Introduction to Data Mining on Grids

Robert Grossman
University of Illinois at Chicago
& Open Data Group

Michal Sabala
University of Illinois at Chicago

Midwest Grid Workshop
University of Illinois at Chicago
March 25, 2007

1

Table of Contents

We emphasize a few basic patterns so that we can use grids for simple data mining applications.

2

What We Are Not Covering

- Non vector data
  - Semi-structured data
  - Graphs
  - Images, continuous media, etc.
- Distributed data mining algorithms
- Workflow
- Data providence
- Knowledge grids
- Many other relevant items

3

Section 1

Introduction to Data Mining

4

What is Data Mining?

Short definition:

- Finding interesting structure in data. (Interesting implies actionable.)

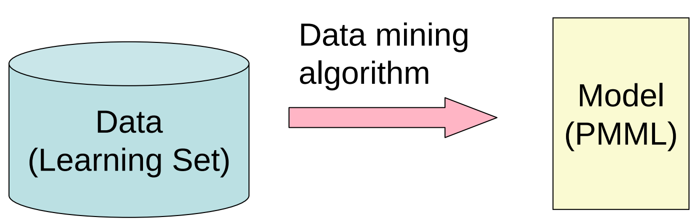
Long definition:

- Semi-automatic discovery of patterns, correlations, changes, associations, anomalies, and other statistically significant structures in large data sets.

5

What is Data Mining?

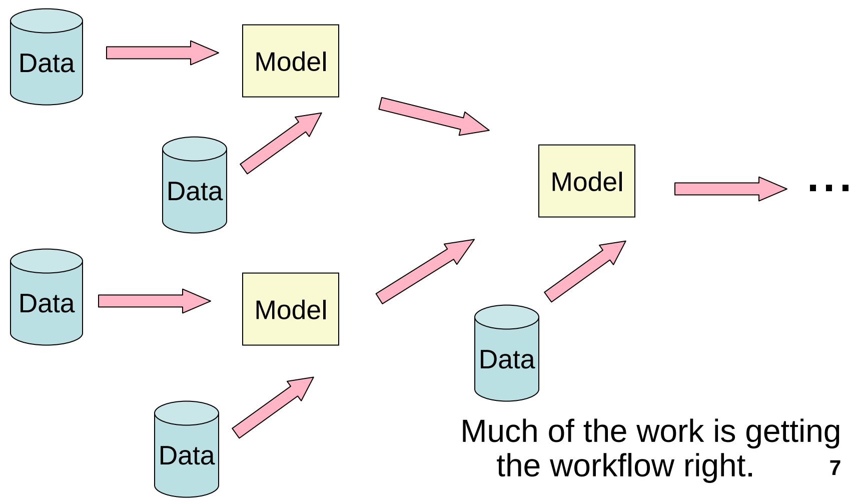
Architectural view



- Actually, usually, this is a component in a workflow
- PMML is the Predictive Model Markup Language

6

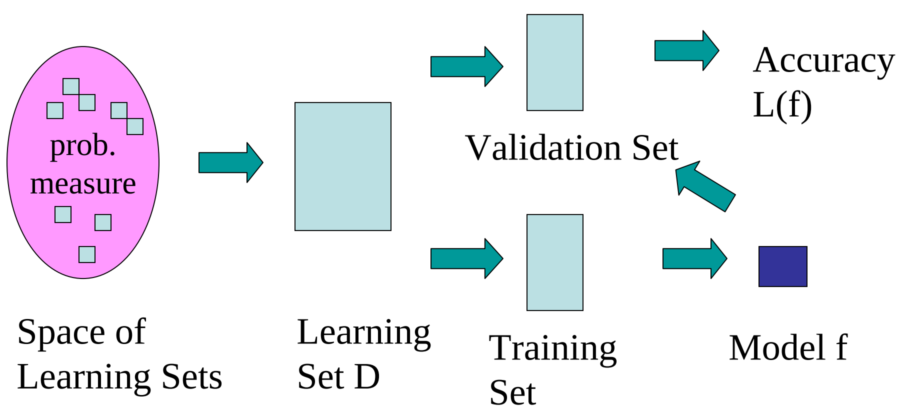
In General, This is Part of a Workflow



Much of the work is getting the workflow right. 7

How Can This Work?

That is Why Does the Model Generalize?

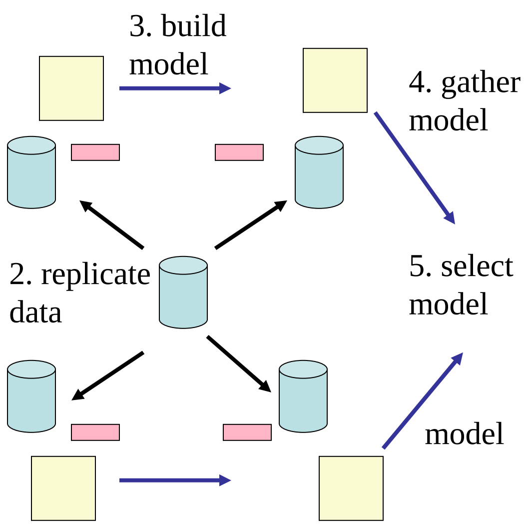


- R^d x {0,1}-valued random pair (X,Y)
- L(f) = P (f(X) = Y), expected accuracy E(L(f)) 8

Section 2

Three Basic Patterns for Using Grids for Data Mining

Pattern 1: Parameter Search



1. Partition parameter space
2. Replicate data
3. Build individual models on separate processors
4. Gather models
5. Finally, **select** best model

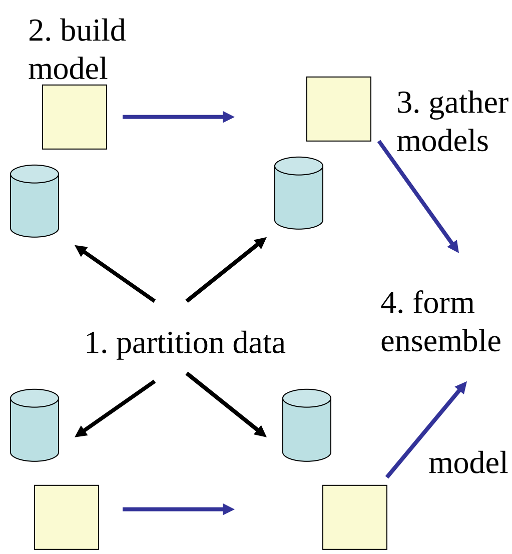
1. partition parameters

Parameter Search (cont'd)

- Basic Steps
  - Fix one data set
  - Divide up space of parameters into parameter segments
  - Scatter data set and assign each processor to a different part of parameter space
  - Gather results
  - Rank results by objective function

11

Pattern 2: Ensemble Models



1. Partition data and scatter
2. Build models (e.g. tree-based model)
3. Gather models
4. Form collection of models into **ensemble** (e.g. majority vote for classification & averaging for regression)

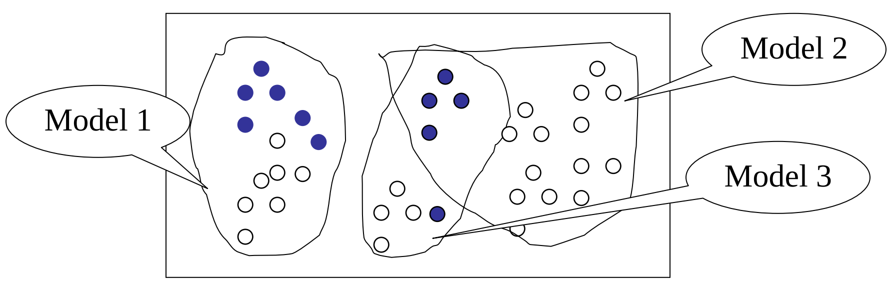
12

Ensemble Models (cont'd)

- Basic Steps
  - Split the data set into segments
  - Scatter segments to different processes
  - Build separate models over each segment
  - Gather the models
  - Form individual models into ensemble of models
  - Evaluate performance of ensemble on hold out set

13

The Key Idea of Ensembles: Combine Weak Learners



- It is often better to build several models, and then to average them, rather than build one complex model.
- Think of model i as function f_i: R^n ---> R and simply average the f_i for regression or use a majority vote for classification.

14

Combining Weak Learners

1 Classifier	3 Classifiers	5 Classifiers
55%	57.40%	59.30%
60%	64.0%	68.20%
65%	71.00%	76.50%

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

15

Three Other Patterns

3. Task level parallelism of data mining algorithms over grids using MPI or related technology
4. Map-reduce and related styles
5. Process data locally, say with a peer-to-peer network

We won’t have time to discuss these.

16

Section 3

Architectures for Data Mining

17

Five Questions to Ask

1. What size is the data and how do we physically access it?
2. What shape is the data?
3. Where is the data?
4. Do you move the data or the query?
5. What data mining APIs or data mining services are available? Are they standards based or custom?

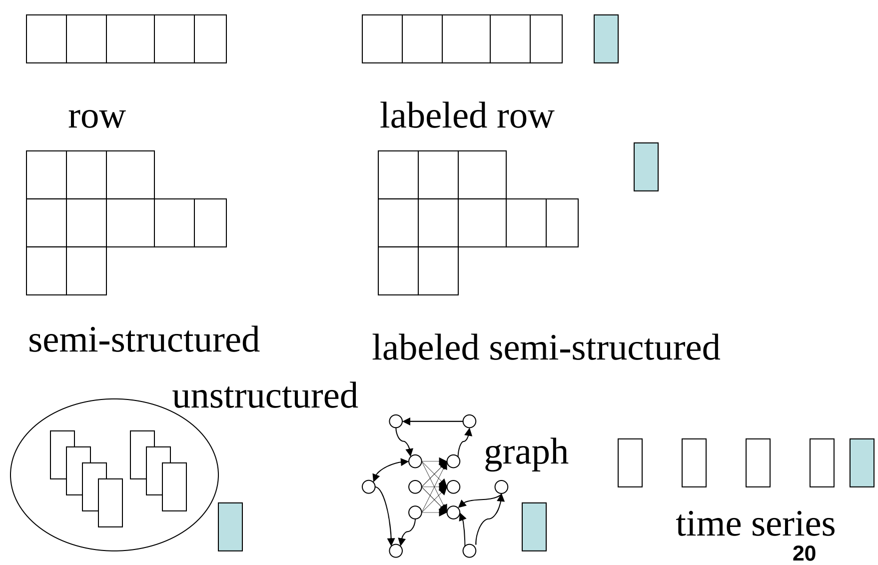
18

What Size is the Data?

- Small
  - Fits into memory
- Medium
  - Too large for memory
  - But fits into a database
  - N.B. database access is essentially row by row
- Large
  - Too large for a database
  - But can use specialized file system
  - For example
    - Column-wise warehouses (i.e. access column by column)
    - Google file system, Google BigTable

19

What is the Shape of the Data?



20

Where is the Data?

- In main memory
- In a database
- In a data warehouse or data cube
- In a grid
- In column-wise warehouses
- In a peer to peer network

21

Do You Move the Data or the Query?

Strategy	Move the data to the processors	Move the query, build models and assemble the results
Situation	Small data, lots of processors	Large data or limited bandwidth

22

What Analytic/Data Mining Services are Available?

- And, how are they are available?
  - Through a proprietary API
  - Through a database API?
  - Through a web service
  - Through a grid service
- Proprietary applications
  - Statistical applications: e.g. SAS, SPSS, S-PLUS?
  - Database applications: Microsoft, IBM, Oracle?
- Open source applications (R, Octave, etc.)
- Specialized applications (Augustus, etc.)

23

Section 4

Three Basic Data Mining Algorithms

24

Three Core Data Mining Algorithms

4.1 Nearest neighbor algorithms

4.2 k-means clustering

4.3 Classification and regression trees

Section 4.1
Nearest Neighbor Learning

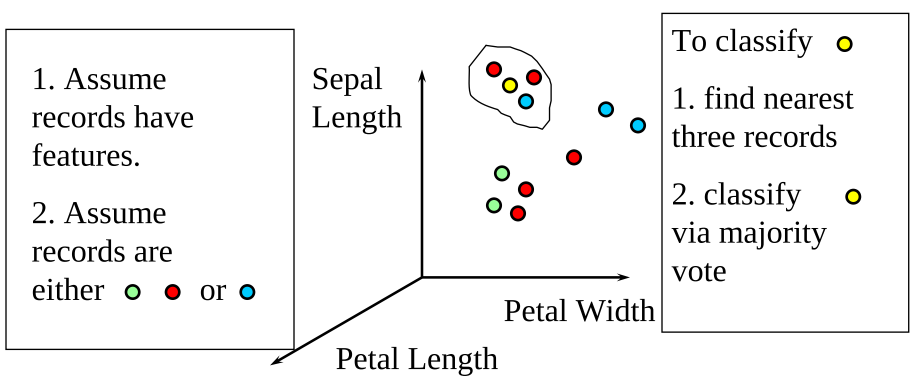
Classification

Petal Len.	Petal Width	Sepal Len.	Sepal Width	Species
02	14	33	50	A
24	56	31	67	C
23	51	31	69	C
13	45	28	57	B

- Assume data is arranged into rows (records) and columns (attributes or features)
- Assume each row is classified A, B or C
- Goal: given unclassified record, to classify it.

27

k-Nearest Neighbor Learning



- View records as points in feature space
- Find k-nearest neighbors and take majority vote.
- Example of supervised learning.

28

(j, k) Nearest Neighbor Learning

- Choose j points from the test set to produce a model f[1]. Choose another j points to produce a model f[2], etc.
  - This gives an ensemble of models:
    {f[1], …, f[p]}
  - Selecting the j points can be done in many different ways.
- To classify a point,
  1. evaluate each of the k-nearest neighbor models in the ensemble
  2. use a majority vote to get an overall class

29

Learning - Map from Data to Models

Petal Len.	Petal Width	Sepal Len.	Sepal Width	Species
02	14	33	50	A
24	56	31	67	C
23	51	31	69	C
13	45	28	57	B

Learning Sets (n data points)

<pmml><nearest-neighbor>…
02 14 33 50 A
13 45 28 57 B
</nearest-neighbor></pmml>

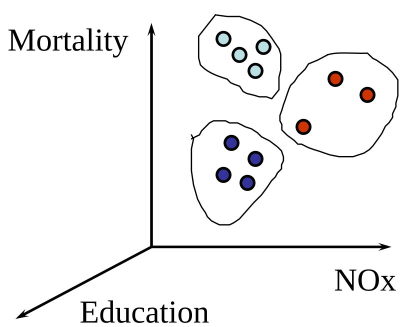
Models or Rules (j points)

30

Section 4.2
Cluster-based Learning

31

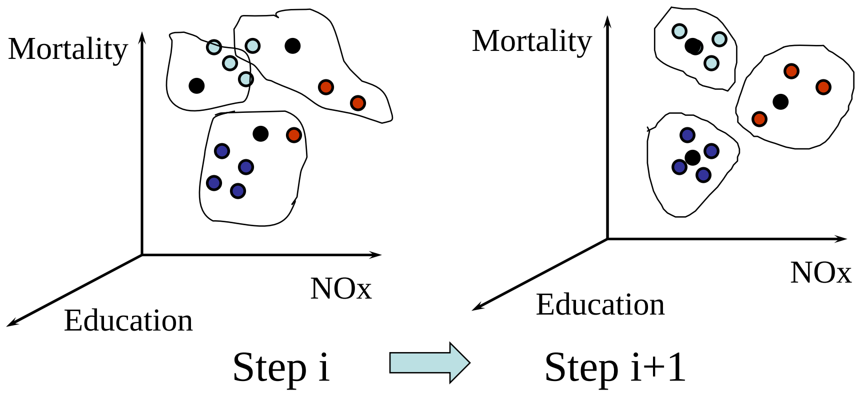
Learning via Clustering



- Form the k=3 “best” clusters in feature space.
- Example of unsupervised learning
  - no prior knowledge needed about classification.

32

K-Means Clustering



- Centroids ● converge to the centroids of the final clusters

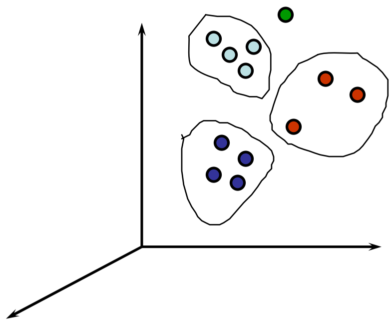
33

K-Means Clustering

- Set i = 0. Choose k centroids a[i, 1], …, a[i, k] in feature space.
- Assign each point in the test set to the nearest centroid (break ties using the lowest index) to form clusters C[1], …, C[k].
- Compute the new centroid a[i+1, j] for each cluster C[j], j=1, …, k.
- Repeat until the centroids converge.

34

Learning via Clustering



To classify ●

1. find nearest cluster

2. classify ● using nearest cluster

- Form the three "best" clusters.
- Example of unsupervised learning
  - no prior knowledge is needed about the classification.
- Use as a basis for subsequent supervised learning.

35

Section 4.3
Trees

For CART trees: L. Breiman, J. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, 1984, Chapman & Hall.

For ACT trees: R. L. Grossman, H. Bodek, D. Northcutt, and H. V. Poor, Data Mining and Tree-based Optimization, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, E. Simoudis, J. Han and U. Fayyad, editors, AAAI Press, Menlo Park, California, 1996, pp 323-326.

36

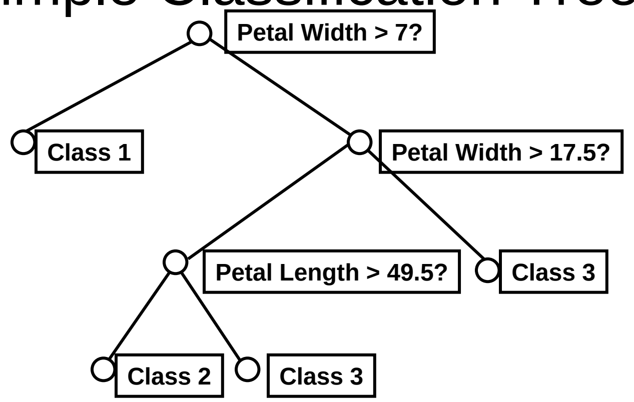
Classification Trees

Petal Len.	Petal Width	Sepal Len.	Sepal Width	Species
02	14	33	50	A
24	56	31	67	C
23	51	31	69	C
13	45	28	57	B

- Want a function Y = g(X), which predicts the red variable Y using one or more of the blue variables X[1], …, X[4]
- Assume each row is classified A, B, or C

37

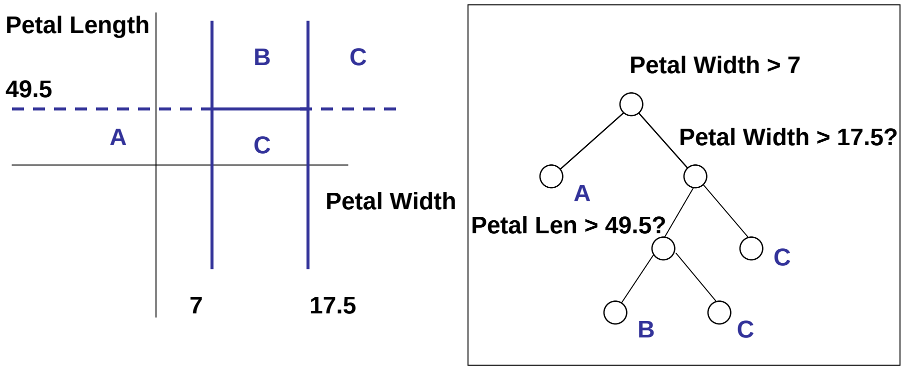
Simple Classification Tree



- Divide feature space into regions
- Use a majority vote to get class A, B, C, etc.

38

Trees Partition Feature Space



- Trees partition the feature space into regions by asking whether an attribute is less than a threshold.

39

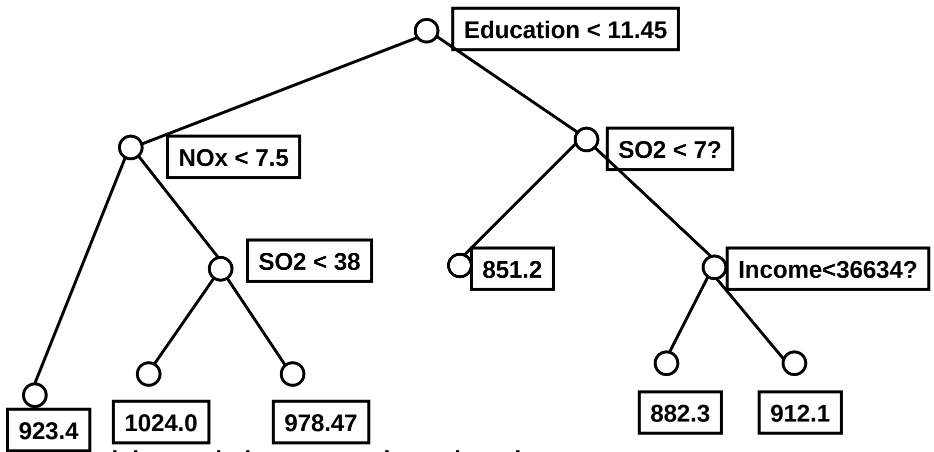
Regression Trees

City	Education	NOx	SO2	Mortality
Akron	11.4	15	59	921.87
Boston	12.1	32	62	934.70
Chicago	10.9	63	278	1024.89
Dallas	11.8	1	1	860.10

- Want a function Y = g(X), which predicts the red variable Y using one or more of the blue variables X[1], …, X[14]

40

Regression Trees



- Divide training sets into buckets.
- Average the dependent variable in each bucket.

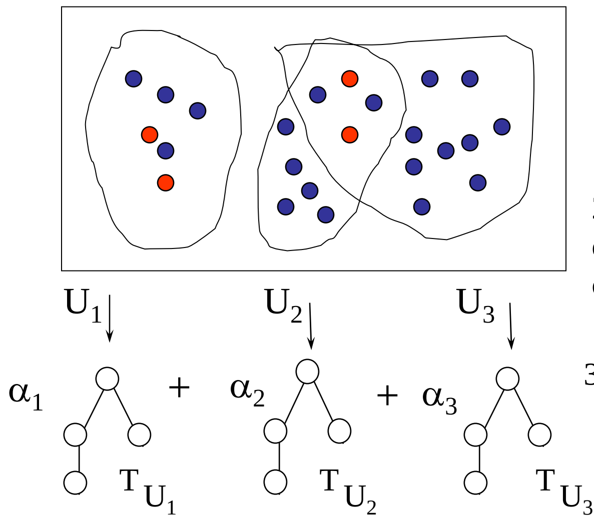
ART and ACT (Averaged Reg. & Class. Trees)

- Define a Cover of the Data. A cover ***U*** of the data x consists of a collection of sets U such that each record is in at least one U.
- Build Trees. Build a tree T_U as usual for the data assigned to each set U in ***U***.
- Average Trees. Fix a finite probability measure α_U on ***U***. Given an object x, ART uses the score:

$$\Sigma\alpha_U \, T_U(x),$$

- This defines an ensemble of trees.

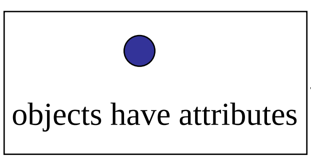
Basic Idea: ART



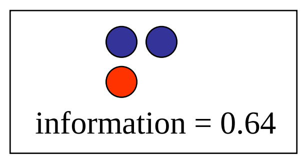
1. Define a cover $\boldsymbol{U} = \{U_1, U_2, U_3\}$ of the data x.
2. Construct a tree T_U on each set U of the cover.
3. Average the trees:

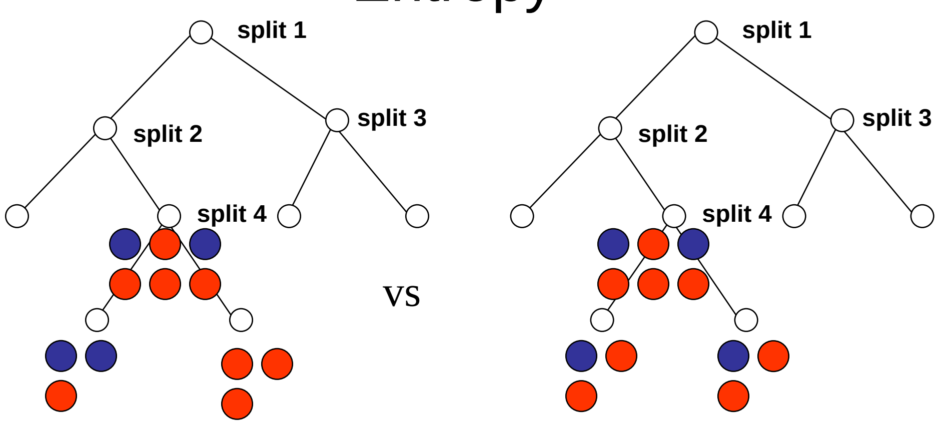
$$\Sigma\, \alpha_j = 1$$
$$\alpha_j > 0$$

43



Split using Entropy

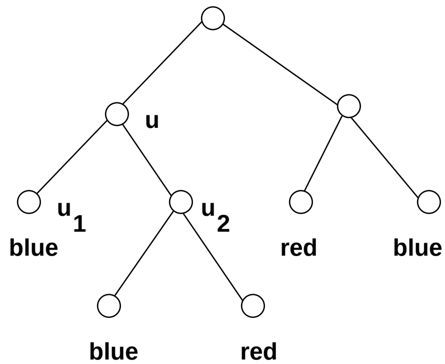




increase in information = 0.32 increase in information = 0

44

Growing the Tree



Step 1. Class proportions.
Node u with n objects
n_1 of class A (red)
n_2 of class B (blue), etc.

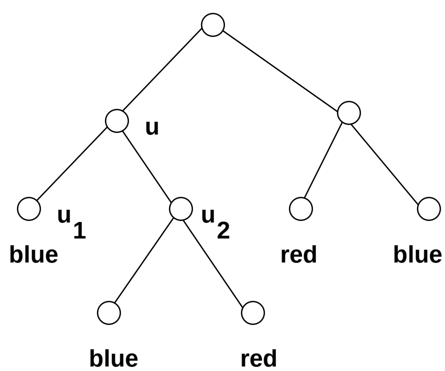
Step 2. Entropy
$I(u) = -\Sigma\, n_j/n \log n_j/n$

Step 3. Split proportions.
m_1 sent to child 1– node u_1
m_2 sent to child 2– node u_2

Step 4. Choose attribute to maximize
$\Delta = I(u) - \Sigma\, m_j/n \; I(u_j)$

45

Split Using GINI Impurity



Step 1. Class proportions.
Node u with n objects
n_1 of class 1 (red)
n_2 of class 2 (blue), etc.

Step 2. Compute Gini Index
$\text{Gini}(u) = 1 - \Sigma\, (n_j/n)^2$

Step 3. Split proportions.
m_1 sent to child 1– node u_1
m_2 sent to child 2– node u_2

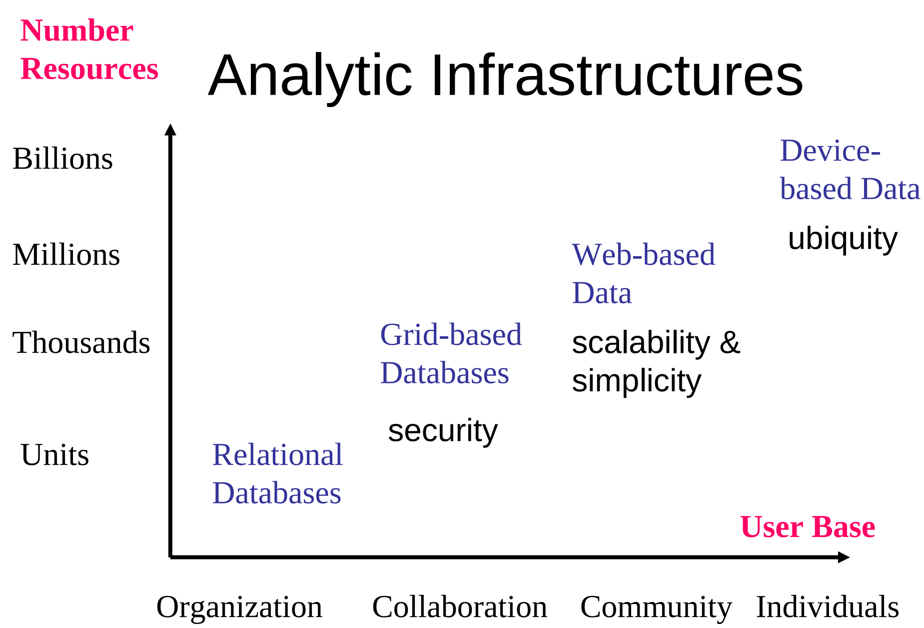
Step 4. Choose split to min
$\text{Gini of Split} = \Sigma\, m_j/n \; \text{Gini}(u_j)$

Section 5

What's Ahead?

47

Analytic Infrastructures



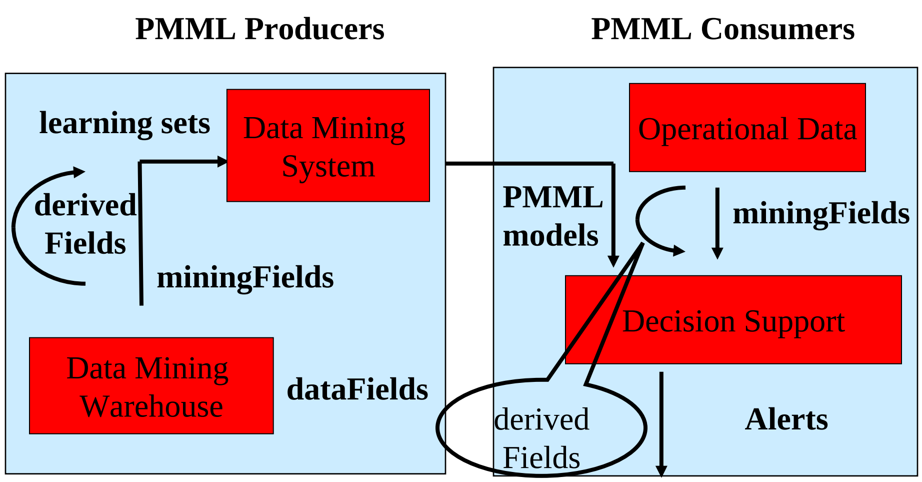
48

Distributed Infrastructures for Data Mining

- Grids built using Globus
- PMML service-based architectures
- Google stacks (GFS, BigTable, Sawzall), Hadoop, etc.
- Data webs (e.g. Swivel, DataSpace)
- Peer to Peer networks (e.g. Sector)

49

PMML Service-Based Architectures for Data Mining



50

For More Information

- www.ncdm.uic.edu (some distributed data mining applications)
- www.dmg.org (PMML)
- sdss.ncdm.uic.edu (Sector)
- www.rgrossman.com (some papers)