
There is no parameters file for AQ created
Lookback parameter is 4 for all attributes
User parameter is ua
Episode number is 281
#)
 user1,281,host19,host19,host19,host19,host19,Wed,Wed,Wed,Wed,Wed,09,09,09,09,09,lt300,lt300,lt300,
 lt300,lt300,msoffice,msoffice,msoffice,msoffice,msoffice,c,c,c,c,n,lt60,lt60,lt60,lt60,gte180,20,0,20,
 0,N/A,3.04452,0,3.04452,0,N/A,lte60,lte60,lte60,lte60,N/A,252,252,252,252,252,532,532,532,532,532,lt60,
 lt60,lt60,lt60,lt60,0,0,0,0,0,lt300,lt300,lt300,lt300,lt300,4.39445,4.39445,4.39445,4.39445,4.39445,
 bt0and02,bt0and02,bt0and02,bt0and02,bt0and02,lt300,lt300,lt300,lt300,lt300,0,0,0,0,0,lt300,lt300,lt300,
 lt300,lt300,4.41884,4.41884,4.41884,4.41884,4.41884,bt20and40,bt20and40,bt20and40,bt20and40,bt20and40,
 bt08and1,bt08and1,bt08and1,bt08and1,bt08and1,lt10,lt10,lt10,lt10,lt10,bt08and1,bt08and1,bt08and1,
 bt08and1,bt08and1,lt100,lt100,lt100,lt100,lt100,1.79176,1.79176,1.79176,1.79176,1.79176,lt20,lt20,lt20,
 lt20,lt20,0.693147,0.693147,0.693147,0.693147,0.693147,4,4,4,4,4,lt20,lt20,lt20,lt20,lt20
 user1,281,host19,host19,host19,host19,host19,Wed,Wed,Wed,Wed,Wed,09,09,09,09,09,lt300,lt300,lt300,
 lt300,lt300,msoffice,msoffice,msoffice,msoffice,msoffice,c,c,c,c,c,lt60,lt60,lt60,lt60,lt60,1,20,0,20,
 0,0.693147,3.04452,0,3.04452,0,lte60,lte60,lte60,lte60,lte60,252,252,252,252,252,532,532,532,532,532,
 lt60,lt60,lt60,lt60,lt60,0,0,0,0,0,lt300,lt300,lt300,lt300,lt300,4.39445,4.39445,4.39445,4.39445,
 4.39445,bt0and02,bt0and02,bt0and02,bt0and02,bt0and02,lt300,lt300,lt300,lt300,lt300,0,0,0,0,0,lt300,
 lt300,lt300,lt300,lt300,4.41884,4.41884,4.41884,4.41884,4.41884,bt20and40,bt20and40,bt20and40,
 bt20and40,bt20and40,bt08and1,bt08and1,bt08and1,bt08and1,bt08and1,lt10,lt10,lt10,lt10,lt10,bt08and1,
 bt08and1,bt08and1,bt08and1,bt08and1,lt100,lt100,lt100,lt100,lt100,1.79176,1.79176,1.79176,1.79176,
 1.79176,lt20,lt20,lt20,lt20,lt20,0.693147,0.693147,0.693147,0.693147,0.693147,4,4,4,4,4,lt20,lt20,lt20,
 lt20,lt20
……
```

The Same Multistate Template in an Abbreviated Form

```
[User = user4]
   <= [hour= << 11..14 : 3393,9171 (3) >> ] &
      [process_name = < netscape,outlook,winword : 3904,18376;
                        csrss,netscape,outlook,winword : 3909,18413;
                        csrss,netscape,outlook,winword : 3909,18397;
                        csrss,netscape,outlook,winword : 3909,18379 > ] &
      [event_status = < c,o : 3997,22090; c,o : 3997,22113; c,o : 3997,22123; * > ] &
      [proc_cpu_time_in_win_lf = < 0.3466..4.049 : 3611,12784; *; *; lt_3.916 : 3994,20119 > ] &
      [win_time_elapsed_lf = << gt_3.337 : 3251,13713 (1) >> ] &
      [delta_time_new_window = << lt_1800 : 3985,21445 (1) >> ] &
      [delta_time_new_window_lf = <<lt_7.748 : 3987,21518 (4) >> ] &
      [new_win_time_elapsed = <<300..18000 : 3954,16719 (4)>>] &
      [prot_words_chars = << lt_20 : 3980,17938 (1) >> ] &
      [proc_count_in_win_lf = << gt_4.063 : 3060,7992 (1) >>] &
      [win_opened_lf = < <1.498..2.636 : 3600,13531 (4) >> ]
      p = 2419, n = 0, P = 4000, N = 25173
```

Example of a Multistate Template in Generalized and Simplified Form

[User = user4]
<= [hour= << 11..14 >>] &
[process_name = < netscape,outlook,winword;
csrss,netscape,outlook,winword; csrss,netscape,outlook,winword;
csrss,netscape,outlook,winword>] &
[event_status = <<c,o>>] &
[proc_cpu_time_in_win_lf = < 0.3..4.0; *; *; lt_3.9>] &
[win_time_elapsed_lf = << gt_3.3 >>] &
[delta_time_new_window = << lt_1800 >>] &
[delta_time_new_window_lf = <<lt_7.7 >>] &
[new_win_time_elapsed = <<300..18000 >>] &
[prot_words_chars = << lt_20 >>] &
[proc_count_in_win_lf = << gt_4.1 >>] &
[win_opened_lf = < <1.5..2.6 >>] : p = 2419, n = 0;
P = 4000, N = 25173

Experiment 040607-1: Parameters

Training Dataset:

Discretization: Dis-3; **Filtering:** Significance-based, conjunctive, rank-threshold = 10, TR+TS

Testing Dataset:

Discretization: Dis-3; **Filtering:** not filtered

AQ21 Learning Parameters:

maxstar = 1 maxrule = 1 ambiguity = ignore-for-learning

trim = optimal exceptions = false mode = tf

Discriminant descriptions

Testing Parameters:

Evaluation of Conjunction = strict

Evaluation of Disjunction = max

Acceptance Threshold = 10%

Accuracy Tolerance = 5%

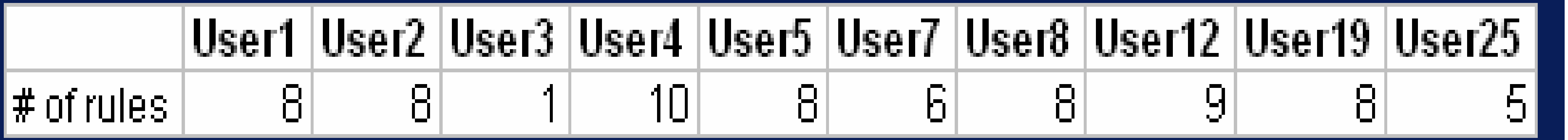
Experiment 040607-1: Results

Learning Results: Total number of rules: 71

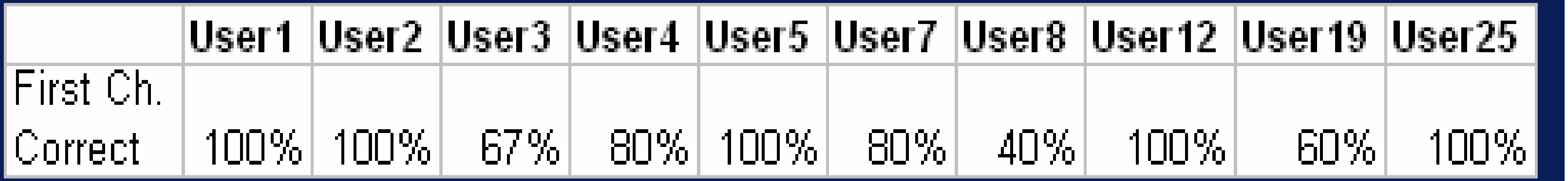
Testing Results: Correct: 79.17% ;Precision: 82.46%;

First Choice Correct: 75%; First Choice Precision: 100%

Whenever training data were sufficiently relevant to the testing data (as measured by the combined forward and backward similarity between the training and testing datastreams) predictive accuracy was high—as expected.



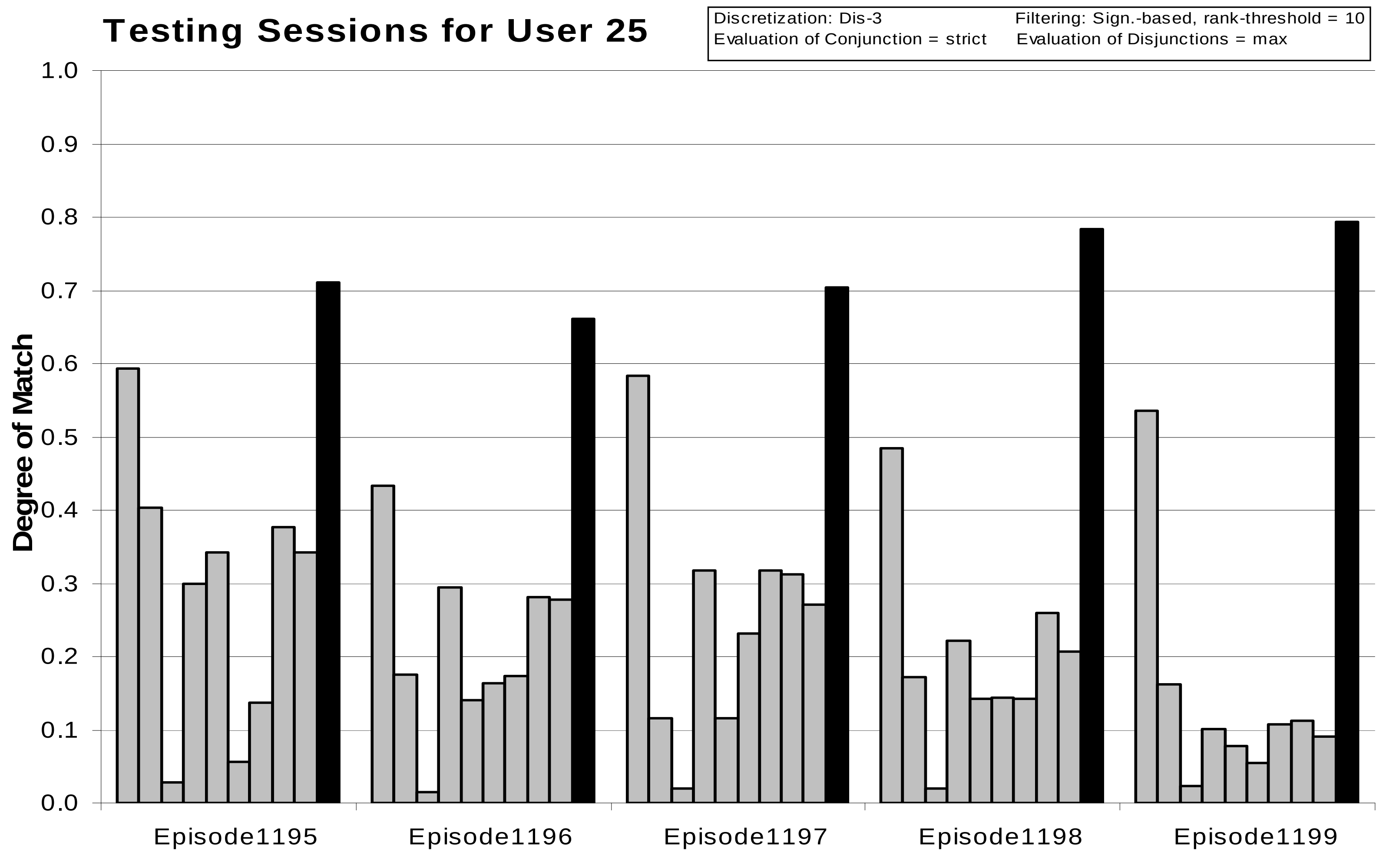
	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
# of rules	8	8	1	10	8	6	8	9	8	5



	User1	User2	User3	User4	User5	User7	User8	User12	User19	User25
First Ch. Correct	100%	100%	67%	80%	100%	80%	40%	100%	60%	100%

Experiment 040607-1 Testing Summary (User 25)

Graphical Illustration



Summary of Experiments with LUS-MT

- We have explored a small subspace of possible experiments on learning and testing of the developed user modeling methods.
- Experiments have shown that LUS-MT method can lead to an effective system for user modeling and intrusion detection under the following conditions:
  1. Sufficient training data stream for each user is available
  2. Target training data set for each user is appropriately determined
  3. The user's future behavior is sufficiently similar to the one recorded in the training data stream

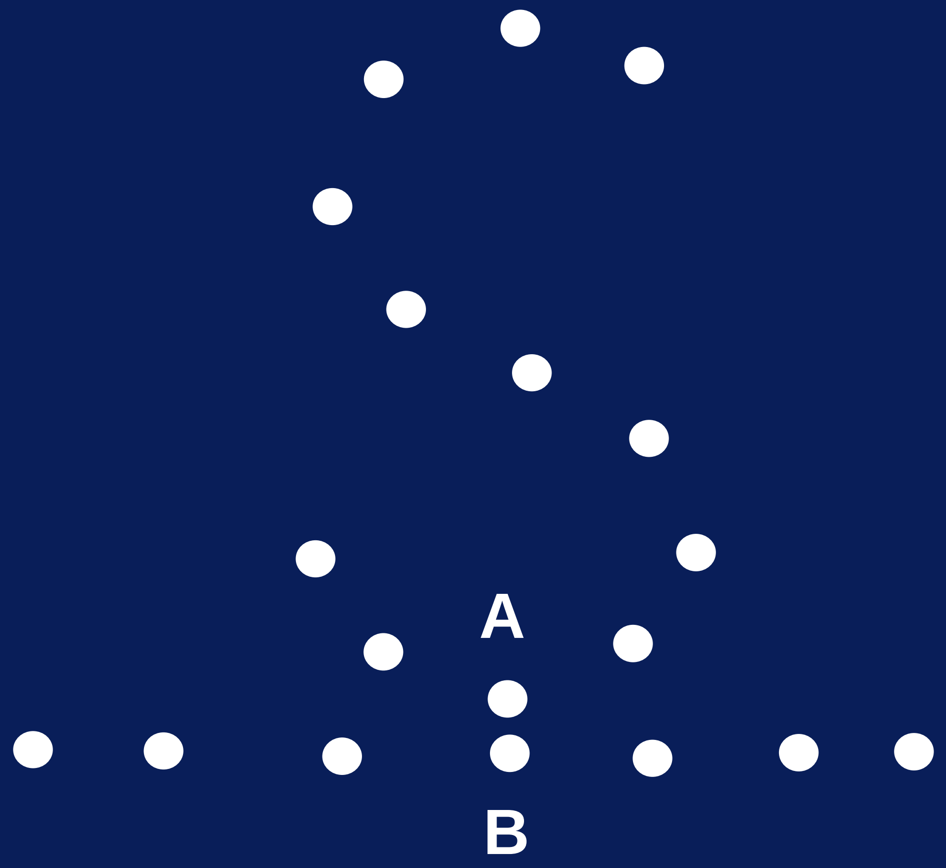
Conceptual Clustering

- It is an approach to clustering (or unsupervised learning) that seeks “conceptual clusters” in data, that is, groups of entities representing simple concepts
- It outputs both clusters and cluster descriptions (the form of attributional conjunctions), unlike conventional clustering that outputs only clusters
- Clusters are determined on the basis of desired properties of their descriptions
- Cluster descriptions are created using a combination machine learning and dynamic clustering (CLUSTER3 applies a version of AQ21).

An Illustration of Conceptual Clustering

How would you cluster the entities below **?**

A method that clusters entities on the basis of similarity (a reciprocal of distance) would put points A and B into the same cluster, while conceptual clustering clusters A and B into different clusters (corresponding to concepts “Letter S” and “Straight line”)



The CLUSTER3 Program for Conceptual Clustering

- Given a set of entities described by attribute values, the program splits them into clusters described by simple attributional descriptions

- Any clustering of given entities can usually be done in many different ways depending on the viewpoint from which one clusters them

- CLUSTER3 applies a *view-relevant attribute subsetting* method, VAS, that automatically determines attributes relevant to the given task

- To seek optimal clustering, CLUSTER3 applies a Lexicographic Evaluation Functional (LEF) that combines several elementary criteria characterizing descriptions of generated clusters

- The next slides illustrate the working of CLUSTER3 on a simple problem.

A Simple Example

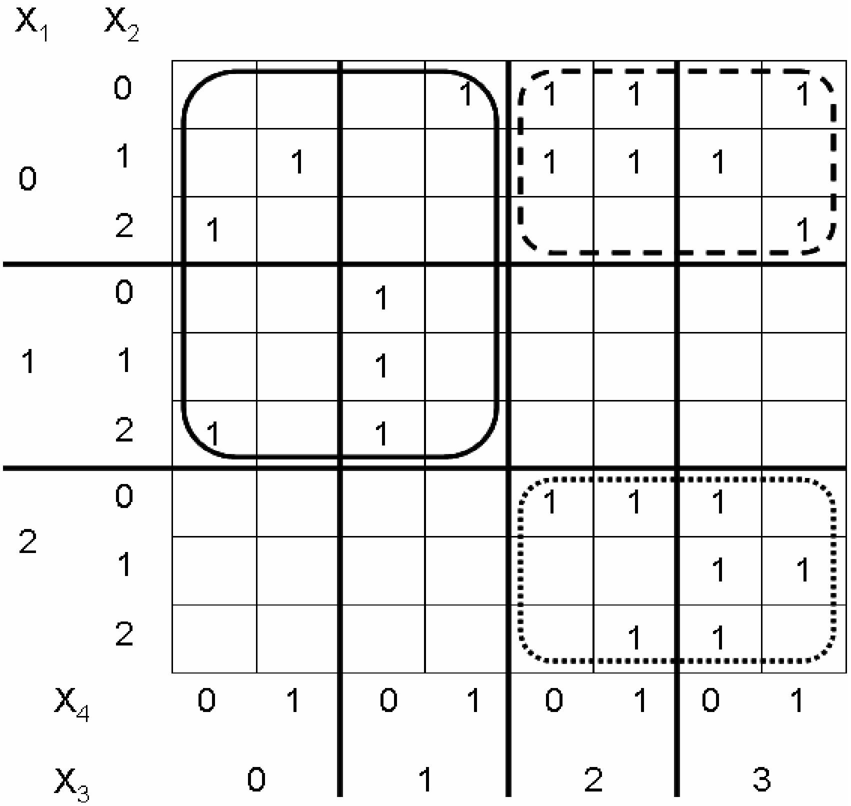
- The dataset to cluster consists of 21 objects described in terms of four attributes:
  - X1: {0,1,2}
  - X2: {0,1,2}
  - X3: {0,1,2,3}
  - X4: {0,1}
- The space spanned over these attributes is visualized using General Logic Diagram on the slide “Clusterings Obtained by CLUSTER3 and KMlocal” (two slides ahead)

CLUSTER3 vs. KMlocal

- To illustrate the difference between conceptual clustering and conventional clustering, CLUSTER 3 and KMlocal were applied to this problem
- KMlocal, implementing Lloyd's algorithm, assigns observations to clusters using the minimum Euclidean distance between the observation and the cluster centroids
- KMlocal was run with default parameters and in 1000 stages
- CLUSTER3 was run with default parameters and the clustering quality criterion combining *balance* and *commonality*, with tolerance τ=10% for both criteria
- The assumed number of clusters was 3 in both programs.

Clusterings Obtained by CLUSTER3 and KMlocal

CLUSTER3

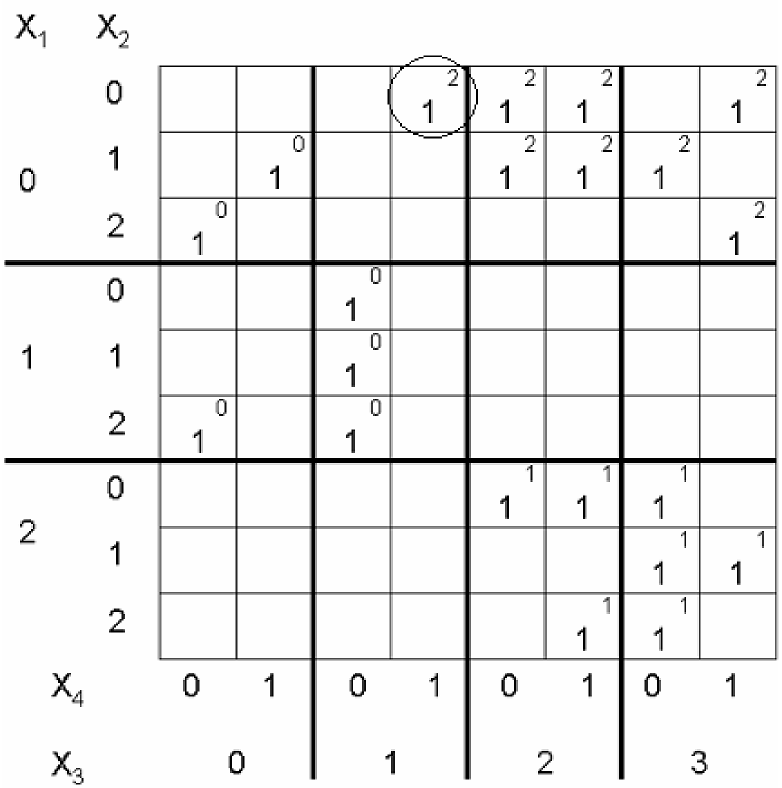


X_1	X_2	X_3 = 0		1		2		3	
	X_4	0	1	0	1	0	1	0	1
0	0				1	1	1		1
	1		1			1	1	1	
	2	1							1
1	0			1					
	1			1					
	2	1		1					
2	0					1	1	1	
	1							1	1
	2						1	1	

Cluster Descriptions;

- ──── Cluster 0: $[X_1=0..1]\&[X_3=0..1]$
- – – – – Cluster 1: $[X_1=0]\&[X_3=2..3]$
- ········ Cluster 2: $[X_1=2]\&[X_3=2..3]$

KMlocal



X_1	X_2	X_3 = 0		1		2		3	
	X_4	0	1	0	1	0	1	0	1
0	0				1^{2}	1^{2}	1^{2}		1^{2}
	1		1^{0}			1^{2}	1^{2}	1^{2}	
	2	1^{0}							1^{2}
1	0			1^{0}					
	1			1^{0}					
	2	1^{0}		1^{0}					
2	0					1^{1}	1^{1}	1^{1}	
	1							1^{1}	1^{1}
	2						1^{1}	1^{1}	

The object (0,0,1,1), denoted by a circle in the diagram, was clustered by KMlocal into cluster 2, while CLUSTER3 put it into cluster 1. Note that KMlocal does not provide any description of generated clusters.

Application to Tax Fraud Detection

(Scott Fischthal)

Method

Combines Conceptual Clustering and Natural Induction

1. Group data according to characteristics of the clustering descriptions determined by the CLUSTER program applied to a subset of data.

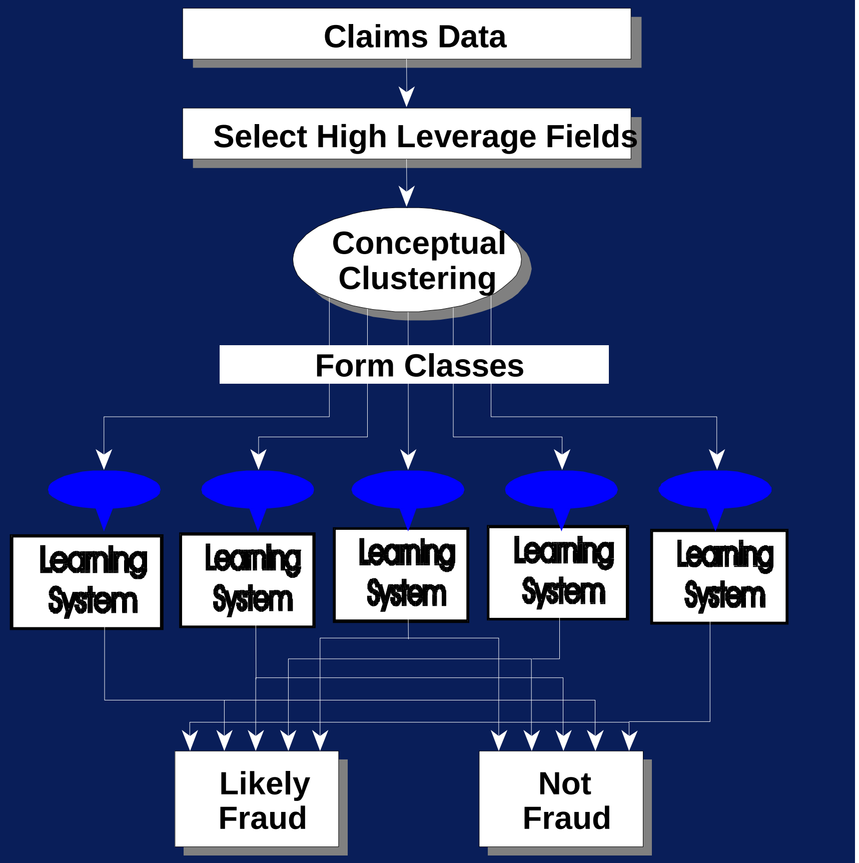
   (Cluster descriptions are in the form of attributional conjunctions and provide an insight into the meaning of the clusters.)

2. Apply natural induction (AQ-based supervised learning) to examples of known fraud in each group discovered by conceptual clustering in order to discover simple rules distinguishing regular and fraudulent tax forms

3. Apply the rules to new tax returns

A Methodology for Fraud Detection Combining Conceptual Clustering and Natural Induction

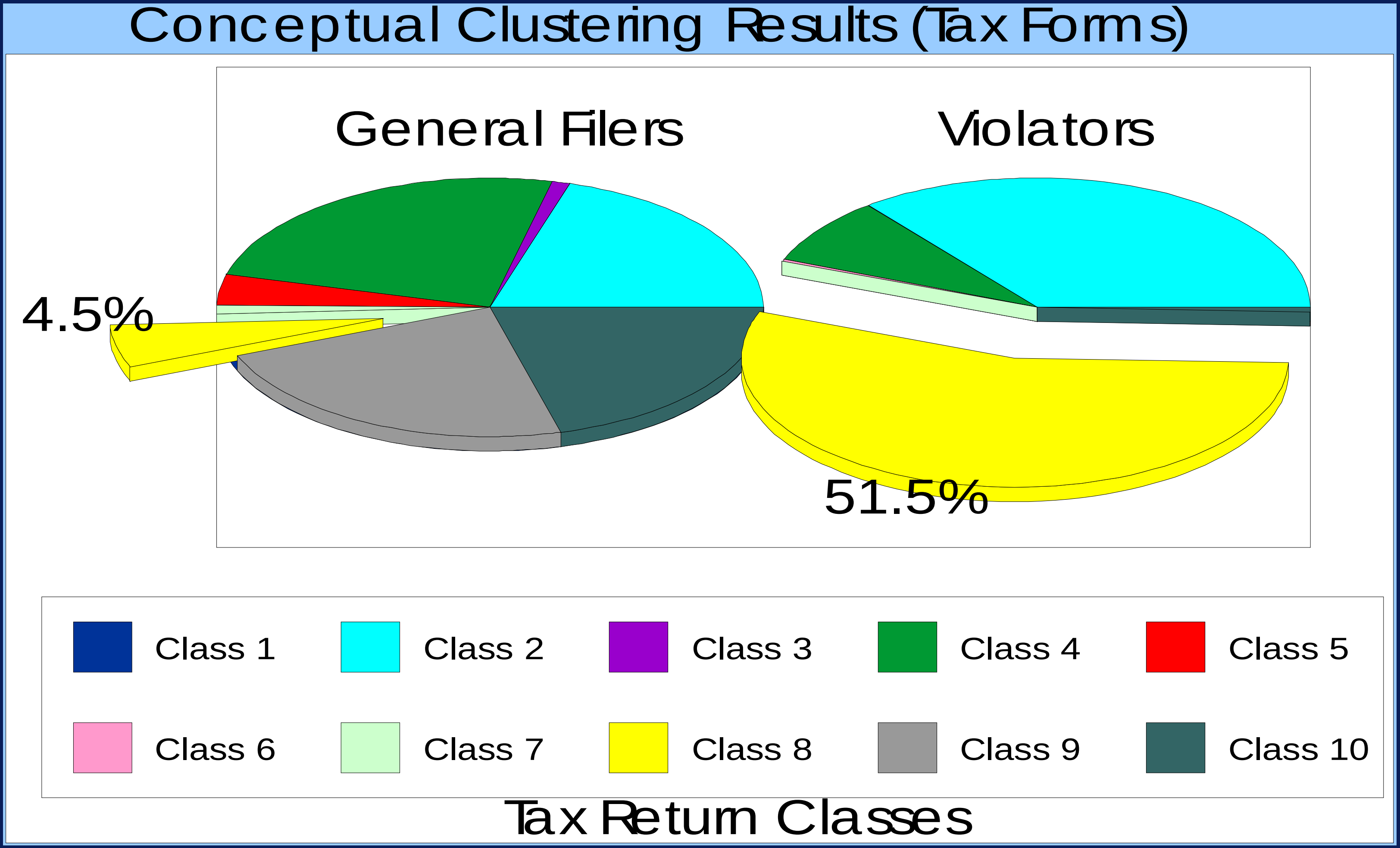
This methodology is applied when training data is not available (unsupervised learning + supervised)



Potential Applications:

- Tax form fraud detection
- Financial disclosure form evaluation
- Health care fraud detection
- Message filtering
- Credit evaluation
- Insurance

Discovery of a Subgroup of Taxpayers with a High Incidence of Tax Violation



Conclusion

- The presented examples illustrate application of natural induction and conceptual clustering to a wide range of practical problems
- Natural induction is currently implemented in AQ21, and conceptual clustering in CLUSTER3
- Only some capabilities of AQ21 have been illustrated in these examples; many aspects have not been presented, such as the ability to learn descriptions with exceptions, to use *count attributes* and *compound attributes*, to perform constructive induction, to generate alternative hypotheses, to handle unknown, not-applicable and irrelevant attribute values in data, and to generate rules at different levels of generality.

See papers in *www.mli.gmu.edu* for more information about MLI projects.