

Incremental Algorithms for Missing Data Imputation based on Recursive Partitioning

Claudio Conversano

Department of Economics

University of Cassino,

via M. Mazzaroppi, I-03043 Cassino (FR)

c.conversano@unicas.it,

<http://cds.unina.it/~conversa>

Interface 2003:

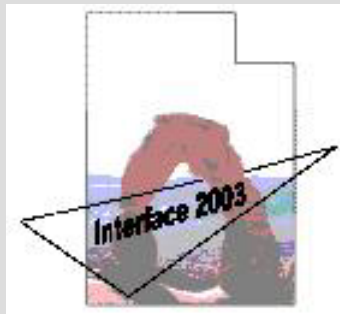
Security and Infrastructure Protection

35th SYMPOSIUM ON THE INTERFACE

Sheraton City Centre

Salt Lake City, Utah

March 12-15, 2003



Outline

- Supervised learning
- Why Trees?
- Trees for Statistical Data Editing
- Examples
- Discussion

Trees in Supervised Learning

Supervised Learning

- Training sample

$$L = \{y, \mathbf{x}_n; n = 1, \dots, N\}$$

from the distribution (Y, \mathbf{X})

➤ Y : *output*

➤ \mathbf{X} : *inputs*

- Decision rule: $d(\mathbf{x}) = y$

Trees

- *Output*

- *Approach* :
Recursive Partitioning

- *Aim*:
Exploration/Decision

- *Steps*:
Growing
Pruning
Testing

Statistical Data Editing

- *Process*: collected data are examined for errors
- *Winkler (2002)*: those methods that can be used to edit (i.e., clean-up) and impute (fill-in) missing or contradictory data”
 - Data Validation
 - Data Imputation
- *How using trees*
 - Incremental Approach for Data Imputation
 - TreeVal for Data Validation

Missing Data: Examples

1. Household surveys (income, savings).
2. Industrial experiment (mechanical breakdowns unrelated to the experimental process).
3. Opinion surveys (people is unable to express a preference for one candidate over another).

Features of Missing Data

Problem

Biased and inefficient estimates

Their relevance is strictly proportional to data dimensionality

Missing Data Mechanisms

- Missing Completely at Random (MCAR)
- Missing at Random (MAR)

Classical Methods

- Complete Case Analysis
- Unconditional Mean
- Hot Deck Imputation

Model Based Imputation

$$\mathbf{y}_{mis} = f(\mathbf{X}_{obs}) + \boldsymbol{\varepsilon}_{obs}$$

Examples:

- Linear Regression (e.g. Little, 1992)
- Logistic Regression (e.g. Vach, 1994)
- Generalized Linear Models (e.g. Ibrahim *et. al*, 1999)
- Nonparametric Regression (e.g. Chu & Cheng, 1995)
- Trees (Conversano & Siciliano, 2002;
Conversano & Cappelli, 2002)

Using Trees in Missing Data Imputation

- Let y_{rs} be the cell presenting a missing input in the r -th row and the s -th column of the matrix \mathbf{X} .
- Any missing input is handled using the tree grown from the learning sample

$$\mathbf{L}_{rs} = \{y_i, \mathbf{x}_i^T; i = 1, \dots, r-1\}$$

where $\mathbf{x}_i^T = (\mathbf{x}_{i1}, \dots, \mathbf{x}_{ij}, \dots, \mathbf{x}_{i,s-1})$ denotes completely observed inputs

- The imputed value is $\hat{f}(\mathbf{x}_r) = \hat{y}_s$

Motivations

- Nonparametric approach
- Deals with numerical and categorical inputs
- Computational feasibility
- Considers conditional interactions among inputs
- Derives simple imputation rules

Incremental Approach: key idea

- **Data Pre-Processing**

rearrange columns and rows of the original data matrix

- **Missing Data Ranking**

define a **lexicographical ordering** of the data, that matches the order by value, corresponding to the numbers of missing values occurring in each record

- **Incremental Imputation**

impute iteratively missing data using tree based models

The original data matrix

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1																										
2						■								■							■					
3																										
4																										
5																										
6										■															■	
7						■																				
8	■													■			■									
9																										
10												■														
11				■														■							■	
12						■																■				
13																										
14																										
15							■				■															

0
3
0
0
0
2
1
3
0
1
3
2
0
0
2

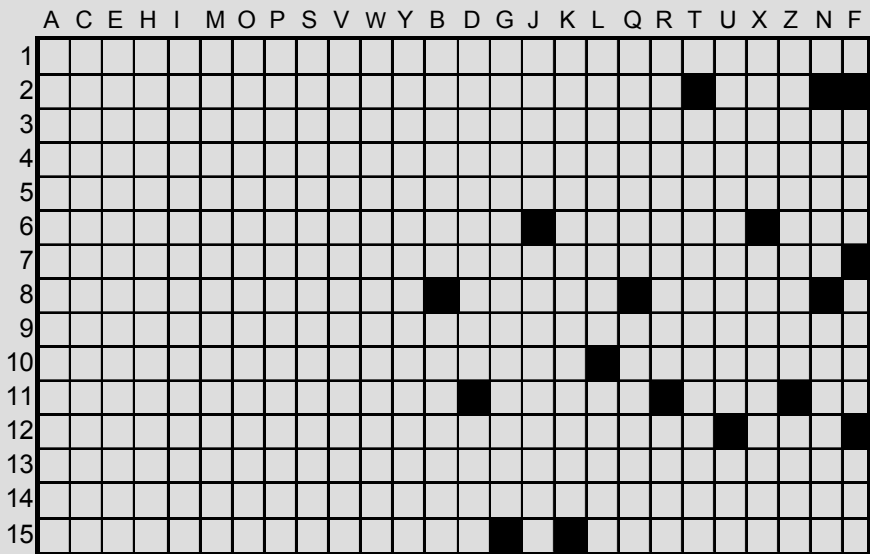
Number of missing values in each row

0 1 0 1 0 3 1 0 0 1 1 1 0 2 0 0 1 1 0 1 1 0 0 1 0 1

Number of missing values in each column

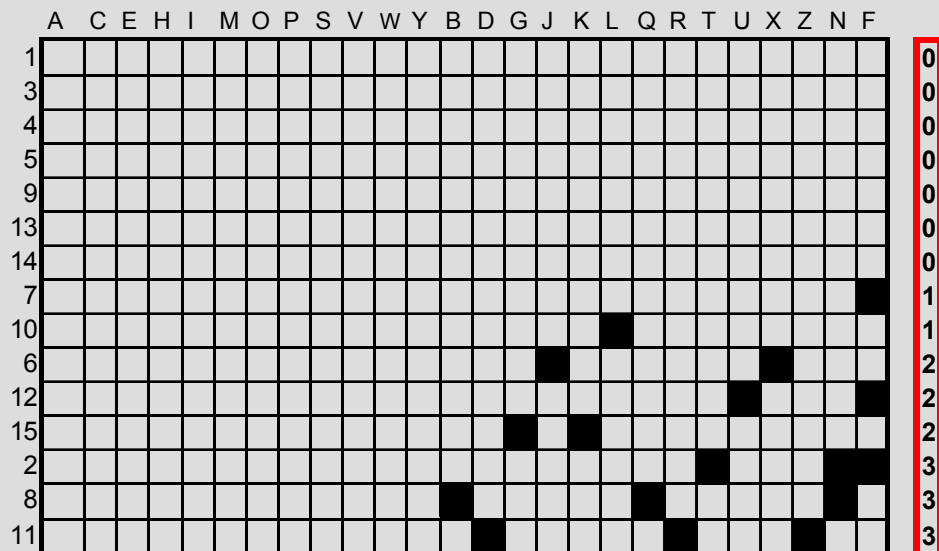
Data re-arrangement

by number of missing in each column



0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2

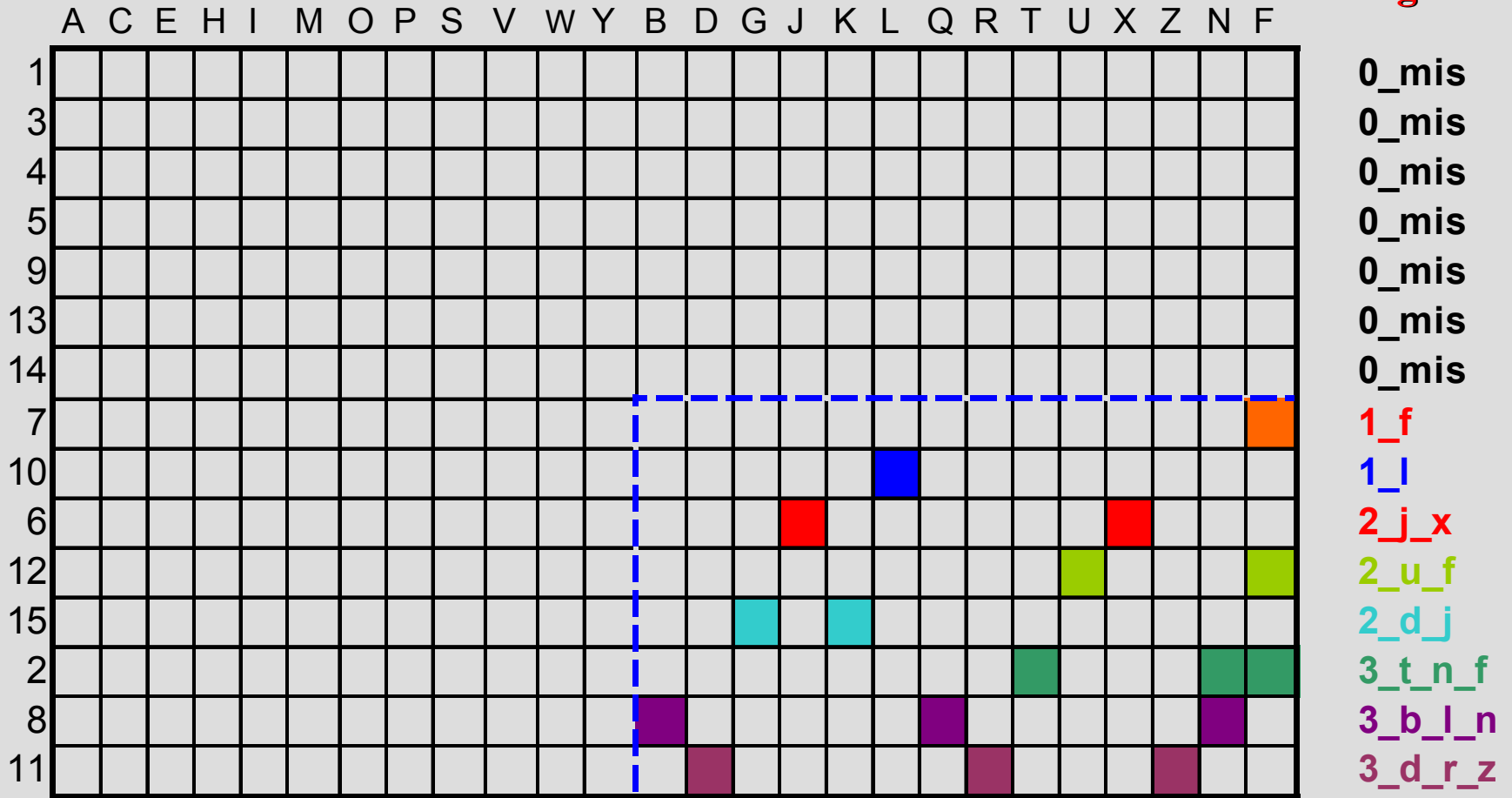
by number of missing in each row



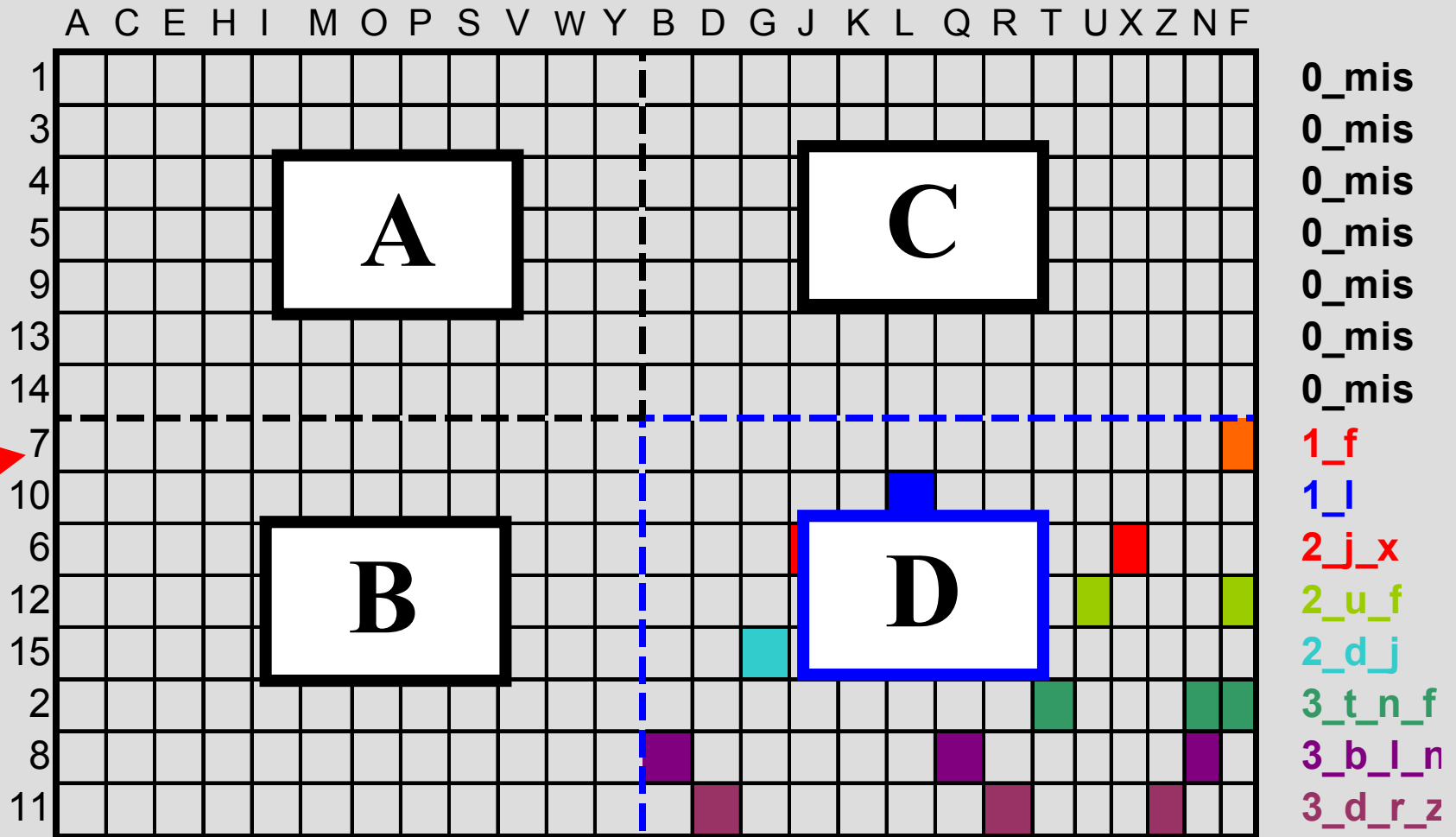
0
0
0
0
0
0
0
0
1
1
2
2
2
2
3
3
3

Missing Data Ranking

**Lexicographical
ordering**



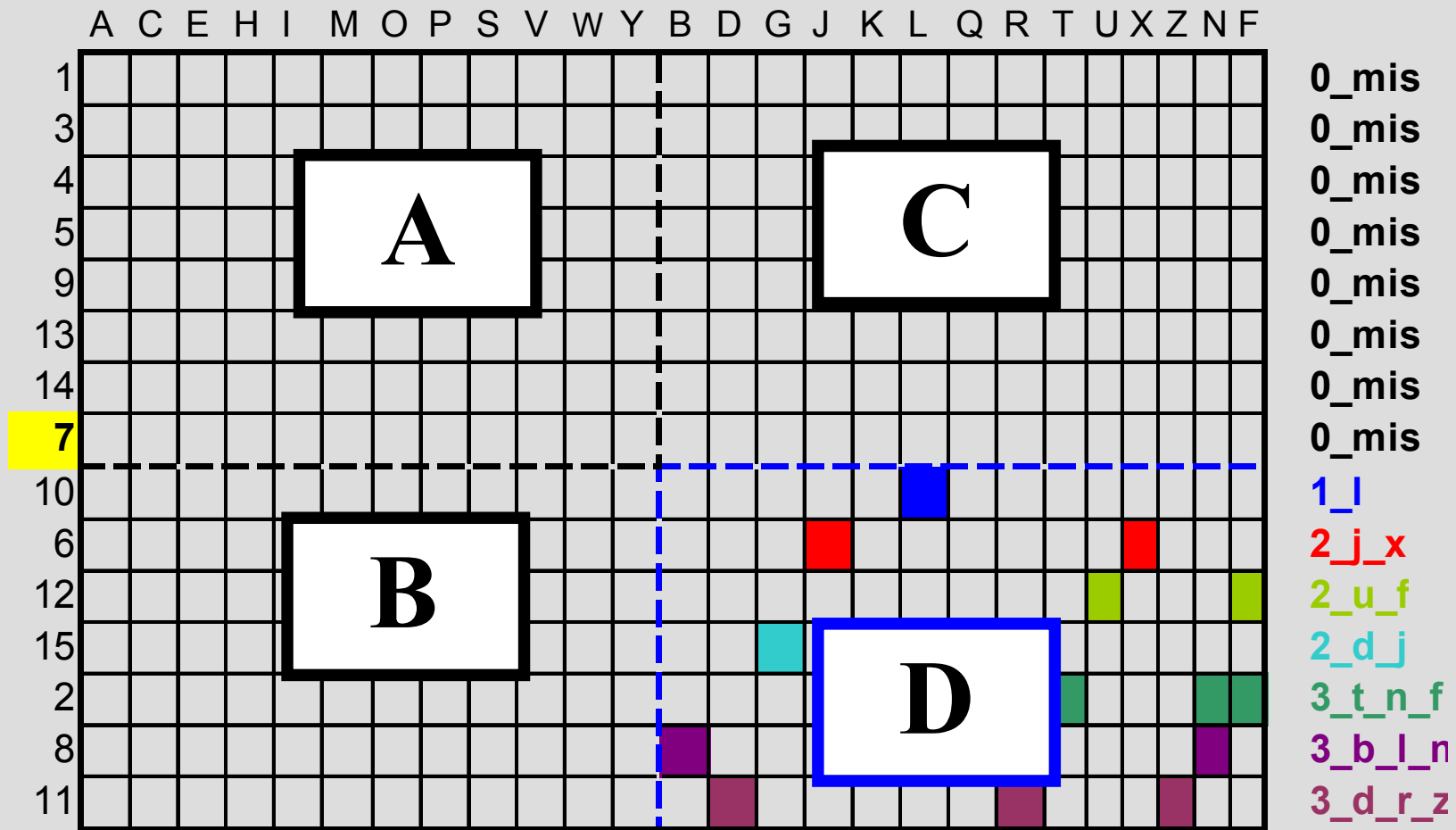
The working matrices



First imputation

D includes 8 missing data types

First Iteration



D includes 7 missing data types

Why Incremental?

The data matrix $\mathbf{X}_{n,p}$
is partitioned in:

$$\mathbf{X}_{n,p} = \begin{bmatrix} \mathbf{A}_{m,d} & \mathbf{C}_{m,p-d} \\ \mathbf{B}_{n-m,d} & \mathbf{D}_{n-m,p-d} \end{bmatrix}$$

where:

A, **B**, **C**: matrix of observed data and imputed data

D: matrix containing missing data

The Imputation is *incremental* because, as it goes on, more and more information is added to the data matrix.

In fact:

- **A**, **B** and **C** are updated in each iteration
- **D** shrinks after each set of records with missing inputs has been filled-in

Simulation Setting

- X_1, \dots, X_p uniform in $[0,10]$
- Data are missing with conditional probability:

$$\psi = \left[1 + \exp(\alpha + \mathbf{X}\boldsymbol{\beta}) \right]^{-1}$$

α being a constant and $\boldsymbol{\beta}$ a vector of coefficients.

- **Goal:** estimate mean and standard deviation of the variable under imputation (in the *numerical response case*), and the expected value π (in the *binary response case*).
- **Compared Methods:**
 - Unconditional Mean Imputation (*UMI*)
 - Parametric Imputation (*PI*)
 - Non Parametric Imputation (*NPI*)
 - Incremental Non Parametric Imputation (*INPI*)

Numerical Response

<i>Data</i>	<i>n</i>	<i>p</i>	<i>Missing variables</i>
<i>sim1.n</i>	500	3	$Y \approx N(-3 + 0.7X_1^2 - 0.3X_2^2, \exp(0.3X_1 + 0.1X_2))$
<i>sim2.n</i>	1000	7	$Y \approx N(X_1 - X_2, \exp(0.2X_1 + 0.1X_2))$ $Y \approx N(X_3 - X_4^2, \exp(0.2X_3 + 0.3X_4))$
<i>sim3.n</i>	1000	7	$Y \approx N(X_1 + \exp(X_2), 0.5X_1 + 0.2X_2)$ $Y \approx N(X_3 - \cos(X_4), 0.7X_3 + 0.4X_4)$

Estimated means and variances

	sim1.n	sim2.n		sim3.n	
	$\hat{\mu}$	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\mu}_1$	$\hat{\mu}_2$
TRUE	-639,2	-28,2	38,5	38,3	-27,8
UMI	-760,7	-33,5	26,9	45,2	-33,6
PI	-618,0	-27,4	37,7	37,5	-27,0
NPI	-612,0	-27,6	39,4	38,3	-27,1
INPI	-622,0	-27,7	37,3	38,3	-27,1

	sim1.n	sim2.n		sim3.n	
	$\hat{\sigma}$	$\hat{\sigma}_1$	$\hat{\sigma}_2$	$\hat{\sigma}_1$	$\hat{\sigma}_2$
TRUE	916,5	30,4	31,8	30,2	29,9
UMI	833,5	27,2	29,6	26,1	26,6
PI	934,2	30,8	30,8	31,0	30,9
NPI	904,3	30,1	29,5	29,2	29,2
INPI	908,5	30,4	31,5	30,3	30,1

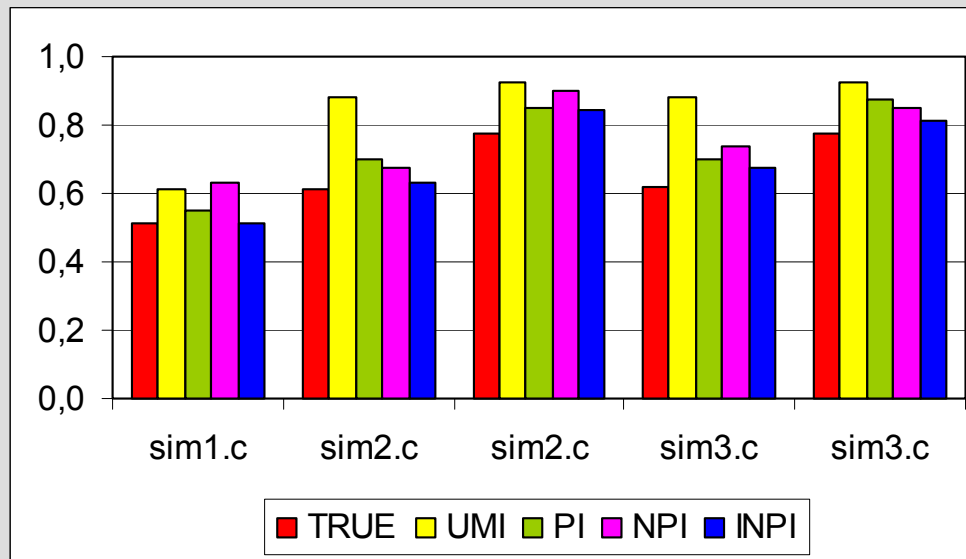
averaged results over 100 independent samples randomly drawn from the original distribution function

Binary Response

<i>Data</i>	<i>n</i>	<i>p</i>	<i>Missing variables</i>
<i>sim1.c</i>	500	3	$Y \approx \text{Bin} \left(n, \frac{\exp(X_1 - X_2)}{1 + \exp(X_1 - X_2)} \right)$
<i>sim2.c</i>	1000	7	$Y \approx \text{Bin} \left(n, [1 + \exp (X_1 - X_2)]^{-1} \right)$ $Y \approx \text{Bin} \left(n, \frac{\exp [\sin (X_3)] + X_4}{1 + \exp [\sin (X_3)] + X_4} \right)$
<i>sim3.c</i>	1000	7	$Y \approx \text{Bin} \left(n, \{1 + \exp [\cos (X_1 - X_2)]\}^{-1} \right)$ $Y \approx \text{Bin} \left(n, \frac{\exp [\sin (X_3)] + X_4}{1 + \exp [\sin (X_3)] + X_4} \right)$

Estimated probabilities

	sim1.c	sim2.c		sim3.c	
	$\hat{\pi}$	$\hat{\pi}_1$	$\hat{\pi}_2$	$\hat{\pi}_1$	$\hat{\pi}_2$
TRUE	0,510	0,610	0,775	0,616	0,775
UMI	0,610	0,884	0,923	0,883	0,924
PI	0,551	0,699	0,851	0,700	0,876
NPI	0,629	0,677	0,897	0,740	0,849
INPI	0,514	0,633	0,845	0,676	0,813



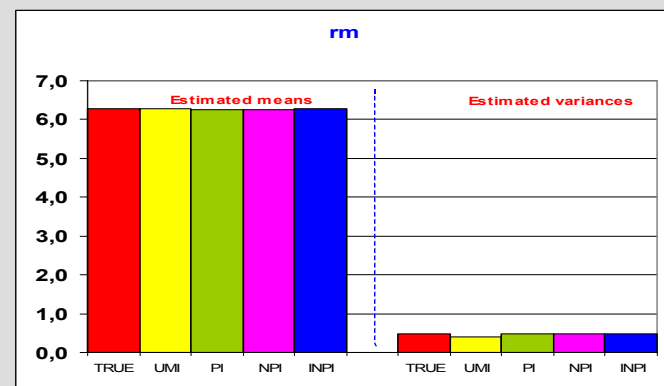
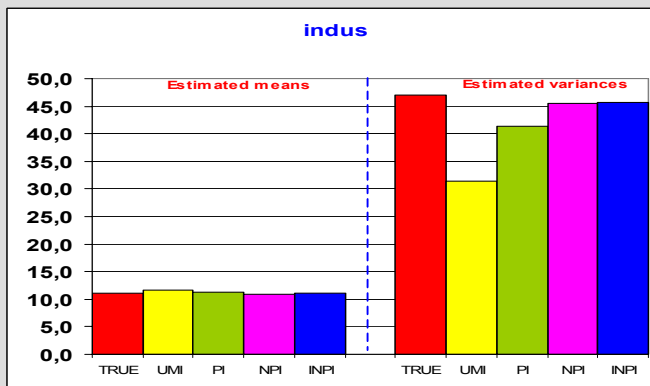
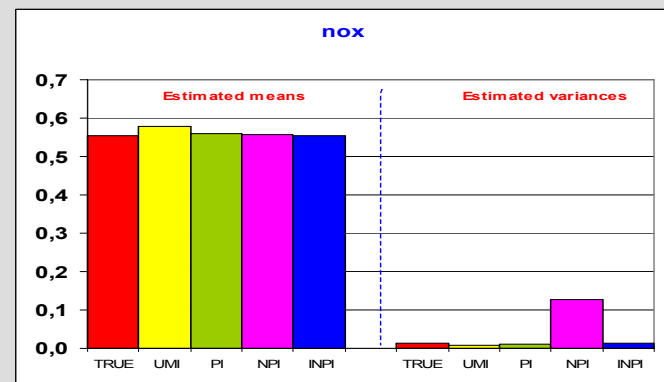
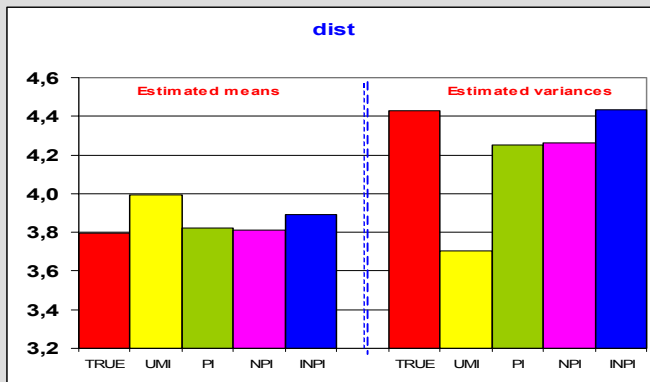
averaged results over 100 independent samples randomly drawn from the original distribution function

Evidence from Real Data

- Source: UCI Machine Learning Repository
- **Boston Housing Data**
 - 506 instances, 13 real valued and 1 binary attributes
 - Variables under imputation
 - distances to 5 employment centers (*dist*, 28%)
 - nitric oxide concentration (*nox*, 32%)
 - proportion of non-retail business acres per town (*indus*, 33%)
 - n. rooms per dwelling (*rm*, 24%)
- **Mushroom Data**
 - 8124 instances, 22 nominally valued attributes
 - Variables under imputation
 - *cap-surface* (4 classes, 3%)
 - *gill-size* (binary, 6%)
 - *stalk-shape* (binary, 12%)
 - *ring-number* (3 classes, 19%)

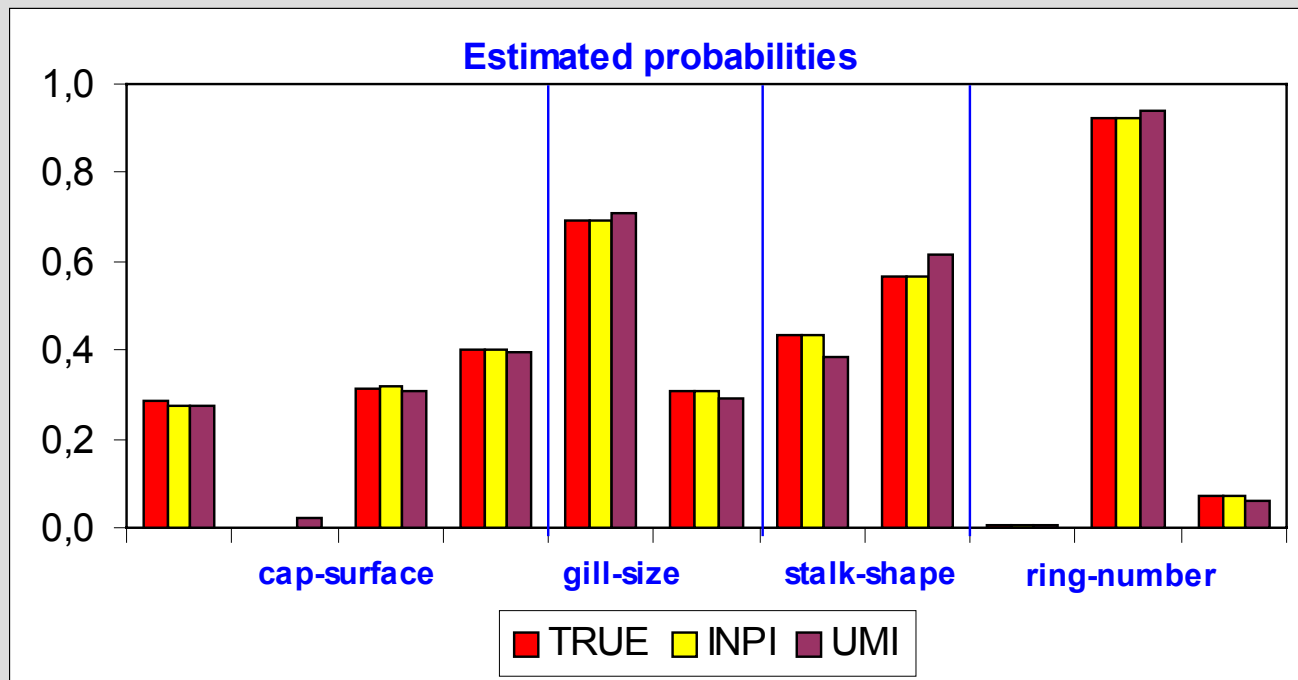
Results for the Boston Housing

	Estimated means				Estimated variances			
	dist	nox	indus	rm	dist	nox	indus	rm
TRUE	3,795	0,555	11,136	6,285	4,434	0,013	47,064	0,494
UMI	3,993	0,579	11,659	6,276	3,703	0,009	31,374	0,389
PI	3,823	0,559	11,228	6,243	4,250	0,012	41,439	0,470
NPI	3,810	0,557	10,919	6,263	4,263	0,126	45,416	0,468
INPI	3,893	0,555	11,051	6,279	4,436	0,013	45,634	0,486



Results for the Mushroom data

	cap-surface				gill-size		stalk-shape		ring-number		
	$\hat{\pi}_1$	$\hat{\pi}_2$	$\hat{\pi}_3$	$\hat{\pi}_4$	$\hat{\pi}_1$	$\hat{\pi}_2$	$\hat{\pi}_1$	$\hat{\pi}_2$	$\hat{\pi}_1$	$\hat{\pi}_2$	$\hat{\pi}_3$
TRUE	0,286	0,000	0,315	0,399	0,691	0,309	0,433	0,567	0,004	0,922	0,074
UMI	0,277	0,021	0,306	0,396	0,710	0,289	0,382	0,618	0,003	0,938	0,059
PI	0,277	0,006	0,324	0,393	0,680	0,320	0,433	0,567	0,003	0,915	0,081
NPI	0,271	0,001	0,316	0,412	0,689	0,311	0,438	0,562	0,004	0,920	0,076
INPI	0,277	0,001	0,319	0,403	0,690	0,310	0,433	0,567	0,004	0,922	0,073



Data Validation

- Accounts for logical inconsistencies in the data
- *Validation Rules*: logical statements about data aimed to find all significant error that may occur.
 - Internal consistency: all rules must not contradict each other.
- *Classical approach*: a subject matter expert defines rules based on the experience.
 - In large surveys it's easy to produce conflicting rules.

Specification of Edits and Validation

- Abstract data model
 - Experts coherence detection
- *Intrinsic* coherence induction
 - TREEVAL
 - **Aim:** *to define validation rules automatically*
 - **Assumption:** increasing order of complexity cannot be handled by experts
 - **Key idea:** to provide an inductive approach to data editing based on trees

TreeVal Method

- Inputs:
 - A learning sample with cross-validation
(to grow and select the tree for each variable)
 - A validation sample
(to check for inconsistencies in the data)
- Steps:
 - Pre-processing: Prior partition of objects
 - TREE: FAST Automated rules detection
 - VAL: Rules validation through divergence measures

Tree Step

- Apply recursive partitioning for each variable (playing the role of response) using the learning sample and select final tree by cross-validation
- Obtain a set of production rules
- Rank production rules based on their **reliability**
(in terms of the impurity reduction when passing from the root node to one of the terminal nodes)
 - Strong Rules
 - Middle Rules
 - Weak Rules

Val Step

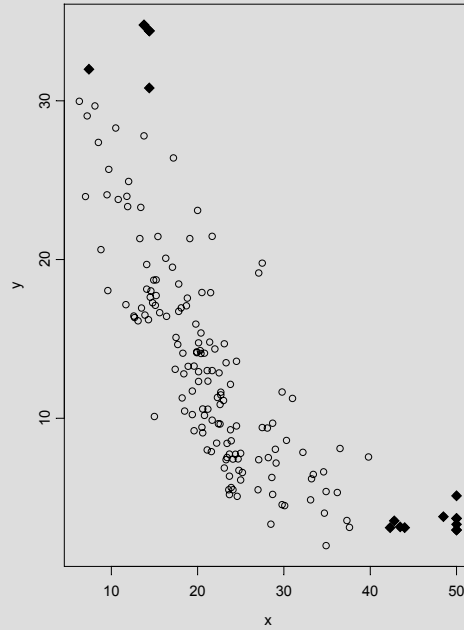
- Each tree generates a distribution of conditional means
- Each observation of the validation sample is compared with the distributions of conditional means
- For a given observation, error may occur when the observed value is far from where the majority of cases is supposed to fall in

An Example

Learning Sample



Validation Sample

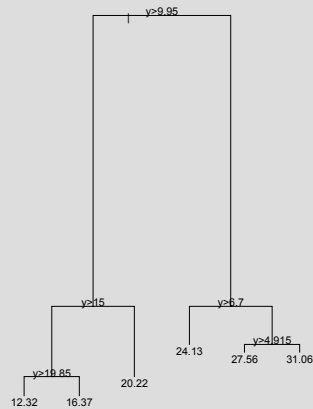
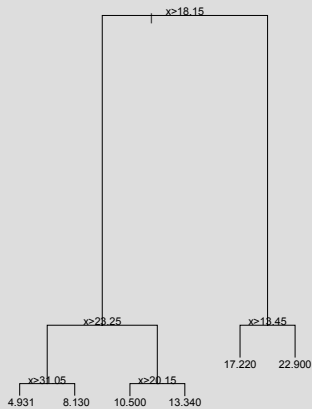


N=500

N=200

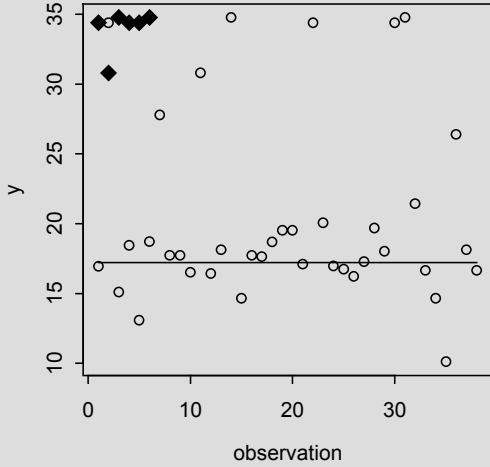
Errors: $x > 40, y = 30$

Errors: 18

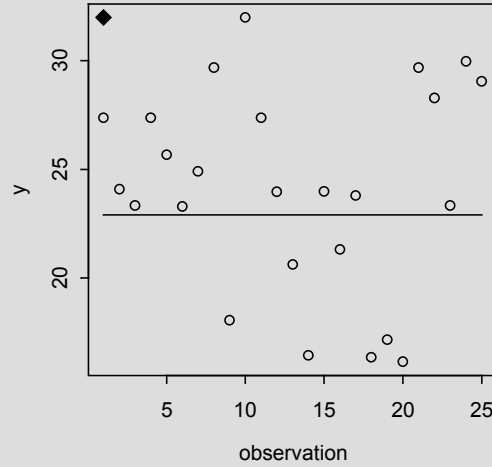


Error Localization

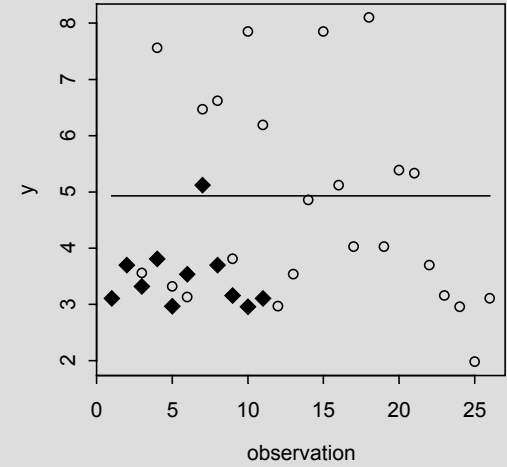
Tree 1, node 6



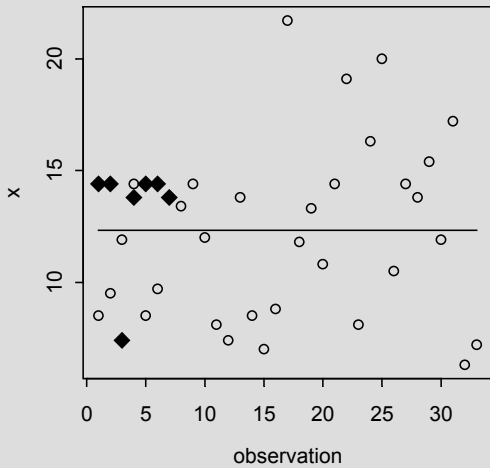
Tree 1, node 7



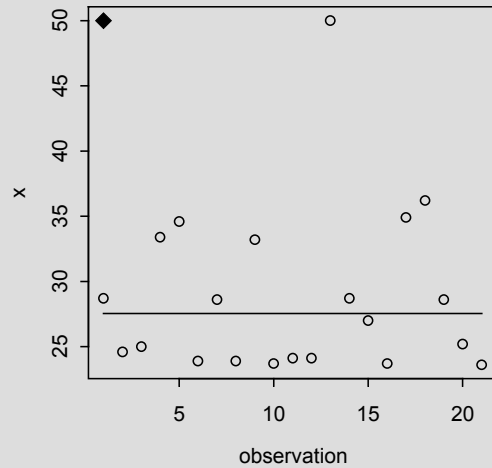
Tree 1, node 8



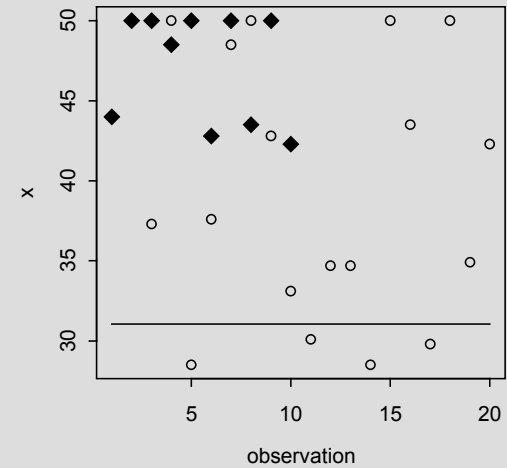
Tree 2, node 8



Tree 2, node 14



Tree 2, node 15



Error Localization (2)

y	x	node	error localization	node	error localization
50.00	2.96	8	no	15	yes
50.00	2.97	8	no	15	yes
50.00	3.32	8	no	15	yes
50.00	3.70	8	no	15	yes
50.00	3.70	8	no	15	yes
50.00	5.12	8	no	14	yes
48.50	3.81	8	no	15	yes
44.00	3.11	8	no	15	yes
43.50	3.16	8	no	15	yes
42.80	3.54	8	no	15	yes
42.30	3.11	8	no	15	yes
14.40	30.81	6	yes	8	no
14.40	34.41	6	yes	8	no
14.40	34.41	6	yes	8	no
14.40	34.41	6	yes	8	no
13.80	34.77	6	yes	8	no
13.80	34.77	6	yes	8	no
7.40	31.99	7	yes	8	no

Evidence from real data

- [Portuguese Survey on Turnover](#) (54,257 instances, 14 attributes)
Source: I.N.E. Statistical Institute of Portugal
- tax: Enterprise tax registry identification number.
- act: Activity indication (whether the enterprise was active during the reference month).
- tot.turn: Total turnover.
- turn.port: Turnover from sales in Portugal.
- turn.intra: Turnover from exports to other EU member states.
- turn.extra: Turnover from exports to non-EU countries.
- sales1: Sales of goods purchased for resale in the same condition as received.
- sales2: Sales of products manufactured by the enterprise.
- services: “Sales” of services.
- n.workers: Number of employees.
- tot.wages: Total wages.
- wage.pay: Wage payments in arrears.
- mh.work: Total man-hours worked.
- nace: NACE code of the enterprise’s activity.

A specific set of validation rules

Sector 1: Response Variable : tot.turn					
node number	n	yval	s	gain	rule
1	2.219	75.974,04	760.627.945		
17	1.517	19.559,13	28.119.832	3,697	turn.port<137130 & sales2<37741 & n.workers<176.5
32	637	77.622,15	19.967.842	2,625	sales2<186541 & sales2>37741 & turn.port<137130
9	64	297.959,05	10.550.534	1,387	sales2<186541 & turn.port>137130
7	5	4.597.005,40	4.140.749	0,544	sales2>3.63091e+006
33	5	-366.945,06	3.786.890	0,498	turn.port<137130 & sales2<37741 & n.workers>176.5
13	11	3.139.405,64	2.970.230	0,39	sales2<3.63091e+006 & sales2>2.71052e+006
5	5	1.064.322,60	2.907.539	0,382	sales2<1.7987e+006 & sales2>186541
12	5	2.360.258,80	1.144.085	0,15	sales2>1.7987e+006 & sales2<2.71052e+006

Task: Compare each observation of the validation sample with the distributions of conditional means derived from each tree.

Dealing with Validation Rules

Classification of validation rules

- a) Strong Rules: gain lower 5%;
- b) Middle Rules: gain between 5% and 10%;
- c) Weak Rules: gain greater than 10%.

Examples of strong rules

Node 32	turn.port<137130 \cap sales2<37741 \cap n.workers<176.5	Gain = 3,697
Node 17	sales2>37741 \cap sales2<186541 \cap turn.port<137130	Gain = 2,625

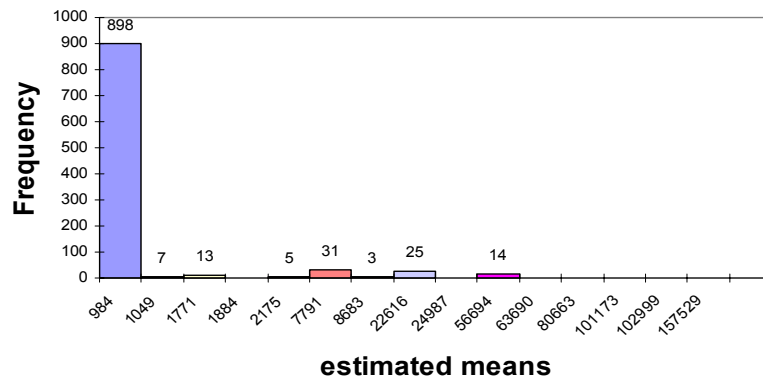
Conditional Means distribution

node	node	node	node	node	node	node	node
33	32	17	9	5	12	13	7
-366945,6	19559,13	77622,55	295972,1	1064323	2360258,8	3139406	4597005

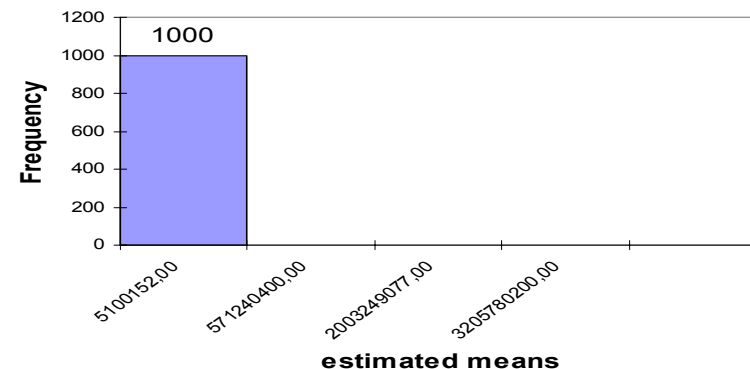
Detection of Logical Errors

Response	Strongest Rule	n. of possible inconsistencies
tot.turn	turn.port<137130 & sales2<37741 n.workers<176.5	31
turn.prot	services<100334 & tot.turn<19608	67
turn.intr	tot.turn<1.4974	0
turn.extr	tot.turn<653313 & n.workers<304	79
sales1	services<334463 & tot.turn<51880	88
sales2	tot.turn<853212 & tot.turn>45899	90
service	turn.port<89341 & sales2>18.5	14
n.worker	turn.port<89341	96
tot.wage	mh.work<112716 & n.workers<105.5	7
wege.pay	turn.extra<1.2628	24
mh.work	tot.wages<152402 & n.workers<45 & n.workers>24	0

**Validation Rules for Sector 1:
Response: sales1, leaf number: 8**



**Validation Rules for Sector 1:
Response: turn.intra, leaf number: 2**



Concluding Remarks

Incremental Approach for Missing Data Imputation

- Results are encouraging when dealing with nonlinear data with non constant variance
- The resulting loss of information is retrieved by the proposed incremental approach

TreeVal for Data Validation

- Trees can be fruitfully used for validation purposes (joining the subject matter expert opinions)
- Attention must be paid to instability of trees and to the relative simplicity of the model (future work)
- **Challenge: Learning with Information Retrieval**

The INSPECTOR Project

Quality in the Statistical Information Life - Cycle:

A Distributed System for Data Validation



Project Partners

website:www.liaison.gr/project/inspector

- Intrsoft Ltd. (Athens, Greece)
- Liaison Systems Ltd. (Athens, Greece)
- Statistical Institute of Greece
- Statistical Institute of Portugal
- University of Naples (Italy)
- University of Vien (Austria)