IDIAP RESEARCH REPORT

IDIAP
Martigny - Valais - Suisse

# Combinatorial Approach for Data Binarization

Eddy Mayoraz [1]    Miguel Moreira [1]

IDIAP–RR 99-08

May 1999

revised in July 99

published in
Principles of Data Mining and Knowledge Discovery: third european conference; proceedings / PKDD'99

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[1] IDIAP—Dalle Molle Institute of Perceptual Artificial Intelligence, P.O. Box 592, CH-1920 Martigny, Switzerland

IDIAP Research Report 99-08

# Combinatorial Approach for Data Binarization

Eddy Mayoraz Miguel Moreira

May 1999

revised in July 99

published in
Principles of Data Mining and Knowledge Discovery: third european conference; proceedings / PKDD'99

**Abstract.** This paper addresses the problem of transforming arbitrary data into binary data. This is intended as preprocessing for a supervised classification task. As a binary mapping compresses the total information of the dataset, the goal here is to design such a mapping that maintains most of the information relevant to the classification problem. Most of the existing approaches to this problem are based on correlation or entropy measures between one individual binary variable and the partition into classes. On the contrary, the approach proposed here is based on a global study of the combinatorial property of a set of binary variable.

**Acknowledgements**: The support of the Swiss National Science Foundation under grant 2000-053902.98 is gratefully acknowledged.

# 1 Introduction

*Supervised classification learning* addresses the general problem of finding a plausible $K$-partition into *classes* of an *input space* $\Omega$, given a $K$-partition of a set of *training examples* $X = X_1 \uplus \ldots \uplus X_K \subset \Omega$.

In practical applications of data mining the input spaces $\Omega$ are usually very large and they combine features of different nature. Therefore, for most of the mining tools to be usable, it is convenient to preprocess the data and an important research effort is now spent on problems such as feature selection or feature discretization.

Some mining technologies require even purely binary data. This is the case of *Logical Analysis of Data* (LAD), which is a general approach for knowledge discovery and automated learning proposed in the mid eighties [Hammer, 1986]. Classification is one particular usage of this theory, which was extensively developed and implemented in the mid nineties and which showed great potentialities [Boros *et al.*, 1996].

Thus, besides data compression, there is a need in data binarization in view of mining, where the most relevant information for further processing has to be maintained (Sect. 2). In Sect. 3, the binarization problem is stated and some classical approaches are briefly presented. Section 4 presents the algorithm IDEAL, specially designed to fit the needs of LAD. Some experimental results are discussed in Sect. 5 and Sect. 6 concludes and discusses further work.

# 2 Requirements for Binarization

Given a set of training examples $X \subset \Omega$ partitioned into $K$-classes $X_1 \uplus \ldots \uplus X_K$, the binarization problem consists in finding a mapping $m : \Omega \rightarrow \{0,1\}^d$ with the following properties: $(i)$ most of the information relevant to the classification problem should be preserved through $m$; $(ii)$ the size $d$ of the binary codes is not too large. The first property is translated into a sharp and a soft constraint. The former states that the mapping should be *consistent* with the training examples, i.e. $m(X_i) \cap m(X_j) = \emptyset,\ \forall i \neq j$. The latter asks that two points of $\Omega$, close to each other according to a reasonable metric, should have their images through $m$ close to each other in the Hamming distance metric.

The second property has to be taken with some care. Clearly, the size of the binary codes should be small in order to reduce the complexity of the processing of binarized data. The research of a binary mapping of minimal size satisfying the consistency constraint is a challenging combinatorial problem proven to be *NP*-Hard in most of its forms [Boros *et al.*, 1997]. However, experience has shown that the final performances of any learning method applied to the binarized data can drop whenever $d$ is too small. This suggests that the consistency constraint is not sufficient to ensure that the relevant information is not lost in the binarization of the data.

In practice, it is useful that the method determining the binarization provides also a way to control the size of the binary codes produced. For this purpose, the consistency constraint can be extended in a natural way as follows. A mapping $m$ is $c$-consistent with the training examples if and only if for any two examples $\boldsymbol{x} \in X_i$ and $\boldsymbol{y} \in X_j,\ i \neq j$, the Hamming distance between $m(\boldsymbol{x})$ and $m(\boldsymbol{y})$ is at least $c$. Clearly, 1-consistency is identical to plain consistency. Experimentations with LAD showed that binary mappings $c$-consistent with the training examples, with $c = 2$ or 3 are still of reasonable size for most of the datasets and allow a strong improvement of the overall behavior of the method.

For the sake of generality, the binarization methods must be able to handle input spaces $\Omega$ composed of attributes of different kinds: binary, nominal (ordered and unordered), or continuous. For the purpose of interpretation simplicity, each one of the $d$ binary functions of the mapping $m : \Omega \rightarrow \{0,1\}^d$ involves only one original attribute of the input space $\Omega$.

In the sequel, each binary function $m_i : \Omega \rightarrow \{0,1\}$, composing the binary mapping $m$ is called a *discriminant* and is restricted to the following types. When associated to an unordered attribute (binary or nominal), a discriminant is identified to one possible value of this attribute (e.g. "*color* = *yellow*"). In the case of an ordered attribute (nominal or continuous), a discriminant is a comparison to a threshold value (e.g. "$age > 45$").

To be usable on real life datasets, a binary mapping must handle properly unknown data, noisy data as well as a priori knowledge such as monotonic relationship between attributes and the target. The algorithm proposed hereafter addresses these issues. Though, space constraints prevent us from going into further details.

# 3 Existing Binarization Methods

Some learning methods generate, as a by-product, a discriminant set $\mathcal{D}$ that defines a binary mapping $m$. For example, when a decision tree is built to learn a classification task, each internal node of the tree is a discriminant. Moreover, if no early stopping criterion is used and if the tree is not pruned, all examples associated to one particular leaf are of the same class. Thus the binary mapping of size $d$ given by the number of nodes is consistent.

In this paper, the focus is put on global binarization algorithms, which usually assume a large (implicit or explicit) initial discriminant set, from which a small subset has to be extracted. For comparisons between global binarization methods and local approaches such as decision trees, please refer to [Moreira *et al.*, 1999].

Given a training set $X$ of examples partitioned as before into $K$ classes, and given a large set $\mathcal{D}$ of discriminants defining a binary mapping consistent with $X$, the problem of finding a small subset of $\mathcal{D}$, still consistent with $X$, can be formalized as a minimum set covering problem. For each discriminant $\tau \in \mathcal{D}$ there is one variable $\boldsymbol{z}_\tau \in \{0, 1\}$ indicating whether $\tau$ belongs to the resulting subset or not. The constraint matrix $\boldsymbol{A}$ has a row for each pair of examples $\boldsymbol{x} \in X_i, \boldsymbol{y} \in X_j,\ i \neq j$. $A_{(x,y),\tau} = 1$ if the discriminant $\tau$ distinguishes the example $\boldsymbol{x}$ from $\boldsymbol{y}$ (i.e. $* \neq m_\tau(\boldsymbol{x}) \neq m_\tau(\boldsymbol{y}) \neq *$) and is 0 otherwise. A subset of discriminants defines a binary mapping $c$-consistent with $X$ if and only if its characteristic vector $\boldsymbol{z} \in \{0,1\}^{|\mathcal{D}|}$ satisfies $\boldsymbol{A}\boldsymbol{z} \geq c$.

The minimum set covering problem is an *NP*-Complete problem, but for our purpose, optimality is not critical and thus, any good heuristic is satisfactory. The most obvious heuristic for the resolution of the minimum set covering problem is the incremental greedy approach. It consists, at each iteration, in selecting the column of $\boldsymbol{A}$ with the highest number of 1s, introducing the corresponding discriminant $\tau$ in the solution (i.e. switching $\boldsymbol{z}_\tau$ from 0 to 1), and suppressing the rows $i$ in $\boldsymbol{A}$ whenever $\boldsymbol{A}_i\boldsymbol{z} \geq c$.

A more critical issue is related to the computational complexity of this approach. If $D$ denotes the initial number of discriminants and if there are $n$ examples in $X$, the construction of the constraint matrix $\boldsymbol{A}$ is in $O(n^2D)$. A naive implementation of this greedy heuristic has a complexity in $O(n^2Dd)$ and has demonstrated its limitations in the experiments reported in [Boros *et al.*, 1996].

A very nice solution proposed in [Almuallim and Dietterich, 1994] (denoted "Simple-Greedy" in Sect. 5) consists in resolving this minimum set covering problem using the same greedy heuristic, but without enumerating any column of $\boldsymbol{A}$. A clever data-structure is used that allows to determine the number of conflicts solved by a discriminant at a given time in $O(n)$. The total complexity of this approach is $O(nDd)$, where $d$ is the size of the final subset of discriminants. However, this approach is designed to solve the problem of the 1-consistent discriminant set and is not easily generalizable to the $c$-consistency case.

The algorithm proposed in the next section is an alternative to this problem as it addresses the $c$-consistency issue. Even though its worst case complexity is in $O(D \log D + n^2D)$, it is shown to be quite efficient in practice even with large training samples.

# 4 An Eliminative Approach

The algorithm Ideal (Iterative Discriminant Elimination Algorithm) is an eliminative procedure for finding a minimal $c$-consistent discriminant set. As for the other global methods discussed in Section 3, the initial discriminant set $\mathcal{D}$ is obtained by placing: along each ordered attribute, one discriminant between every two projected examples of different classes and of consecutive values ; and for each unordered attribute, one discriminant for each one of its possible values.

IDEAL iteratively selects discriminants from $\mathcal{D}$ minimizing a merit function $w(\tau)$. Each selected discriminant is eliminated if approved on a redundancy test (checking whether the elimination of this discriminant would still leave at least $c$ others discriminating every pair distinguished by this discriminant), and kept otherwise for the final solution. This process is repeated until all the discriminants have been tested once. Choosing as $w(\tau)$ the total number of pairs of examples from different classes discriminated by $\tau$ would lead to an algorithm very similar to the greedy heuristic described in Section 3. The only difference would be that in the former case the solution is built iteratively while here it is pruned iteratively.

Among the various merit functions $w(\tau)$ experimented [Moreira *et al.*, 1999], the one finally selected for IDEAL measures the number of *local conflicts* defined as follows. If the discriminant $\tau$ is based on the original attribute $a$, $w(\tau)$ is the number of pairs of examples from different classes, discriminated by $\tau$ and by no other discriminants based on $a$.

This choice for $w(\tau)$ has several advantages, the most important one is the computational complexity. The merit of each discriminant is computed (cheaply) once at the beginning and then, whenever a discriminant $\tau$ is pruned, the merit changes for only few discriminants (each one associated to the same original attribute $a$ as $\tau$, and in the case of an ordered attribute even only the two discriminants just below and just after $\tau$ along $a$). The second advantage of this merit function is that it makes IDEAL sensitive to the relation between discriminants and original attributes. Consequently, this introduces a bias in the final solution towards sets of discriminants well spread over the different attributes, i.e. avoid (if possible) having many discriminants related to the same attribute. We consider that in many applications, this is a desirable property.

# 5 Experiments

The response of IDEAL to the rise of the consistency constraint has been studied empirically. The algorithm was tested on 21 datasets from the UCI repository of machine learning databases [Blake *et al.*, 1998], with different values of $c$, from 1 to 4. Table 1 contains the results, including the test done with Simple-Greedy, for comparison purposes.

Table 1 shows that, against expectations, the obtained number of discriminants increases more than linearly with $c$ for 6 of the datasets. We find explanation for this in the fact that there is a set of important discriminants providing large amounts of separations and thus covering a significant part of the plain-consistency solution, but as the consistency constraint is tightened, discriminants of increasing specificity are added to the solution, resulting in a faster increase of the latter. Nevertheless, for the majority of the remaining datasets the increase is less than linear, corresponding to the expected behavior.

In terms of execution time, although it generally decreases with $c$, that is not a general behavior. Both increases and decreases can be explained. However, no element allows to predict the particular evolution for a given dataset, that being dependent on its intrinsic, non-observable characteristics. The redundancy test of IDEAL is composed mainly of two nested loops, the outermost dedicated to the pairs of examples to be tested and the innermost to the search in other dimensions for at least $c$ alternative discriminants separating those pairs. The raise of $c$ abbreviates the outermost loop, since less time will probably be needed to find a non-compliant pair, but it will prolong the innermost loop because more alternative dimensions must be analyzed until the minimal separability is found.

The observed decreasing tendency in execution time is an argument in favor of eliminative procedures, as their search path is shortened when the consistency constraint is strengthened, as opposed to constructive approaches.

Table 1: Evolution of IDEAL with the raise of the minimal consistency level. Simple-Greedy (SG), a 1-consistent, constructive procedure is provided for reference. The left-hand part of the table shows the size of the obtained discriminant sets. The $D$ column gives the initial size.

| | | FINAL SIZE $d$ | | | | | EXECUTION TIME | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | consist. level $c$ (IDEAL) | | | | | consist. level $c$ (IDEAL) | | | | |
| dataset | $D$ | 1 | 2 | 3 | 4 | SG | 1 | 2 | 3 | 4 | SG |
| `abalone` | 5779 | 192 | 319 | 513 | 861 | 171 | 21.1 | 16.4 | 11.4 | 8.2 | $195e^3$ |
| `allhyper` | 440 | 21 | 36 | 59 | 110 | 18 | 53.7 | 43.1 | 31.6 | 22.2 | 16.3 |
| `allhypo` | 548 | 20 | 57 | 110 | 207 | 19 | 60.2 | 43.7 | 26.9 | 15.5 | 14.6 |
| `anneal` | 134 | 31 | 60 | 88 | 103 | 33 | 3.0 | 1.3 | 0.8 | 0.6 | 6.2 |
| `audiology` | 92 | 22 | 47 | 70 | 76 | 22 | 0.8 | 0.9 | 0.6 | 0.5 | 10.4 |
| `car` | 15 | 14 | 15 | 15 | 15 | 14 | 2.2 | 0.3 | 0.2 | 0.2 | 0.3 |
| `dermatology` | 141 | 13 | 21 | 27 | 34 | 13 | 5.0 | 7.3 | 7.9 | 8.2 | 1.2 |
| `ecoli` | 301 | 24 | 47 | 93 | 203 | 20 | 0.3 | 0.3 | 0.2 | 0.1 | 5.9 |
| `glass` | 692 | 17 | 30 | 47 | 71 | 15 | 0.4 | 0.4 | 0.4 | 0.3 | 5.6 |
| `heart-dise.` | 309 | 14 | 21 | 34 | 51 | 12 | 0.8 | 0.6 | 0.7 | 0.5 | 1.4 |
| `krkopt` | 39 | 34 | 34 | 34 | 34 | 34 | 35.3 | 3.8 | 3.2 | 3.2 | 217.6 |
| `letter` | 234 | 59 | 90 | 128 | 151 | 58 | 3598.3 | 2531.5 | 1368.2 | 395.9 | 2586.9 |
| `mushroom` | 112 | 7 | 14 | 29 | 42 | 6 | 1469.2 | 1853.2 | 1453.5 | 718.3 | 4.6 |
| `nursery` | 19 | 17 | 19 | 19 | 19 | 17 | 110.7 | 1.8 | 1.8 | 1.8 | 2.5 |
| `page-blocks` | 3378 | 45 | 82 | 126 | 193 | 39 | 112.0 | 88.8 | 64.0 | 45.1 | 396.1 |
| `diabetes` | 856 | 24 | 40 | 66 | 127 | 22 | 2.2 | 1.8 | 1.5 | 1.0 | 17.0 |
| `segmentati.` | 9817 | 28 | 53 | 74 | 100 | 24 | 109.2 | 147.6 | 200.9 | 177.6 | 698.8 |
| `soybean` | 97 | 25 | 35 | 44 | 52 | 22 | 6.7 | 9.5 | 8.0 | 8.2 | 43.9 |
| `vehicle` | 1215 | 34 | 49 | 71 | 92 | 26 | 15.0 | 20.2 | 17.3 | 18.4 | 44.9 |
| `vowel` | 7077 | 26 | 38 | 59 | 86 | 22 | 9.7 | 10.1 | 10.3 | 9.6 | 773.6 |
| `yeast` | 374 | 39 | 82 | 173 | 271 | 41 | 4.0 | 2.6 | 1.0 | 0.3 | 139.4 |
| average ratio $d/D$ | | 20.4 | 27.1 | 34.8 | 42.9 | 20.1 | | | | | |
| (std) | | ±30.3 | ±32.4 | ±33.5 | ±34.2 | ±30.4 | | | | | |
| average evolution | | | 0.7 | 1.7 | 3.1 | | | -0.15 | -0.28 | -0.43 | |
| (std) | | | ±0.4 | ±1.1 | ±2.4 | | | ±0.4 | ±0.5 | ±0.5 | |

# 6 Conclusions and Further Research

We have described the basic concepts of Logical Analysis of Data and highlighted the need for finding a suitable binary mapping that can transform data of arbitrary form into binary data, unique format tractable by LAD.

IDEAL, an eliminative algorithm for finding a minimal discriminant set consistent with a set of training examples, has been described. The relation between the enlargement of the minimal differentiability among binarized objects with their resulting size growth was also examined. It has been shown that the growth rate depends on the data, although in the majority of the tested cases less than linear growth has been observed.

For comparison, an alternative, constructive approach has been briefly described and tested, with 1-consistent constraint. We speculate that constructive procedures are, in principle, less adapted to the referred constraint tightening, due to the consequent longer search path. No empirical evidence has been provided, though, due to the absence of a constructive approach of satisfying efficiency that is able to deal with the problem.

Concerning further work, we refer that an early stopping criterion could accelerate the proposed algorithm execution without major result deterioration. This aspect is discussed in [Moreira *et al.*, 1999], although a suitable solution is yet to be developed. In fact, the time complexity of the redundancy tests tends to $O(n^2)$ as the elimination of discriminants proceeds. In this latter phase, small decreases are verified in the discriminant set size.

A natural goal for follow-up activity consists in measuring the quality of the obtained binary mappings applied to LAD in classification tasks.

## References

[Almuallim and Dietterich, 1994] Hussein Almuallim and Thomas G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1–2):279–306, 1994.

[Blake *et al.*, 1998] C. Blake, E. Keogh, and C.J. Merz. UCI repository of machine learning databases, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[Boros *et al.*, 1996] E. Boros, P. L. Hammer, Toshihide Ibaraki, A. Kogan, E. Mayoraz, and I. Muchnik. An implementation of logical analysis of data. RRR 22-96, RUTCOR–Rutgers University's Center For Operations Research, `http://rutcor.rutgers.edu:80/~rrr/`, July 1996. To appear in IEEE Trans. on Knowledge and Data Engineering.

[Boros *et al.*, 1997] E. Boros, P. L. Hammer, Toshihide Ibaraki, and A. Kogan. Logical analysis of numerical data. Technical Report RRR 4-97, RUTCOR, 1997.

[Hammer, 1986] Peter L. Hammer. Partially defined Boolean functions and cause-effect relationships. Int. Conf. on Multi-Attribute Decision Making Via OR-Based Expert Systems, University of Passau, Germany, April 1986.

[Moreira *et al.*, 1999] Miguel Moreira, Alain Hertz, and Eddy Mayoraz. Data binarization by discriminant elimination. In Ivan Bruha and Marco Bohanec, editors, *Proceedings of the ICML-99 Workshop: From Machine Learning to Knowledge Discovery in Databases*, pages 51–60, 1999. `ftp://ftp.idiap.ch/pub/reports/1999/rr99-04.ps.gz`.