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.co
nversano@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
  \[ L = \{ y, x_n ; n = 1, \ldots, N \} \]
  from the distribution \((Y, X)\)
  ➢ \( Y \): output
  ➢ \( X \): inputs
• Decision rule: \( d(x) = y \)

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

\[ y_{mis} = f(X_{obs}) + \varepsilon_{obs} \]

**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 \( X \).

• Any missing input is handled using the tree grown from the learning sample

\[
L_{rs} = \{ y_p, x_{iT} ; i = 1, \ldots, r-1 \}
\]

where \( x_{iT} = (x_{i1}, \ldots, x_{ij}, \ldots, x_{i,s-1}) \) denotes completely observed inputs

• The imputed value is \( \hat{f}(x_r) = \hat{y}_s \)
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 | 0 | 1 | 0 | 1 | 0 | 3 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

Number of missing values in each column

Number of missing values in each row

0 3 0 0 0 2 1 3 0 1 3 2 0 0 2
Data re-arrangement

by number of missing in each column

by number of missing in each row
Missing Data Ranking

Lexicographical ordering
The working matrices

D includes 8 missing data types

First imputation
D includes 7 missing data types
Why Incremental?

The data matrix $X_{n,p}$ is partitioned in:

$$
X_{n,p} = \begin{bmatrix}
A_{m,d} & C_{m,p-d} \\
B_{n-m,d} & D_{n-m,p-d}
\end{bmatrix}
$$

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

- $X_1, \ldots, X_p$ uniform in $[0,10]$
- Data are missing with conditional probability:
  \[
  \psi = \left[1 + \exp(\alpha + X\beta)\right]^{-1}
  \]
  \[\alpha\] being a constant and \[\beta\] a vector of coefficients.

- **Goal:** estimate mean and standard deviation of the variable under imputation (in the numerical response case), and the expected value $\pi$ (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

<table>
<thead>
<tr>
<th>Data</th>
<th>n</th>
<th>p</th>
<th>Missing variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim1.n</td>
<td>500</td>
<td>3</td>
<td>$Y \approx N\left(-3 + 0.7X_1^2 - 0.3X_2^2, \exp(0.3X_1 + 0.1X_2)\right)$</td>
</tr>
<tr>
<td>sim2.n</td>
<td>1000</td>
<td>7</td>
<td>$Y \approx N\left(X_1 - X_2, \exp(0.2X_1 + 0.1X_2)\right)$ $Y \approx N\left(X_3 - X_4^2, \exp(0.2X_3 + 0.3X_4)\right)$</td>
</tr>
<tr>
<td>sim3.n</td>
<td>1000</td>
<td>7</td>
<td>$Y \approx N\left(X_1 + \exp(X_2), 0.5X_1 + 0.2X_2\right)$ $Y \approx N\left(X_3 - \cos(X_4), 0.7X_3 + 0.4X_4\right)$</td>
</tr>
</tbody>
</table>
Estimated means and variances

<table>
<thead>
<tr>
<th>sim1.n</th>
<th>sim2.n</th>
<th>sim3.n</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{\mu} )</td>
<td>( \hat{\mu}_1 )</td>
<td>( \hat{\mu}_2 )</td>
</tr>
<tr>
<td>TRUE</td>
<td>-639,2</td>
<td>-28,2</td>
</tr>
<tr>
<td>UMI</td>
<td>-760,7</td>
<td>-33,5</td>
</tr>
<tr>
<td>PI</td>
<td>-618,0</td>
<td>-27,4</td>
</tr>
<tr>
<td>NPI</td>
<td>-612,0</td>
<td>-27,6</td>
</tr>
<tr>
<td>INPI</td>
<td>-622,0</td>
<td>-27,7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sim1.n</th>
<th>sim2.n</th>
<th>sim3.n</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{\sigma} )</td>
<td>( \hat{\sigma}_1 )</td>
<td>( \hat{\sigma}_2 )</td>
</tr>
<tr>
<td>TRUE</td>
<td>916,5</td>
<td>30,4</td>
</tr>
<tr>
<td>UMI</td>
<td>833,5</td>
<td>27,2</td>
</tr>
<tr>
<td>PI</td>
<td>934,2</td>
<td>30,8</td>
</tr>
<tr>
<td>NPI</td>
<td>904,3</td>
<td>30,1</td>
</tr>
<tr>
<td>INPI</td>
<td>908,5</td>
<td>30,4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Data</th>
<th>n</th>
<th>p</th>
<th>Missing variables</th>
</tr>
</thead>
</table>
| sim1.c | 500 | 3 | \[
Y \approx Bin \left( n, \frac{\exp(X_1 - X_2)}{1 + \exp(X_1 - X_2)} \right) \]
| sim2.c | 1000 | 7 | \[
Y \approx Bin \left( n, \left[ 1 + \exp \left( X_1 - X_2 \right) \right]^{-1} \right) \\
Y \approx Bin \left( n, \frac{\exp \left[ \sin \left( X_3 \right) \right] + X_4}{1 + \exp \left[ \sin \left( X_3 \right) \right] + X_4} \right) \]
| sim3.c | 1000 | 7 | \[
Y \approx Bin \left( n, \left\{ 1 + \exp \left[ \cos \left( X_1 - X_2 \right) \right] \right\}^{-1} \right) \\
Y \approx Bin \left( n, \frac{\exp \left[ \sin \left( X_3 \right) \right] + X_4}{1 + \exp \left[ \sin \left( X_3 \right) \right] + X_4} \right) \]
## Estimated probabilities

<table>
<thead>
<tr>
<th></th>
<th>sim1.c</th>
<th>sim2.c</th>
<th>sim3.c</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\hat{\pi}$</td>
<td>$\hat{\pi}_1$</td>
<td>$\hat{\pi}_2$</td>
</tr>
<tr>
<td><strong>TRUE</strong></td>
<td>0,510</td>
<td>0,610</td>
<td>0,775</td>
</tr>
<tr>
<td><strong>UMI</strong></td>
<td>0,610</td>
<td>0,884</td>
<td>0,923</td>
</tr>
<tr>
<td><strong>PI</strong></td>
<td>0,551</td>
<td>0,699</td>
<td>0,851</td>
</tr>
<tr>
<td><strong>NPI</strong></td>
<td>0,629</td>
<td>0,677</td>
<td>0,897</td>
</tr>
<tr>
<td><strong>INPI</strong></td>
<td>0,514</td>
<td>0,633</td>
<td>0,845</td>
</tr>
</tbody>
</table>

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

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

**Estimated means**

<table>
<thead>
<tr>
<th></th>
<th>dist</th>
<th>nox</th>
<th>indus</th>
<th>rm</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>3,2</td>
<td>0,0</td>
<td>0,5</td>
<td>0,1</td>
</tr>
<tr>
<td>UMI</td>
<td>3,4</td>
<td>0,1</td>
<td>5,0</td>
<td>0,2</td>
</tr>
<tr>
<td>PI</td>
<td>3,6</td>
<td>0,2</td>
<td>10,0</td>
<td>0,3</td>
</tr>
<tr>
<td>NPI</td>
<td>3,8</td>
<td>0,3</td>
<td>15,0</td>
<td>0,4</td>
</tr>
<tr>
<td>INPI</td>
<td>4,0</td>
<td>0,4</td>
<td>20,0</td>
<td>0,5</td>
</tr>
</tbody>
</table>

**Estimated variances**

<table>
<thead>
<tr>
<th></th>
<th>dist</th>
<th>nox</th>
<th>indus</th>
<th>rm</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>3,3</td>
<td>0,0</td>
<td>0,5</td>
<td>0,1</td>
</tr>
<tr>
<td>UMI</td>
<td>3,4</td>
<td>0,1</td>
<td>5,0</td>
<td>0,2</td>
</tr>
<tr>
<td>PI</td>
<td>3,6</td>
<td>0,2</td>
<td>10,0</td>
<td>0,3</td>
</tr>
<tr>
<td>NPI</td>
<td>3,8</td>
<td>0,3</td>
<td>15,0</td>
<td>0,4</td>
</tr>
<tr>
<td>INPI</td>
<td>4,0</td>
<td>0,4</td>
<td>20,0</td>
<td>0,5</td>
</tr>
</tbody>
</table>

### Diagrams

- **dist**
- **nox**
- **indus**
- **rm**
## Results for the Mushroom data

<table>
<thead>
<tr>
<th>cap-surface</th>
<th>gill-size</th>
<th>stalk-shape</th>
<th>ring-number</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>0.000</td>
<td>0.315</td>
<td>0.399</td>
</tr>
<tr>
<td>UMI</td>
<td>0.277</td>
<td>0.021</td>
<td>0.306</td>
</tr>
<tr>
<td>PI</td>
<td>0.277</td>
<td>0.006</td>
<td>0.324</td>
</tr>
<tr>
<td>NPI</td>
<td>0.271</td>
<td>0.001</td>
<td>0.316</td>
</tr>
<tr>
<td>INPI</td>
<td>0.277</td>
<td>0.001</td>
<td>0.319</td>
</tr>
</tbody>
</table>

### Estimated probabilities

- **TRUE**: Red
- **INPI**: Yellow
- **UMI**: Purple
Data Validation

- Accounts for logical inconsistencies in the data

- **Validation Rules**: logical statements about data aimed to find all significant errors 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 rote 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 fell 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)

<table>
<thead>
<tr>
<th>y</th>
<th>x</th>
<th>node</th>
<th>error localization</th>
<th>node</th>
<th>error localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.00</td>
<td>2.96</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>50.00</td>
<td>2.97</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>50.00</td>
<td>3.32</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>50.00</td>
<td>3.70</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>50.00</td>
<td>3.70</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>50.00</td>
<td>5.12</td>
<td>8</td>
<td>no</td>
<td>14</td>
<td>yes</td>
</tr>
<tr>
<td>48.50</td>
<td>3.81</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>44.00</td>
<td>3.11</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>43.50</td>
<td>3.16</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>42.80</td>
<td>3.54</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>42.30</td>
<td>3.11</td>
<td>8</td>
<td>no</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>14.40</td>
<td>30.81</td>
<td>6</td>
<td>yes</td>
<td>8</td>
<td>no</td>
</tr>
<tr>
<td>14.40</td>
<td>34.41</td>
<td>6</td>
<td>yes</td>
<td>8</td>
<td>no</td>
</tr>
<tr>
<td>14.40</td>
<td>34.41</td>
<td>6</td>
<td>yes</td>
<td>8</td>
<td>no</td>
</tr>
<tr>
<td>14.40</td>
<td>34.77</td>
<td>6</td>
<td>yes</td>
<td>8</td>
<td>no</td>
</tr>
<tr>
<td>13.80</td>
<td>34.77</td>
<td>6</td>
<td>yes</td>
<td>8</td>
<td>no</td>
</tr>
<tr>
<td>7.40</td>
<td>31.99</td>
<td>7</td>
<td>yes</td>
<td>8</td>
<td>no</td>
</tr>
</tbody>
</table>
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

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

**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

<table>
<thead>
<tr>
<th>Node</th>
<th>Condition</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>turn.port&lt;137130 (\cap) sales2&lt;37741 (\cap) n.workers&lt;176.5</td>
<td>3,697</td>
</tr>
<tr>
<td>17</td>
<td>sales2&gt;37741 (\cap) sales2&lt;186541 (\cap) turn.port&lt;137130</td>
<td>2,625</td>
</tr>
</tbody>
</table>

Conditional Means distribution

<table>
<thead>
<tr>
<th>node</th>
<th>node</th>
<th>node</th>
<th>node</th>
<th>node</th>
<th>node</th>
<th>node</th>
<th>node</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>32</td>
<td>17</td>
<td>9</td>
<td>5</td>
<td>12</td>
<td>13</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>-366945.6</td>
<td>19559.13</td>
<td>77622.55</td>
<td>295972.1</td>
<td>1064323</td>
<td>2360258.8</td>
<td>3139406</td>
<td>4597005</td>
<td></td>
</tr>
</tbody>
</table>
Detection of Logical Errors

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

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

Project Partners

- 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)

website: www.liaison.gr/project/inspector