Studies in Classification, Data Analysis, and Knowledge Organization

Managing Editors
H.-H. Bock, Aachen
W. Gaul, Karlsruhe
M. Vichi, Rome

Editorial Board
Ph. Arabie, Newark
D. Baier, Cottbus
F. Critchley, Milton Keynes
R. Decker, Bielefeld
E. Diday, Paris
M. Greenacre, Barcelona
C. Lauro, Naples
J. Meulman, Leiden
P. Monari, Bologna
S. Nishisato, Toronto
N. Ohsumi, Tokyo
O. Opitz, Augsburg
G. Ritter, Passau
M. Schader, Mannheim
C. Weihs, Dortmund

Titles in the Series

H.-H. Bock and P. Ihm (Eds.)
Classification, Data Analysis,
and Knowledge Organization. 1991
(out of print)

M. Schader (Ed.)
Analyzing and Modeling Data
and Knowledge. 1992

O. Opitz, B. Lausen, and R. Klar (Eds.)
Information and Classification. 1993
(out of print)

H.-H. Bock, W. Lenski, and M. M. Richter
(Eds.)
Information Systems and Data Analysis.
1994 (out of print)

E. Diday, Y. Lechevallier, M. Schader,
P. Bertrand, and B. Burtschy (Eds.)
New Approaches in Classification and
Data Analysis. 1994 (out of print)

W. Gaul and D. Pfeifer (Eds.)
From Data to Knowledge. 1995

H.-H. Bock and W. Polasek (Eds.)
Data Analysis and Information Systems.
1996

E. Diday, Y. Lechevallier, and O. Opitz
(Eds.)
Ordinal and Symbolic Data Analysis. 1996

R. Klar and O. Opitz (Eds.)
Classification and Knowledge
Organization. 1997

C. Hayashi, N. Ohsumi, K. Yajima,
Y. Tanaka, H.-H. Bock, and Y. Baba (Eds.)
Data Science, Classification,
and Related Methods. 1998

I. Balderjahn, R. Mathar, and M. Schader
(Eds.)
Classification, Data Analysis,
and Data Highways. 1998

A. Rizzi, M. Vichi, and H.-H. Bock (Eds.)
Advances in Data Science
and Classification. 1998

M. Vichi and O. Opitz (Eds.)
Classification and Data Analysis. 1999

W. Gaul and H. Locarek-Junge (Eds.)
Classification in the Information Age. 1999

H.-H. Bock and E. Diday (Eds.)
Analysis of Symbolic Data. 2000

H. A. L. Kiers, J.-P. Rasson, P. J. F. Groenen,
and M. Schader (Eds.)
Data Analysis, Classification,
and Related Methods. 2000

W. Gaul, O. Opitz and M. Schader (Eds.)
Data Analysis. 2000

R. Decker and W. Gaul (Eds.)
Classification and Information
Processing at the Turn of the Millenium.
2000

S. Borra, R. Rocci, M. Vichi,
and M. Schader (Eds.)
Advances in Classification
and Data Analysis. 2001

W. Gaul and G. Ritter (Eds.)
Classification, Automation,
and New Media. 2002

K. Jajuga, A. Sokołowski, and H.-H. Bock
(Eds.)
Classification, Clustering and Data
Analysis. 2002

M. Schwaiger, O. Opitz (Eds.)
Exploratory Data Analysis
in Empirical Research. 2003

M. Schader, W. Gaul, and M. Vichi (Eds.)
Between Data Science and
Applied Data Analysis. 2003

H.-H. Bock, M. Chiodi, and A. Mineo
(Eds.)
Advances in Multivariate Data Analysis.
2004

D. Banks, L. House, F. R. McMorris,
P. Arabie, and W. Gaul (Eds.)
Classification, Clustering, and Data
Mining Applications. 2004

Daniel Baier
Klaus-Dieter Wernecke
Editors

Innovations in Classification, Data Science, and Information Systems

Proceedings of the 27th Annual Conference
of the Gesellschaft für Klassifikation e.V., Brandenburg
University of Technology, Cottbus, March 12–14, 2003

With 143 Figures and 111 Tables

Springer

Prof. Dr. Daniel Baier
Chair of Marketing and Innovation Management
Institute of Business Administration and Economics
Brandenburg University of Technology Cottbus
Konrad-Wachsmann-Allee 1
03046 Cottbus
Germany
daniel.baier@tu-cottbus.de

Prof. Dr. Klaus-Dieter Wernecke
Department of Medical Biometrics
Charité Virchow-Klinikum
Humboldt University Berlin
13344 Berlin
Germany
klaus-dieter.wernecke@charite.de

ISBN 3-540-23221-4 Springer-Verlag Berlin Heidelberg New York

Library of Congress Control Number: 2004114682

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer · Part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin · Heidelberg 2005
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Softcover-Design: Erich Kirchner, Heidelberg

SPIN 11326427 43/3130/DK - 5 4 3 2 1 0 - Printed on acid-free paper

Preface

This volume contains revised versions of selected papers presented during the 27th Annual Conference of the Gesellschaft für Klassifikation (GfKl), the German Classification Society. The conference was held at the Brandenburg University of Technology (BTU) Cottbus, Germany, in March 2003. Klaus-Dieter Wernecke chaired the program committee, Daniel Baier was the local organizer. Krzysztof Jajuga and Andrzej Sokolowski and their colleagues in Sekcja Klasyfikacji i Analizy Danych (SKAD), the Polish Classification Society, provided strong support during all phases of the conference.

The program committee was able to select 124 talks for 36 sessions. Additionally, it was possible to recruit 19 notable and internationally renowned invited speakers for plenary and semi-plenary talks on their current research work regarding the conference topic "Innovations in Classification, Data Science, and Information Systems" or, respectively, on the GfKl members' general fields of interest "Classification, Data Analysis, and Knowledge Organization". Thus, the conference, which was traditionally designed as an interdisciplinary event, again provided a large number of scientists and experts from Germany and abroad with an attractive forum for discussions and the mutual exchange of knowledge.

Besides on traditional subjects, the talks in the different sections focused on topics such as Methods of Data Analysis for Business Administration and Economics as well as Medicine and Health Services. This suggested the presentation of the papers of the volume in the following eight chapters:

- Discrimination and Clustering,
- Probability Models and Statistical Methods,
- Pattern Recognition and Computational Learning,
- Time Series Analysis,
- Marketing, Retailing, and Marketing Research,
- Finance, Capital Markets, and Risk Management,
- Production, Logistics, and Controlling,
- Medicine and Health Services.

The conference owed much to its sponsors (in alphabetical order)

- BTU Cottbus,
- Chair of Marketing and Innovation Management, BTU Cottbus,
- Holiday Inn Hotel, Cottbus,
- MTU Maintenance Berlin-Brandenburg GmbH, Ludwigsfelde,
- Scicon Scientific Consulting GmbH, Karlsruhe,
- Sparkasse Spree-Neiße, Cottbus,

- Synergy Microwave Europe GmbH & Co. KG, München,
- Volkswagen AG, Wolfsburg, and
- various producers of Scottish single malt whisky

who helped in many ways. Their generous support is gratefully acknowledged.

Additionally, we wish to express our gratitude towards the authors of the papers in the present volume, not only for their contributions, but also for their diligence and timely production of the final versions of their papers. Furthermore, we thank the reviewers for their careful reviews of the originally submitted papers, and in this way, for their support in selecting the best papers for this publication.

We would like to emphasize the outstanding work of Dr. Alexandra Rese who made an excellent job in organizing the refereeing process and preparing this volume. We also wish to thank Michael Brusch and his GfKl-2003 team for perfectly organizing the conference and helping to prepare the final program. In this context, special thanks are given to Jörg Swienty, Nadja Schütz, Matthias Kaiser, Christoph Schauenburg, and other members of the Chair of Marketing and Innovation Management, BTU Cottbus.

Finally, we want to thank Dr. Martina Bihn of Springer-Verlag, Heidelberg, for her support and dedication to the production of this volume.

Cottbus and Berlin, September 2004 *Daniel Baier*
Klaus-Dieter Wernecke

Contents

Part VII. Production, Logistics, and Controlling

Part VIII. Medicine and Health Services

Part I

Discrimination and Clustering

A New Agglomerative 2-3 Hierarchical Clustering Algorithm

Sergiu Chelcea[1], Patrice Bertrand[2,3], and Brigitte Trousse[1]

[1] INRIA, AxIS Research Group, BP 93, 06902 Sophia-Antipolis Cedex, France
[2] GET-ENST Bretagne, IASC, Technopôle Brest-Iroise
CS 83818, 29238 BREST Cedex, France
[3] INRIA, AxIS Research Group, BP 105, 78 153 Le Chesnay Cedex, France

Abstract. We studied a new general clustering procedure, that we call here Agglomerative 2-3 Hierarchical Clustering (2-3 AHC), which was proposed in Bertrand (2002a, 2002b). The three main contributions of this paper are: first, the theoretical study has led to reduce the complexity of the algorithm from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2 \log n)$. Secondly, we proposed a new 2-3 AHC algorithm that simplifies the one proposed in 2002 (its principle is closer to the principle of the classical AHC). Finally, we proposed a first implementation of a 2-3 AHC algorithm.

1 Motivations

Our motivation concerns the use of clustering techniques for user profiling and case indexing inside a Case-Based Reasoning framework (Jaczynski (1998)). It is in this context that we studied a new clustering strategy, called Agglomerative 2-3 Hierarchical Clustering (2-3 AHC). This strategy was recently proposed in Bertrand (2002a, 2002b) to generalize and to make more flexible the Agglomerative Hierarchical Clustering method (AHC).

Section 2 briefly presents the concept of 2-3 hierarchy together with the 2-3 AHC algorithm introduced in Bertrand (2002a). Section 3 derives a new 2-3 AHC algorithm while proposing to integrate the refinement step into the merging step. Before concluding in Section 5, Section 4 presents the complexity analysis, the implementation and some tests.

2 The 2-3 hierarchies and the 2-3 AHC algorithm

The following definitions and results of this section were established in Bertrand (2002a), in order to extend the framework of hierarchies[1].

In this text, we denote as E an arbitrary set of n objects to be clustered, and we suppose that E is described by a *dissimilarity*, say δ, i.e. $\delta(x, y)$ indicates the degree of dissimilarity between two arbitrary objects x and y.

[1] For the usual definitions in classification the reader can refer to (Gordon 1999).

2.1 2-3 Hierarchies

We consider a collection $\mathcal{C}$ of nonempty subsets of E, often called *clusters* in the rest of the text. If $X, Y \in \mathcal{C}$ satisfy $X \cap Y \neq \emptyset$, $X \not\subseteq Y$ and $Y \not\subseteq X$, then it will be said that X *properly intersects* Y. A *successor* of $X \in \mathcal{C}$ is any largest cluster, say X', that is strictly contained in X. If X' is a successor of X, then X is said to be a *predecessor* of X' (see Figure 1). The collection $\mathcal{C}$ is said to be *weakly indexed* by a map $f : \mathcal{C} \to \mathbf{R}^+$ if $X \subset Y$ implies $f(X) \leq f(Y)$ and if $f(X) = f(Y)$ with $X \subset Y$, implies that X is equal to the intersection of its predecessors. We recall also that $\mathcal{C}$ is said to be *indexed* by f if $X \subset Y$ implies $f(X) < f(Y)$. A 2-3 *hierarchy* on E is a collection $\mathcal{C}$ which contains E and its singletons, which is closed under nonempty intersections, and such that each element of $\mathcal{C}$ properly intersects no more than one other element of $\mathcal{C}$. A small example of a 2-3 hierarchy is presented in Figure 1a.

A 2-3 hierarchy on E is a family of intervals of at least a linear order defined on E. This property allows to represent graphically a 2-3 hierarchy as a pyramidal classification (cf. Figure 1). According to Theorem 3.3 in Bertrand (2002a), any 2-3 hierarchy on E has a maximum size of $\lfloor \frac{3}{2}(n-1) \rfloor$, excluding the singletons. In the following, we will say that two clusters X and Y are *noncomparable* if $X \not\subseteq Y$ and $Y \not\subseteq X$, and that a cluster X is *maximal* if $\nexists\, Z \in \mathcal{C}$ such that $X \subset Z$.

2.2 From AHC to 2-3 AHC

We first recall that the principle of AHC is to merge repeatedly two clusters until the cluster E is formed, the initial clusters being all the singletons. Each cluster is merged only once, and two clusters can be merged if they are closest - in the sense of a chosen *aggregation link*, denoted μ, and called simply *link*. Usual links are *single link*, *complete link*, *average link* and *Ward link*. When two clusters X and Y are merged, the link $\mu(X, Y)$ between these two clusters can be interpreted as a measurement, denoted $f(X \cup Y)$, of the degree of heterogeneity of $X \cup Y$. In addition, if we set $f(X) = 0$ for $|X| = 1$, the so defined map f on the set of clusters is not necessarily a weak index in the sense of Section 2.1, so that a refinement step (removing of certain clusters) is performed, in order that f becomes a weak index.

The 2-3 AHC algorithm below (Bertrand (2002a)) extends the AHC.

Algorithm of the 2-3 AHC (Bertrand (2002a)):

1. **Initialization**: $i = 0$; The set of clusters and the set of candidate[2] clusters $\mathcal{M}_i$ coincide with the set of singletons of E.
2. **Merge**: $i = i + 1$; Merge a pair $\{X_i, Y_i\}$ such that $\mu(X_i, Y_i) \leq \mu(X, Y)$, among the pairs $\{X, Y\} \subseteq \mathcal{M}_{i-1}$, which are noncomparable and satisfy α or β :
   (α) X and Y are maximal, and X (resp. Y) is the only cluster that may properly intersect Y (resp. X).

(β) One of X or Y is maximal, and the other admits a single predecessor Z. No cluster is properly intersected by X, Y or Z.

3. **Update**: $\mathcal{M}_i \leftarrow \mathcal{M}_{i-1} \cup \{X_i \cup Y_i\}$, from which we eliminate any cluster strictly included in at least a cluster of $\mathcal{M}_{i-1}$ and in $X_i \cup Y_i$. Update μ by using an extension of Lance and Williams Formula. Update f by using $f(X_i \cup Y_i) = \max\{f(X_i), f(Y_i), \mu(X_i, Y_i)\}$.
4. **Ending test**: repeat steps 2 et 3, until the cluster E is created.
5. **Refinement**: remove some clusters so that f is a weak index.

It has been proved in Bertrand (2002a) that for any choice of μ, this algorithm converges in at most $O(n^3)$, that after each step of the algorithm, the set of created clusters (completed by E) is a 2-3 hierarchy (cf. Bertrand (2002a), Proposition 5.4), and that the final structure is weakly indexed.

3 Proposition of a new 2-3 AHC algorithm

We present here a new 2-3 AHC algorithm derived from the previous one and based on the ideas presented in the following two subsections. Besides a simpler formulation (cf. Fact 34), the interest of this new algorithm (cf. Section 3.2) is two-fold: first, its principle is more similar to the principle of the AHC algorithm (cf. Fact 33) and second, we will see that the integration of the refinement phase into the merging phase (cf. Fact 35), allows to reduce the complexity of the algorithm (cf. Section 4).

3.1 Modifying the update and the merging steps

We begin with a reformulation of the update of candidates set $\mathcal{M}_i$ (Step 3).

Proposition 31 *In the 2-3 AHC algorithm, we can, without changing the results of the merging, choose $\mathcal{M}_i$ (step 3) in the following way: $\mathcal{M}_i$ equals $\mathcal{M}_{i-1} \cup \{X_i \cup Y_i\}$, from which we eliminate every successor of X_i or Y_i, and also the two clusters X_i and Y_i, if $X_i \cap Y_i \neq \emptyset$ or the merging of X_i and Y_i is of type β.*

Proof: In the initial algorithm, like in the new formulation, $\mathcal{M}_i$ is equal to $\mathcal{M}_{i-1} \cup \{X_i \cup Y_i\}$, deprived of certain clusters included in $X_i \cup Y_i$. It is thus enough to compare the two ways of defining $\mathcal{M}_i$ only for the clusters of $\mathcal{M}_{i-1}$ which are included in $X_i \cup Y_i$. We first examine the successors of X_i or of Y_i. In the initial algorithm, they don't belong to $\mathcal{M}_i$, because they are included in X_i or Y_i, and in $X_i \cup Y_i$. It is also clearly the case in the new formulation. In addition, in both ways of choosing $\mathcal{M}_i$, if a cluster W is included in one of the successors of X_i (resp. Y_i), then W does not belong to $\mathcal{M}_{i-1}$, because W

[2] X is *candidate* if $\exists Y \in \mathcal{C}$ such that X and Y are noncomparable, and their merging satisfy the 2-3 hierarchy definition (conditions α and β below).

was already eliminated from $\mathcal{M}_{i'}$ with $i' \leq i-1$ (we use the same arguments as for the elimination of the successors of X_i or Y_i, but to a stage previous to the formation of $X_i \cup Y_i$). Since X_i and Y_i are the only successors of $X_i \cup Y_i$, these are thus the only clusters left to examine, in order to determine if the choice of $\mathcal{M}_i$ varies according to the two formulations for choosing $\mathcal{M}_i$.

There are only three possible cases according to whether the merging of X_i and Y_i, is (a) of the type α with $X_i \cap Y_i = \emptyset$, (b) of the type α with $X_i \cap Y_i \neq \emptyset$, and (c) of the type β.

Case (a): α merging of X_i and Y_i, with $X_i \cap Y_i = \emptyset$. In this case, $X_i \cup Y_i$ is the only cluster containing X_i (resp. Y_i), because X_i (resp. Y_i) was maximal before the creation of $X_i \cup Y_i$. Thus neither X_i nor Y_i are removed from $\mathcal{M}_i$ in the initial algorithm, and also in the new formulation. It results that the two formulations are equivalent here.

Case (b): α merging of X_i and Y_i, with $X_i \cap Y_i \neq \emptyset$. Using the same argument as in case (a), we deduce that neither X_i nor Y_i are removed from $\mathcal{M}_i$ in the initial algorithm. On the other hand, X_i and Y_i do not belong to $\mathcal{M}_i$, if the new formulation is used. However according to the initial algorithm, neither X_i nor Y_i will be aggregate during a later merging of this algorithm. Indeed on the one hand, none of the clusters X_i and Y_i can be used for a β type merging, because X_i and Y_i properly intersect each other. On the other hand, none of the clusters X_i and Y_i can be used for an α merging, because X_i and Y_i are not maximal any more. Thus, the pairs of clusters that can be merged are the same in the two approaches.

Case (c): β merging of X_i and Y_i. Let us suppose - without any loss of generality - that Z is the (only) predecessor of X_i. Thus $X_i \notin \mathcal{M}_i$ in the initial algorithm, but $Y_i \in \mathcal{M}_i$ because Y_i is included in only one cluster $(X_i \cup Y_i)$. On the other hand, X_i and Y_i do not belong to $\mathcal{M}_i$, if the new formulation is used. However according to the initial algorithm, Y_i will not be aggregate during a later merging of the algorithm. Indeed, Y_i has a single predecessor $X_i \cup Y_i$ but $X_i \cup Y_i$ properly intersects Z (because Z strictly contains X_i but is disjoint of Y_i). Thus Y_i cannot be used for a β type merging, nor for an α type one. Thus, again the pairs of clusters that can be merged are the same in the two approaches, which finally proves that the new way of choosing $\mathcal{M}_i$ does not change the possibilities of merging at each iteration. □

The following property highlights the need of adding a merging step, that we call *intermediate merging* step, at the end of each β merging.

Proposition 32 *If the merging of the i^{th} step of the algorithm is of type β, then the cluster $X_i \cup Y_i$ formed at this stage, will necessarily be merged with the predecessor of X_i or Y_i, in a later step of the algorithm.*

Proof: Let us suppose - without any loss of generality - that Z is the (only) predecessor of X_i, before the β merging of X_i and Y_i. Let us place at the end of the β merging. Clearly $X_i \cup Y_i$ is maximal and $X_i \cup Y_i \in \mathcal{M}_i$.
Suppose that Z is not maximal, then $X_i \subset Z \subset Z'$, which implies that X_i has been eliminated from $\mathcal{M}_{i'}$ $(i' < i)$ no later than during the update following

the creation of Z': this contradicts $X_i \in \mathcal{M}_{i-1}$. Thus Z is maximal, and so $Z \in \mathcal{M}_i$, because a maximal cluster cannot be eliminated from any $\mathcal{M}_j$ ($j \leq i$). It results that the clusters $X_i \cup Y_i$ and Z belonging to $\mathcal{M}_i$, are maximal and properly intersect themselves. Thus they can be merged together in an α merging. Moreover, cluster $X_i \cup Y_i$ (resp. cluster Z) can be merged together only with cluster Z (resp. cluster $X_i \cup Y_i$) according to algorithm conditions. Assume that these two clusters are not merged together. Then we would merge together two other clusters A and B. These clusters A and B cannot be neither successors of X_i or of Y_i, nor X_i or Y_i themselves by Proposition 31. Moreover, A and B cannot be Z or its successors, since Z already properly intersects $X_i \cup Y_i$. Thus A and B would be included in $E - (X_i \cup Y_i \cup Z)$. Otherwise, the algorithm ends only when cluster E is created and we known that it ends (cf. Bertrand 2002a). However E cannot be created as long as only clusters included in $E - (X_i \cup Y_i \cup Z)$ are merged, so as long as the merging of $X_i \cup Y_i$ and Z is not performed, which completes the proof. □

3.2 New 2-3 AHC algorithm integrating the refinement step

We begin with three facts before presenting our new 2-3 AHC algorithm:

Fact 33 If at the end of any β merging of X_i and Y_i (i unspecified), we decide, following the Proposition 32, to merge $X_i \cup Y_i$ with the predecessor Z (of X_i or Y_i), then at the end of the so modified step 2, no cluster properly intersects a maximal cluster. In other words, *at the end of each modified step 2, the maximal clusters form a partition of* E, which underlines a strong analogy with the AHC algorithm characterized by this property.

Fact 34 For each i, the set $\mathcal{M}_i$ represents all the maximal clusters plus their successors when these successors are disjoint. This is a direct consequence of Proposition 31 and to the fact that each merging creates a maximal cluster. It results (taking into account the significant remark according to which the maximal clusters are disjoint) that one reformulates the (α) and (β) conditions in the following way, where $X, Y \in \mathcal{M}_{i-1}$: ($\alpha$) "$X$ and Y are maximal", (β) "only one of the clusters X and Y is maximal".

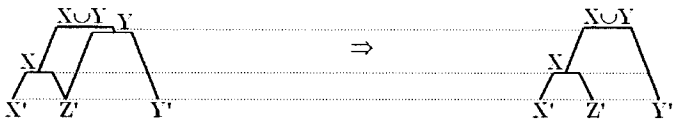

Fig. 1a. Refinement example **Fig. 1b.**

Fact 35 The refinement step can be integrated into the merging step, in order to obtain a weak indexing f. For this, each time we create a cluster $X \cup Y$, we compare $f(X \cup Y)$ with $f(X)$ and $f(Y)$. If $f(X \cup Y) = f(X)$ (resp. $f(X \cup Y) = f(Y)$), we remove X (resp. Y), provided that $X \cup Y$ is the only predecessor of X (resp. Y). This last case is illustrated in the example

from Figure 1 where $f(X) < f(Y) = f(X \cup Y)$: Y must then be eliminated from the structure.

New 2-3 AHC algorithm (see also Chelcea et al. (2002)):

1. **Initialization**: The candidate clusters set, $\mathcal{M}_0$, is the set of singletons of E. Let $i = 0$.
2. a) **Merge**: Let $i = i + 1$; Merge two clusters X_i and Y_i which are closest (in the sense of μ) among the pairs from $\mathcal{M}_{i-1}$, which are noncomparable and such that at least one of them is maximal;
   b) **Intermediate Merge**: If Z is a predecessor of the cluster X_i or Y_i such that $Z \neq X_i \cup Y_i$, then merge Z and $X_i \cup Y_i$, and eliminate from $\mathcal{M}_i$ these two clusters and their successors.
3. **Refinement**: Eliminate any cluster $W \in \{X_i, Y_i, X_i \cup Y_i, Z\}$ such that W has one predecessor, W', and such that $f(W) = f(W')$.
4. **Update**: Update $\mathcal{M}_i$ by adding the last formed cluster and eliminating the successors of the merged clusters and also the merged clusters if they properly intersect each other.
   Update μ and f.
5. **Ending test**: Repeat steps 2-4 until E is a cluster.

Concerning this new algorithm, we may notice that facts 33 and 34 imply that the clusters generated by the new merging step 2, form a 2-3 hierarchy. The integration of the refinement step inside the loop defined by steps 2-5, ensures that the clustering structure is weakly indexed by f, whereas it is clear that the deletion of some clusters having only one predecessor, does not change the property for the generated clusters to form a 2-3 hierarchy.

4 Complexity analysis and tests

4.1 Specifications

With the aim to specify and implement the new 2-3 AHC algorithm, we need to choose a link μ. In order to compare two non disjoint clusters, the definition of μ must extend the classical definitions of link used for disjoint clusters. Here we will use $\mu(X, Y) = \min\{\delta(x, y) : x \in X - Y, y \in Y - X\}$, together with an extension of the Lance and Williams formula.

In order to store and manage the matrix containing the link values between clusters, which is the most time expensive operation, we propose to use an *ordered tree structure* that puts in correspondence these values and the pairs of candidate clusters. The purpose is to search among all candidate cluster pairs for merging, the one that minimise a/several criteria/criterions.

We use three criterions in order to choose the merging pair: (1) *Minimal link*, since we search two closest clusters, (2) *Minimal cardinality*, meaning the number of elements of the clusters to be merged, when we have multiple pairs at a minimal link and (3) *Minimal lexicographical order* on the clusters

identifiers, when the two first criteria are satisfied by several pairs. Therefore, we have on the first level of the structure the ordered link values, on the second the ordered cardinalities of the pairs situated at the same link between clusters and on the third the lexicographically ordered identifiers.

4.2 Complexity analysis

The complexity of the **Initialization** (step 1) is larger than in Bertrand (2002a): $\mathcal{O}(n^2 \log n)$. The other steps are repeated n times and in the worst case the operations complexity will be reduced to $\mathcal{O}(n \log n)$ instead of $\mathcal{O}(n^2)$.

As follows we will analyze the complexity of the steps 2-4, which are repeated until the cluster E is created, that's at most $\lfloor \frac{3}{2}(n-1) \rfloor$ times. In the **Merging** step (Step 2.a), we first retrieve the pair that minimise our criteria, in $\mathcal{O}(1)$, and we create the new cluster $X_i \cup Y_i$ also in $\mathcal{O}(1)$. If one of the merged clusters has another predecessor, we perform an **Intermediate merge** (Step 2.b) with the same complexity as the one before. Thus the whole complexity of the step 2 is $\mathcal{O}(n)$.

In the **Refinement** step (Step 3), we will eliminate from the structure the clusters found on the same level with their predecessors and we will update the *predecessor*, *successor* links between the remaining clusters, which is done in $\mathcal{O}(n)$, since a cluster can have at most $\lfloor \frac{3}{2}(n-1) \rfloor$ successors.

In the **Update** step (Step 4) we first update $\mathcal{M}_i$ in $\mathcal{O}(n)$ since adding the new formed cluster is constant and since a cluster can have at most n successors to eliminate from $\mathcal{M}_i$. In the μ update we eliminate from the structure the pairs containing at least a cluster to be eliminated. Since a pair is eliminated in $\mathcal{O}(\log n)$ and we have at most $\lfloor \frac{3}{2}(n-1) \rfloor$ clusters, we have here an $\mathcal{O}(n \log n)$ complexity. Then, the links between the new formed cluster and the rest of the candidates are computed, each in $\mathcal{O}(n)$, and inserted into the matrix, in $\mathcal{O}(\log n)$ each. Therefore, the complexity of step 4 is $\mathcal{O}(n \log n)$.

Thus, the total worst case complexity is then reduced to $\mathcal{O}(n^2 \log n) + n \times \mathcal{O}(n \log n) = \mathcal{O}(n^2 \log n)$.

4.3 Implementation and tests

We designed an object-oriented model of the algorithm, which was implemented in Java, and integrated into the CBR*Tools framework (Jaczynski (1998)). We begun to test this algorithm as an indexing method in a CBR application for car insurance, based on a database[3] usually used in CBR. Then we carried out a series of tests on random generated data. Figure 2 indicates the execution times of our algorithm compared to the AHC algorithm depending on the size n of the uniformly generated data set E. Figure 3 shows the convergence of the ratio $\frac{Execution\ time(n)}{\mathcal{O}(n^2 \log n)}$ for the AHC and our 2-3 AHC algorithm, which confirms the theoretical complexity analysis.

[3] ftp://ftp.ics.uci.edu/pub/ml-repos/machine-learning-databases/autos

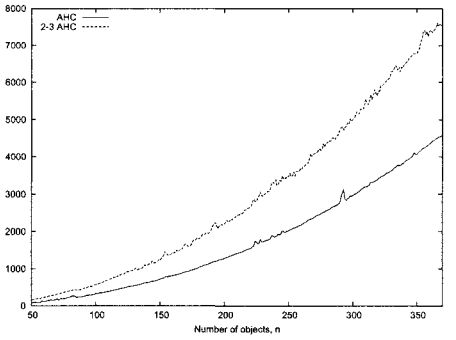

Fig. 2. Execution times

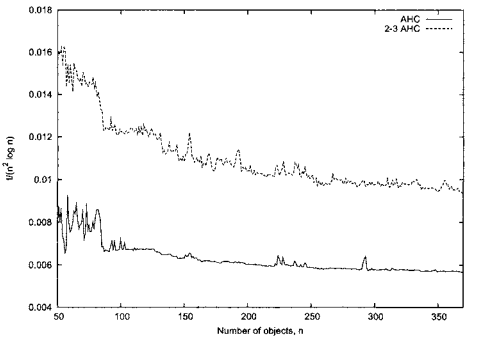

Fig. 3. Complexity validation

5 Conclusions and future work

The originality of this work is based on the four following points: (1) a new 2-3 AHC clustering algorithm, which simplifies the one proposed in 2002 (its principle is closer to the principle of the classical AHC), (2) a complexity reduction of the 2-3 AHC algorithm from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2 \log n)$, where n represents the number of objects to cluster, (3) a first object-oriented design and implementation of such an algorithm (in Java) and its integration in CBR*Tools, a Case-Based Reasoning framework and (4) an experimental validation of the algorithm complexity on simulated data.

Our current and future work concerns the following topics: (1) study of the quality of the 2-3 AHC compared with AHC and other classification methods and (2) study of the relevance of this new algorithm in the context of Web Usage Mining.

References

BERTRAND, P. (2002a): *Set systems for which each set properly intersects at most one other set - Application to pyramidal clustering.* Cahier du Ceremade numéro 0202, Ceremade, Université Paris-9, France.

BERTRAND, P. (2002b): *Les 2-3 hiérarchies : une structure de classification pyramidale parcimonieuse.* Actes du IX ème Congrès de la Société Francophone de Classification. 16-18 September, Toulouse, France.

CHELCEA, S., BERTRAND, P., and TROUSSE, B. (2002): *Theoretical study of a new 2-3 hierarchical clustering algorithm.* Symbolic and Numeric Algorithms for Scientific Computing, 9-12 Octobre, Timisoara, Romania.

GORDON, A.D. (1999): *Classification.* 2nd ed., Chapman and Hall, London.

JACZYNSKI, M. (1998): *Scheme and Object-Oriented Framework for case Indexing By Behavioural Situations : Application in Assisted Web Browsing.* Doctorat Thesis of the University of Sophia-Antipolis (in french), December, France.

Symbolic Classifier with Convex Hull Based Dissimilarity Function

Francisco de A.T. de Carvalho and Simith T. D'Oliveira Júnior

Centro de Informática - UFPE,
Av. Prof. Luiz Freire, s/n - Cidade Universitária,
CEP - 50740-540 - Recife - PE - Brasil
email: {stdj,fatc}@cin.fupe.br

Abstract. This work presents a new symbolic classifier based on a region oriented approach. At the end of the learning step, each class is described by a region (or a set of regions) in $\Re^p$ defined by the convex hull of the objects belonging to this class. In the allocation step, the assignment of a new object to a class is based on a dissimilarity matching function that compares the class description (a region or a set of regions) with a point in $\Re^p$. This approach aims to reduce the over-generalization that is produced when each class is described by a region (or a set of regions) defined by the hyper-cube formed by the objects belonging to this class. It then seeks to improve the classifier performance. In order to show its usefulness, this approach was applied to a study of simulated SAR images.

1 Introduction

New approaches have been recently proposed to discover knowledge and summarize the information stored in large data sets. Symbolic Data Analysis (SDA) is a new domain related to multivariate analysis, pattern recognition, databases and artificial intelligence. It is concerned with the generalization of classical exploratory data analysis and statistical methods (visualization, factorial analysis, regression, clustering methods, classification, etc.) into symbolic data (Bock and Diday (2000)). Symbolic data are more complex than the standard data because they contain internal variations and are structured.

In Ichino et al. (1996), a symbolic classifier was introduced as a region-oriented approach. The learning step uses an approximation of the Mutual Neighborhood Graph (MNG) and a symbolic operator (join) to furnish the symbolic description of each class. In the classification step, the allocation of an individual to a class is based on a matching function that compares the description of the individual with the symbolic description of the class. In Souza et al. (1999) and De Carvalho et al. (2000), another MNG approximation was proposed to reduce the learning step complexity without losing the classifier performance in terms of prediction accuracy. In the allocation step, alternative similarity and dissimilarity functions have been used to assign an individual to a class.

This work presents a new symbolic classifier based on a region-oriented approach. At the end of the learning step, each class is described by a region

(or a set of regions) in $\Re^p$ defined by the convex hull formed by the objects belonging to this class. This is obtained through a suitable approximation of a Mutual Neighborhood Graph (MNG). In the allocation step, the assignment of a new object to a class is based on a dissimilarity matching function that compares the class description (a region or a set of regions) with a point in $\Re^p$. This approach aims to reduce the over-generalization that is produced when each class is described by a region (or a set of regions) in $\Re^p$ defined by the hyper-cube formed by the objects belonging to this class. It then seeks to improve the classifier performance. In order to show its usefulness, this approach was applied to a study of simulated SAR images.

2 Symbolic data

In this paper, we are concerned with symbolic data that are represented by quantitative feature vectors. More general symbolic data type can be found in Bock and Diday (2000). Let $\Omega = \{\omega_1, \cdots, \omega_n\}$ be a set of n individuals described by p quantitative features $X_j (j = 1, \ldots, p)$. Each individual ω_i $(i = 1, \ldots, n)$ is represented by a quantitative feature vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, where x_{ij} is a *quantitative feature value.* A quantitative feature value may be either a continuous value (e.g., $x_{ij} = 1.80$ meters in height) or an interval value (e.g., $x_{ij} = [0.2]$ hours, the duration of a student evaluation).

Example. A segment (set of pixels) described by the grey level average and standard deviation calculated from its set of pixels may be represented by the continuous feature vector $\mathbf{x} = (50, 7.5)$. The description of a group of segments may be represented by the interval feature vector $\mathbf{y} = ([120.68, 190.53], [0.36, 0.65])$, where the grey level average and standard deviation calculated from the set of pixels of each segment takes values in the interval $[120.68, 190.53]$ and in the interval $[0.36, 0.65]$, respectively.

2.1 Regions

Let $C_k = \{\omega_{k1}, \ldots, \omega_{kN_k}\}, k = 1, \ldots, m$, be a class of individuals with $C_k \cap C_{k'} = \emptyset$ if $k \neq k'$ and $\cup_{k=1}^{m} C_k = \Omega$. The individual ω_{kl}, $l = 1, \ldots N_k$, is represented by the continuous feature vector $\mathbf{x}_{kl} = (x_{kl1}, \ldots, x_{klp})$.

A symbolic description of the class C_k can be obtained by using the join operator (Ichino et al. (1996)).

Definition 1. The join between the continuous feature vectors $\mathbf{x}_{kl}$ $(l = 1, \ldots, N_k)$ is an interval feature vector defined as $\mathbf{y}_k = \mathbf{x}_{k1} \oplus \ldots \oplus \mathbf{x}_{kN_k} = (x_{k11} \oplus \ldots \oplus x_{kN_k1}, \ldots, x_{k1j} \oplus \ldots \oplus x_{kN_kj}, \ldots, x_{k1p} \oplus \ldots \oplus x_{kN_kp})$, where $x_{k1j} \oplus \ldots \oplus x_{kN_kj} = [min\{x_{k1j}, \ldots, x_{kN_kj}\}, max\{x_{k1j}, \ldots, x_{kN_kj}\}]$.

We can associate two regions in $\Re^p$ to each class C_k: one spanned by the join of its elements and another spanned by the convex hull of its elements.

Definition 2. The *J-region* associated to class C_k is a region in $\Re^p$ that is spanned by the join of the objects belonging to class C_k. It is defined as

$R_J(C_k) = \{\mathbf{x} \in \Re^p : min\{x_{k1j}, \ldots, x_{kN_kj}\} \leq x_j \leq max\{x_{k1j}, \ldots, x_{kN_kj}\}, j = 1, \ldots, p\}$. The volume associated to the hyper-cube defined by $R_J(C_k)$ is $\pi(R_J(C_k))$.

Definition 3. The *H-region* associated to class C_k is a region in $\Re^p$ that is spanned by the convex hull formed by the objects belonging to class C_k. It is defined as $R_H(C_k) = \{\mathbf{x} = (x_1, \ldots, x_j, \ldots, x_p) \in \Re^p : \mathbf{x}$ is inside the envelop of the convex hull defined by the continuous feature vectors $\mathbf{x}_{kl} = (x_{kl1}, \ldots, x_{klp}), l = 1, \ldots N_k\}$. The volume associated to the internal points within the convex hull envelop defined by $R_H(C_k)$ is $\pi(R_H(C_k))$.

2.2 Graph concepts

The *mutual neighborhood graph (MNG)* (Ichino et al. (1996)) yields information on interclass structure.

Definition 4. The objects belonging to class C_k are each *mutual neighbors* (Ichino et al. (1996)) if $\forall \omega_{k'l} \in C_{k'}$ $(k' \in \{1, \ldots, m\}, k' \neq k), \mathbf{x}_{k'l} \notin R_J(C_k)$ $(l = 1, \ldots, N_{k'})$. In such a case, the MNG of C_k against $\overline{C_k} = \cup_{\substack{k'=1 \\ k' \neq k}}^{m} C_{k'}$, which is constructed by joining all pairs of objects that are mutual neighbors, is a complete graph. If the objects belonging to class C_k are not each mutual neighbors, we look for all the subsets of C_k where the elements are each mutual neighbors and which are a *maximal clique* in the MNG. In such a case, the MNG is not a complete graph. We can associate a *J-region* to each of these subsets of $C_k$and calculate the volume of the corresponding hyper-cube it defines.

In this paper we introduce an additional definition to the MNG.

Definition 5. The objects belonging to class C_k are each *mutual neighbors* if $\forall \omega_{k'l} \in C_{k'}, k' \in \{1, \ldots, m\}, k' \neq k, \mathbf{x}_{k'l} \notin R_H(C_k)(l = 1, \ldots, N_{k'})$. The MNG of C_k against $\overline{C_k} = \cup_{\substack{k' \neq k \\ k'=1}}^{m} C_{k'}$ defined in this way is also a complete graph. If the objects belonging to class C_k are not each mutual neighbors, again we look for all the subsets of C_k where the elements are each mutual neighbors and which are a maximal clique in the MNG. We can then associate *H-region* to each of these subsets of C_k and calculate the volume of the corresponding convex-hull it defines.

3 Symbolic classifier

This section introduces the learning and allocation steps of the symbolic classifier presented in this paper.

3.1 Learning step

The idea of this step is to learn the regions associated to each class so as to allow the classification of a new individual into a class through the comparison of the class description (regions) with a point in $\Re_p$ according to a dissimilarity matching function.

We have two basic remarks concerning this step. The first is that a difficulty arises when the objects belonging to a class C_k are not each mutual neighbors. In such a case, we look for all the subsets of C_k where its elements are each mutual neighbors and which are a *maximal clique* in the MNG (which is not a complete graph in such a case). However, it is well known that the computational complexity in time to find all cliques on a graph is exponential. It is then necessary to construct *an approximation* of the MNG.

The second remark concerns what kind of region (*J-region* or *H-region*) is suitable for describing a class C_k. Figure 1 illustrates the description of a class by a *J-region* and by a *H-region*. It is clear that the representation based on a *J-region* (see Ichino et al. (1996), Souza et al. (1999), De Carvalho et al. (2000)) over-generalizes the class description given by a *H-region*. For this reason, the latter option will be used in this paper.

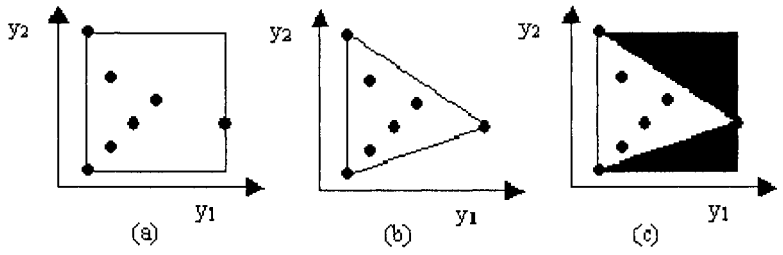

Fig. 1. (a) *J-region*, (b) *H-region*, (c) Over-generalization

The construction of the MNG for the classes C_k $(k = 1, \ldots, m)$ and the representation of each class by a *H-region* (or by a set of *H-regions*) is accomplished in the following way:

For $k = 1, \ldots, m$ do

1 Find the the region $R_H(C_k)$ (according to *definition 3*) associated to class C_k and verify if the objects belonging to this class are each mutual neighbors according to *definition 5*
2 If so, construct the MNG (which is a complete graph) and stop.
3 If this is not the case, (MNG approximation) do the following:
  3.1 choose an object of C_k as a seed according to the lexicographic order of these objects in C_k; do $t = 1$ and put the seed in C_k^t; remove the seed from C_k
  3.2 add the next object of C_k (according to the lexicographic order) to C_k^t if all the objects belonging now to C_k^t each remain mutual neighbors according to *definition 5*; if this is true, remove this object from C_k
  3.3 repeat step 2) for all remaining objects in C_k

3.4 Find the region $R_H(C_k^t)$ (according to *definition 3*) associated to C_k^t)
3.5 if $C_k \neq \emptyset$, do $t = t + 1$ and repeat steps 3.1 to 3.4) until $C_k = \emptyset$
4 construct the MNG (which is now not a complete graph) and stop.

At the end of this algorithm the subsets $C_k^1, \ldots, C_k^{n_k}$ of class C_k are computed and the description of this class is obtained by the *H-regions* $R_H(C_k^1), \ldots, R_H(C_k^{n_k})$.

As an example in the case of two classes, Figure 3 shows a) the complete Mutual Neighborhood Graph and the class descriptions based on *J-regions* (Ichino et al. (1996)), b) The MNG approximation and the class descriptions based on *J-regions* (Souza et al. (1999), De Carvalho et al. (2000)) and c) the MNG approximation and the class descriptions based on *H-regions* (the approach presented in this paper).

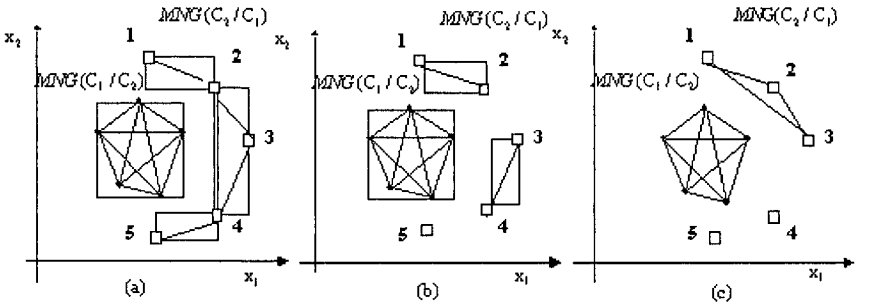

Fig. 2. (a) Complete MNG and *J-regions*, (b) MNG approximation and *J-regions*, (c) MNG approximation and *H-regions*

3.2 Allocation step

In the allocation step, a new object ω is compared with each class C_k and a dissimilarity score is computed according to a suitable matching function. Then, the minimal dissimilarity score is sought out and we assign the object ω to the class that corresponds to this minimal score.

Let ω be a new object to be assigned to a class C_k that is described by a continuous feature vector $\mathbf{x} = (x_1, \ldots, x_p)$. Remember that the subsets $C_k^1, \ldots, C_k^{n_k}$ of C_k are computed from the learning step.

The *classification rule* is defined as following: ω is affected to the class C_k if

$$\delta(\omega, C_k) \leq \delta(\omega, C_h), \forall h \in \{1, \ldots, m\} \tag{1}$$

where $\delta(\omega, C_h) = min\{\delta(\omega, C_h^1), \ldots, \delta(\omega, C_h^{n_h})$.

In this paper, the dissimilarity matching function δ is defined as

$$\delta(\omega, C_h^s) = \frac{\pi(R_H(C_h^s \cup \{\omega\}) - \pi(R_H(C_h^s)}{\pi(R_H(C_h^s \cup \{\omega\})}, \; s = 1, \ldots, n_h \qquad (2)$$

4 Monte Carlo experience

In order to show the usefulness of the method proposed in this paper, a special kind of SAR simulated image is classified in this section.

4.1 SAR simulated images

Synthetic Aperture Radar (SAR) is a system that possesses its own illumination and produces images with a high capacity for discriminating objects. It uses coherent radiation, generating images with speckle noise. SAR data display random behaviour that is usually explained by a multiplicative model (Frery et al. (1997)). This model considers that the observed return signal Z is a random variable defined as the product of two other random variables: X (the terrain backscatter) and Y (the speckle noise).

The process for obtaining simulated images consists in creating classes of idealized images (a phantom), and then associating a particular distribution to each class.

Different kinds of detection (intensity or amplitude format) and types of regions can be modelled by different distributions associated to the return signal. The homogeneous (e.g. agricultural fields), heterogeneous (e.g. primary forest) and extremely heterogeneous (e.g. urban areas) region types are considered in this work. According to Frery et al. (1997), we assume that the return signal in the amplitude case has the square root of a Gamma distribution, the K-Amplitude distribution and the G0-Amplitude distribution in homogeneous, heterogeneous and extremely heterogeneous areas, respectively.

Two situations of images are considered ranging in classification from moderate to greatly difficult. We generate the distribution associated to each class in each situation by using an algorithm for generating gamma variables.

The *Lee filter* (Lee (1981)) was applied to the data before segmentation in order to decrease the speckle noise effect. The segmentation was obtained using the region growing technique (Jain (1988)), based on the t-student test (at the 5% significance level) for the merging of regions.

Each segment (set of pixels) is described by two features (gray level average and standard deviation calculated from the segment set of pixels). The convex hull of a set of points (segments) in $\Re^2$ is defined as the minimal convex polygon encompassing these points. A number of algorithms have been developed to construct a convex hull from a given set of points. We have chosen the Graham scan algorithm (O'Rourke (1998)) because, it has the minimal time complexity ($O(nlogn)$, n being the cardinality of the set) among the thus far algorithms when applied to points in $\Re^2$.

4.2 Experimental evaluation

The evaluation of the approach presented in this paper (named here *H-region approach*, where class representation, MNG approximation and dissimilarity matching function are based on *H-regions*, is performed based on prediction accuracy, in comparison with the approach where class representation, MNG approximation and dissimilarity matching function are based on the *J-regions* (named here *J-region approach*).

The Monte Carlo experience was performed for images of sizes $64 \times 64, 128 \times 128$ and 256×256, taking into consideration situations 1 and 2. 100 replications were obtained with identical statistical properties and the prediction accuracy, speed and storage were calculated.

The prediction accuracy of the classifier was measured through the error rate of classification obtained from the test set. The estimated error rate of classification corresponds to the average of the error rates found for these replications.

The comparison according to the average of the error rate was achieved by a paired Student's t-test at the significance level of 5%. Table 3 shows the average error rate, suitable (null and alternative) hypothesis and the observed values of the test statistics for various sizes and the two image situations. In this table, the test statistics follow a Student's t distribution with 99 degrees of freedom, and μ_1 and μ_2 are, respectively, the average error rate for the *H-region approach* and the *J-region approach*.

From Table 3, we can conclude that in all cases (size and image situation) the average error rate for the *H-region approach* is lower than that for the the *J-region approach*. Also, the test statistics shows that the *H-region approach* outperforms the *J-region approach*.

SAR images	*H-region* Approach	*J-region* Approach	$H_0 : \mu_2 \geq \mu_1$ $H_1 : \mu_2 < \mu_1$
64×64 situation 1	5.78	8.29	-5.19
64×64 situation 2	24.83	24.92	-0.15
128×128 situation 1	2.68	3.42	-5.03
128×128 situation 2	16.52	16.89	-1.45
256×256 situation 1	1.39	1.87	-8.38
256×256 situation 2	13.67	14.34	-4.57

Table 1. Comparison between the classifiers according to the average error rate.

5 Conclusion

A new symbolic classifier based on a region-oriented approach is presented in this paper. At the end of the learning step, each class is described by a region

(or a set of regions) in $\Re^p$ defined by the convex hull formed by the objects belonging to this class, which is obtained through a suitable approximation of a Mutual Neighborhood Graph (MNG). This approach aims to reduce the over-generalization that is produced when each class is described by a region (or a set of regions) in $\Re^p$ defined by the hyper-cube formed by the objects belonging to this class. It then seeks to improve the classifier performance.

In order to show its usefulness, this approach was applied in the study of simulated SAR images presenting situations ranging in classification from "moderately easy" to "greatly difficult". The input (segments of images) is a set of continuous feature vectors. To assign a segment to a region, a dissimilarity matching function, comparing the class description (a region or a set of regions) with a point in $\Re^p$, was introduced.

The evaluation of the approach presented in this paper (called the *H-region approach* was based on prediction accuracy as measured through the error rate of classification obtained from the test set in comparison with the *J-region approach.* This measurement was accomplished in the framework of a Monte Carlo experience. The results showed that, concerning the prediction accuracy, the *H-region approach* outperforms the *J-region approach.* Future work must also consider the speed and storage performance of the *H-region approach* in comparison with the *J-region approach.*

Acknowledgments: The authors would like to thank CNPq (Brazilian Agency) for its financial support.

References

BOCK, H.-H. and DIDAY, E. (2000): *Analysis of Symbolic Data.* Springer, Heidelberg.

DE CARVALHO, F.A.T., ANSELMO, C.A.F., and SOUZA, R.M.C.R. (2000): Symbolic approach to classify large data sets, In: H.A.L. Kiers, J.-P. Rasson, P.J.F. Groenen, and M. Schader (Eds.): *Data Analysis, Classification, and Related Methods.* Springer, Berlin, 375–380.

FRERY, A.C., MUELER, H.J., YANASSE, C.C.F., and SANT'ANA, S.J.S. (1997): A model for extremely heterogeneous clutter. *IEEE Transactions on Geoscience and Remote Sensing, 1, 648–659.*

ICHINO, M., YAGUCHI, H., and DIDAY, E. (1996): A fuzzy symbolic pattern classifier In: E. Diday, Y. Lechevallier, and O. Opitz (Eds.): *Ordinal and Symbolic Data Analysis.* Springer, Berlin, 92–102.

JAIN, A.K. (1988): *Fundamentals of Digital Image Processing.* Prentice Hall International Editions, Englewood Cliffs.

LEE, J.S. (1981): Speckle analysis and smoothing of synthetic aperture radar images. *Computer Graphics and Image Processing, 17, 24-32.*

O'ROURKE, J. (1998): Computational Geometry in C (Second Edition), Cambridge University Press, New York.

SOUZA, R.M.C.R., DE CARVALHO, F.A.T., and FRERY, A.C. (1999): Symbolic approach to SAR image classification. *IEEE 1999 International Geoscience and Remote Sensing Symposium*, Hamburg, 1318–1320.

Two-Mode Cluster Analysis via Hierarchical Bayes

Wayne S. DeSarbo, Duncan K. H. Fong, and John Liechty

Marketing Dept., Smeal College of Business, Pennsylvania State University, University Park, PA, USA 16802

Abstract. This manuscript introduces a new Bayesian finite mixture methodology for the joint clustering of row and column stimuli/objects associated with two-mode asymmetric proximity, dominance, or profile data. That is, common clusters are derived which partition both the row and column stimuli/objects simultaneously into the same derived set of clusters. In this manner, interrelationships between both sets of entities (rows and columns) are easily ascertained. We describe the technical details of the proposed two-mode clustering methodology including its Bayesian mixture formulation and a Bayes factor heuristic for model selection. Lastly, a marketing application is provided examining consumer preferences for various brands of luxury automobiles.

1 Introduction

Two-mode cluster analysis involves the simultaneous and joint amalgamation of both the row and column objects contained in a two-mode data matrix. Examples of such two-mode data include: asymmetric two-mode proximity data (e.g., confusions data), two-way dominance data (e.g., subjects eliciting preferences or choices with respect to different column objects), two-way profile data (e.g., objective quantitative features or attributes for a set of designated objects), etc. A number of psychometric and classification related procedures for the clustering of such two-mode data have been published over the past few decades (see DeSarbo, Fong, Liechty, and Saxton, 2003 for an excellent literature review on two-mode clustering).

Bayesian approaches to traditional *one-mode* cluster analysis began with the seminal work of Binder (1978) who described a general class of normal mixture models and introduced various ingredients of Bayesian approaches to classification, clustering, and discrimination into this finite mixture framework. Later, work on Bayesian estimation of finite mixture models for classification via posterior simulation followed by Gilks, Oldfield, and Rutherford (1989), Diebolt and Robert (1994), Gelman and King (1990), Verdinelli and Wasserman (1991), Evans, Guttman and Olkin (1992). Lavine and West (1992) extended Binder's (1978) work by applying an iterative resampling approach to Monte Carlo inference, Gibbs sampling, to this same mixture framework, stressing the ease with which such analyses may be performed in more general settings. Their Bayesian framework allowed for the generalization to several normal mixture components having different covariance