Mathematical Models for Feature Selection
And their Application
In Bioinformatics

Paola Bertolazzi, Giovanni Felici
Istituto di Analisi dei Sistemi ed Informatica IASI-CNR

Paola Festa
Dipartimento di Matematica e Applicazioni R.M. Caccioppoli, Università
degli Studi di Napoli Federico II
Summary

Logic Data Mining System: online dmb.iasi.cnr.it

Focus on:
Formulation of the Feature Selection Problem
GRASP Methods
Applications
The Logic Data Mining flow

**RAW DATA**
Samples from more classes, data in any format

**DISCRETIZATION**
Identify significant thresholds for the value of rational variables; intervals generate discrete variables, then logic

**FEATURE SELECTION**
Select few logic variables that appear to have a strong capability of telling one class from the other over the whole sample

**LEARNING**
Build logic formulas using the selected variables that are able to classify correctly the training data: 

IF(X&Y) THEN Z
Feature Selection

• FS is a projection of a set of multidimensional points from their original space to a space of smaller dimension with little "loss of information" or large "reduction of noise".

• Information and noise must defined w.r.t. to the objective of the specific application: clustering, classification, synthesis...

• in supervised learning application, we want to preserve or enhance the relative distances between observations belonging to different groups.
FS as a Combinatorial Problem

When the projection of the points is simply a selection of a subset of the available dimensions, the FS problem has a combinatorial nature.

Such fact has been pointed out and exploited already in the literature:


Notations and Definitions

we assume that \( n \) \( m \)-dimensional points are the input data for the FS problems. The points are represented in the rational matrix \( A \)

\( M \) (resp. \( N \)) is the index set of the columns of \( A \) (resp. rows); then

\[
n = |N|, \quad m = |M|, \quad A = n \times m, \quad A \in R^{mn}
\]

An appropriate measure of the information contained in \( A \) is given by:

\[
I(A) = \sum_i \sum_{j \neq i} \sum_k \left( a_{ij} - a_{jk} \right)^2
\]

\( I(A) \approx \) the average quadratic distance of the points in \( A \), directly related to the \textbf{variance} expressed by \( A \), a widely used measure in Statistics and Data Analysis.
A Simple Optimization Problem

Consider now the projection of $A$ on a subset of its dimensions $M'$, such that $|M'| = \beta < m$, and

$$x_k = \begin{cases} 1 & \text{if } k \in M' \\ 0 & \text{otherwise} \end{cases}$$

and therefore

$$I_x(A) = \sum_i \sum_{j \neq i} \sum_k (a_{ik} - a_{jk})^2 x_k$$

represents the portion of information preserved by the projection of the points of $A$ on their $M'$ dimensions.

The simplest optimization problem that can be defined would be:

$$\max I_x(A) = \sum_i \sum_{j \neq i} \sum_k (a_{ik} - a_{jk})^2 x_k$$

$$\sum_k x_k \leq \beta$$

$$x_k \in \{0,1\}$$
A (proper) extension: minimization of the infimum-norm

An alternative to the average approach consists in requiring a minimum level of distance between each pair, and requiring a projection that maximizes such level:

\[
\max \alpha \\
\sum_k (a_{ik} - a_{jk})^2 x_k \geq \alpha, \quad \forall i, j, i \neq j \\
\sum_k x_k \leq \beta \\
x_k \in \{0,1\}
\]

Relation between the two models
Let \( h = m \times n \) and \( \Omega \subset \mathbb{R}^n \) be the Euclidean subspace where a point \( \omega \) is defined as follows:

\[
\omega = \{\omega_1, \ldots, \omega_l : \omega_l = \sum_k (a_{ik} - a_{jk})^2, l = i \times (n-1) + j, i \neq j\}
\]

With a proper definition of the projection \( \omega_x \) we have that the 2 models become:

\[
\max \|\omega_x\|^1 \\
\sum_k x_k \leq \beta \\
x \in \{0,1\}^m
\]

\[
\max \|\omega_x\|^{\inf} \\
\sum_k x_k \leq \beta \\
x \in \{0,1\}^m
\]
1) Special Case: Binary Data

Let \( d_{ij}^k = (a_{ik} - a_{jk})^2 \)

If data in A is binary, then \( a_{ij} \in \{0,1\} \Rightarrow d_{ij}^k = \begin{cases} 1 & \text{if } (a_{ik} = a_{jk}) \\ 0 & \text{otherwise} \end{cases} \)

and the FS problem can be rewritten as:

\[
\max \alpha \\
\sum_k d_{ij}^k x_k \geq \alpha, \quad \forall i, j, i \neq j \\
\sum_k x_k \leq \beta \\
x_k \in \{0,1\}
\]

2) Special Case: Supervised Learning

The row vectors of A are partitioned into two different classes

\[
A = \tilde{A} \cup \tilde{B} \\
a_i \in \tilde{A}, c(i) = \tilde{A}, \quad a_i \in \tilde{B}, c(i) = \tilde{B} \\
n_A = |\tilde{A}|, \quad n_B = |\tilde{B}|
\]

\[
\max \alpha \\
\sum_k d_{ij}^k x_k \geq \alpha, \quad \forall i, j, c(i) \neq c(j) \\
\sum_k x_k \leq \beta \\
x_k \in \{0,1\}
\]

Only the distance between points of different classes is taken into account; but the number of constraints is still very large, as it grows quadratically with n.
An example

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

constraint(1,2): \( x_4 + x_5 + x_6 + x_{10} \geq 1 \)

constraint(1,3): \( x_1 + x_6 + x_7 \geq 1 \)

solution with minimal size \((X_6 = 1, X_i = 0, i \neq 6)\)

# constraints proportional to \(N_a \times N_b\)

With \(\beta \leq 2\) the max value of \(\alpha\) is still 1.

We need \(\beta = 3\) for a solution with \(\alpha = 2\)
Variant 1) A Compact Model

Assume the case of supervised learning, and consider the subset of constraints related to row i belonging to class A, and add over the elements of class B:

\[
\sum_k d^k_{ij} x_k \geq \alpha, \forall j, c(j) = \tilde{B} \quad \sum \left( \sum_k d^k_{ij} x_k \geq \alpha \right)
\]

\[
\sum_{j:c(j)=\tilde{B}} \sum_k d^k_{ij} x_k \geq \sum \alpha \approx \sum_k \sum_{j:c(j)\neq\tilde{B}} d^k_{ij} x_k \geq \alpha \times n_B
\]

\[
\tilde{d}_i^k = \sum_{j:c(j)=\tilde{B}} d^k_{ij}, \quad \tilde{d}_i^k = \begin{cases} 1 & k \text{ separates perfectly the 2 classes} \\ 0 & k \text{ is useless for separation} \end{cases}
\]
A Compact Model (2)

The value $\tilde{d}_{ik}$ can be adopted as a direct measure of the importance of column $k$ for row $i$:

$$\tilde{d}_{ik} = \begin{cases} \frac{\tilde{d}_{ik}}{n_B}, i: c(i) = \tilde{A} \\ \frac{\tilde{d}_{ik}}{n_A}, i: c(i) = \tilde{B} \end{cases}$$

$$\tilde{f}_{ik} = \left\lfloor f_{ik} + \lambda \right\rfloor$$

$$\max \alpha \quad \sum_k \tilde{f}_{ik} x_k \geq \alpha, \quad \forall i$$

$$\sum_k x_k \leq \beta$$

$$x_k \in \{0,1\}$$

And $\lambda$ controls the density of the constraint matrix of the IP problem

- if $\lambda = 0.5$, the coefficients of the constraints have value 1 only when the value of $k$ for element $i$ is different from the mode of the values of $k$ over the element of the other class;

- If $f$ is not rounded, then the constraints represent the maximization of the average hamming distance between the $k^{th}$ coordinate of element $i$ and the same coordinate of all the elements belonging to the other class.
How to solve those large (and hard) IPs?

• At optimality: whit contained dimensions; else heuristics...

**RELEVANT ISSUES**

• The quality of solution depends on the chosen sample as well as on the solution algorithms
• There are many equivalent solutions for a given problem
• Cross validation approach: integrate the solutions obtained on different subset of the available data (re-sampling)

• It is required to solve many instances of the same problems over different input data ...

  Good heuristics seem to be the right approach...
  Their weakness w.r.t. optimal methods are balanced by data sampling

  Is it better to have MANY GOOD SOLUTIONS or FEW OPTIMAL ONES?
H1) GRASP HEURISTICS

FS is NP-hard

- **GRASP**: Greedy Randomized Adaptive Search Procedure, successfully applied to find approx solutions to hard combinatorial problems (Festa and Resende, '02, '08).

- Each GRASP iteration consists of two phases:
  1. an iterative greedy adaptive randomized construction phase that builds a feasible solution;
  2. local search phase.

1. Define candidate elements C;
2. Apply a greedy function $g(e)$, $e$ in C;
3. Rank candidates in C according to greedy function values $g(e)$;
4. Put well ranked candidates into a restricted candidate list RCL;
5. Randomly choose one element in RCL and add it to the solution under construction;
6. Adaptive component: greedy function values depend on the partial solution under construction.
GRASP for Feature Selection

The objective function is composed of three parts, with weights of decreasing importance:

- the value of $\alpha$
- the number of rows covered at value larger than $\alpha$,
- the total extra coverage spent on the rows.

A swap local search procedure is applied to improve the solution, i.e. a new set of columns with lower cardinality (removal of redundant columns) and/or corresponding to a higher coverage;

At each local search iterations, candidate sets of columns to be swapped are defined and all swaps are tested. Ad hoc data structures enable the construction and local search steps to be very fast.
RCL construction

\[ g_{\min} = \min_{e \in C} g(e) \]
\[ g_{\max} = \max_{e \in C} g(e) \]

- **Cardinality based**: RCL is made of the k elements with the best greedy value.

- **Value based**: RCL is associated with a parameter \( \delta \) in [0,1] and a threshold value \( \mu = g_{\min} + \delta(g_{\max} - g_{\min}) \)

- \( \delta = 0 \): Pure greedy
- \( \delta = 1 \): Pure random
# Results on Feature Selection

<table>
<thead>
<tr>
<th>Name</th>
<th>Number of Problems</th>
<th>maximum solution time (secs)</th>
<th>Best solution proved optimal</th>
<th>GRASP finds Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>t01</td>
<td>5</td>
<td>120</td>
<td>5/5</td>
<td>5/5</td>
</tr>
<tr>
<td>t02</td>
<td>5</td>
<td>120</td>
<td>5/5</td>
<td>5/5</td>
</tr>
<tr>
<td>t03</td>
<td>5</td>
<td>120</td>
<td>5/5</td>
<td>5/5</td>
</tr>
<tr>
<td>t04</td>
<td>5</td>
<td>900</td>
<td>3/5</td>
<td>4/5</td>
</tr>
<tr>
<td>t05</td>
<td>5</td>
<td>1.800</td>
<td>0/5</td>
<td>3/5</td>
</tr>
<tr>
<td>t06</td>
<td>5</td>
<td>3.600</td>
<td>0/5</td>
<td>5/5</td>
</tr>
</tbody>
</table>
Application:
Mining transcriptome data of the AD11 transgenic mouse model

Joint work with European Brain Research Institute Rita Levi-Montalcini, Roma, Italy (EBRI)

- The αD11 anti-NGF antibody is composed by the light (VK) and heavy (VH) chains. Crossing mice expressing the light chain (VK mice) with mice expressing the heavy chain (VH mice) yields double transgenic offspring, which expresses a functional αD11 antibody (anti-NGF AD11 mice).

- The AD11 anti-NGF mice represent a comprehensive transgenic model for an Alzheimer-like neurodegeneration, displaying in a progressive way a full complement of phenotypic hallmarks for the disease.

For a total of 120 samples
Aims of the Project

1. To characterize the gene expression profile of the AD11 mice in different brain areas following temporal progression

2. To identify a limited set of genes able to discriminate between the neurodegeneration and the healthy state

3. Explain the onset of the Alzheimer disease and thus identify early biomarkers of the pathology

a) Discretization

- The data is transformed from numerical to qualitative/binary
  1) create many intervals for the expression of the genes
  2) merge intervals based on explained entropy
- Each gene receives its most appropriate number and type of intervals
- genes that are not varying across the samples are discarded

b) Feature Clustering

- Features with the same discretized profile over the samples are clustered

c) Feature Selection

- The FS model is solved with GRASP with $\beta$ values

d) Learning

- Apply Lsquare as described on the reduced feature set
Some Results

- There are few genes (7) that are able, one by one, to separate exactly all the healthy from the sick mice (iterative application of the method) in leave-1-out cross validation.

- The 7 genes are highly co-regulated or contro-regulated and identify a regulatory network that is presently under study at EBRI.

- More genes are strongly co-regulated with the 7 genes network:
  
  CLASS 1: A_52_P58XXXX \geq 0.87  
  CLASS 2: A_52_P58XXXX < 0.87
Application:
Species Classification through Barcode

- A BARCODE is a small portion of mitochondrial DNA where the nucleotides change rapidly among species.
- Given samples from different species, the objective is to identify those combinations of muted nucleotides that have determined the differences from one to another in the evolution path.
- BARCODING is a relatively new problem that is drawing attention of the bio-comp community.

The international Consortium CBOL, funded by the Sloan Foundation, is investing money since 2005 in collecting barcodes of many species and in putting together a library of algorithms for its analysis. The CBOL website now makes available to researchers more than 2M barcodes.

IASI is member of the Data Analysis Working Group of CBOL since 2006 and has developed a species classifier based on Logic Programming (BLOG) made available on the Consortium website.
Experiments

a) Discretization

The A,C,G,T values of the sites are associated to presence/absence logic variables

b) Feature Selection

Using the compact FS model solved optimally few sites are identified (10-30)

c) Learning

Apply Lsquare as described on the reduced feature set and obtain a formula that tells a species from the others

1. For each species k, we solve a 2-class learning problem:
   ✓ class A: subset of samples in class k
   ✓ class B: samples of class different from k
   • We use a training subset (80%, 90%) of the available data, and then test their classification capabilities on the remaining data.
   • Training and testing samples are drawn at random maintaining the same proportion for each species
Experiments

1700 samples
150 different species
648 to 690 sites (or nucleotides).

826 samples
82 different species
660 sites (or nucleotides).

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$\alpha$</th>
<th>test%</th>
<th>Error Rates</th>
<th>Error Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>training</td>
<td>testing</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>10</td>
<td>8.02%</td>
<td>17.00%</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>10</td>
<td>10.14%</td>
<td>20.00%</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>20</td>
<td>11.90%</td>
<td>21.52%</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>20</td>
<td>13.50%</td>
<td>21.19%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>average</td>
<td>10.89%</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>10</td>
<td>0.87%</td>
<td>10.00%</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>10</td>
<td>1.93%</td>
<td>12.50%</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>20</td>
<td>1.50%</td>
<td>10.93%</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>20</td>
<td>2.04%</td>
<td>12.25%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>average</td>
<td>1.58%</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>15</td>
<td>0.20%</td>
<td>8.94%</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>15</td>
<td>0.61%</td>
<td>7.72%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>average</td>
<td>0.40%</td>
</tr>
</tbody>
</table>

Table 2: Optimal values and Error Rates (first data set)

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$\alpha$</th>
<th>test%</th>
<th>Error Rates</th>
<th>Error Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>training</td>
<td>testing</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>10</td>
<td>15.47%</td>
<td>15.06%</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>10</td>
<td>16.48%</td>
<td>15.90%</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>20</td>
<td>19.97%</td>
<td>21.03%</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>20</td>
<td>21.39%</td>
<td>22.56%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>average</td>
<td>18.32%</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>10</td>
<td>5.52%</td>
<td>6.67%</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>10</td>
<td>6.37%</td>
<td>8.33%</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>20</td>
<td>7.40%</td>
<td>10.42%</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>20</td>
<td>10.88%</td>
<td>10.42%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>average</td>
<td>7.54%</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>10</td>
<td>0%</td>
<td>1.38%</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>10</td>
<td>2.22%</td>
<td>3.08%</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>20</td>
<td>1.97%</td>
<td>5.50%</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>20</td>
<td>1.58%</td>
<td>5.13%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>average</td>
<td>1.44%</td>
</tr>
</tbody>
</table>

Table 4: Optimal values and Error Rates (second data set)

<table>
<thead>
<tr>
<th>SPECIES</th>
<th>CC</th>
<th>WC</th>
<th>CLAUSE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>1.00</td>
<td>0.00</td>
<td>(v100=c) and (v346=a) and (v499=t) and (v502=a)</td>
</tr>
<tr>
<td>A2</td>
<td>0.77</td>
<td>0.00</td>
<td>(v82=t) and (v238=t) and (v502=c)</td>
</tr>
<tr>
<td>A3</td>
<td>1.00</td>
<td>0.00</td>
<td>(v58=a) and not(v100=c) and not(v106=a)</td>
</tr>
<tr>
<td>A4</td>
<td>1.00</td>
<td>0.00</td>
<td>(v106=t) and (v139=g)</td>
</tr>
<tr>
<td>A5</td>
<td>1.00</td>
<td>0.00</td>
<td>not(v106=g) and not(v295=a) and not(v295=g)</td>
</tr>
</tbody>
</table>

Table 3: Logic Formulas for Species 1 to 5 (first data set)