Improving Pseudobagging techniques

Angela Chieppa\textsuperscript{a,1}, Karina Gibert\textsuperscript{a}, Ignasi Gómez-Sebastià\textsuperscript{b}, Miquel Sánchez-Marrè\textsuperscript{b}

\textsuperscript{a}Statistics and Operations Research Department
\textsuperscript{b}Computer Software Department
Technical University of Catalonia (UPC), Barcelona

Abstract. We present an important improvement related to the computation and use of Mutual Information index in Pseudobagging, a technique that adapts “bagging” to unsupervised context. The Mutual Information index plays a key role in this technique, assessing the quality of a partition. We propose the use of such an index to improve the Pseudobagging voting scheme for determining the final partition of the data. Issues related to the estimation of Mutual Information index in the multivariate continuous case become crucial for the application of Pseudobagging to real data: we discuss some practical approaches to computation in this situation. Finally, experimental results are presented, related to application of new “pooled voting” scheme and to the evaluation of the impact of different computing methods for Mutual Information.

Keywords. cluster ensemble, discretization, mutual information, pooled voting.

1. Introduction

Class discovery in an unknown domain is the goal of clustering algorithms, which usually result in partitioning the observational dataset: for many years efforts have been concentrated upon improving single algorithms, but recently this problem has been addressed in a different way \cite{1,5}: combining the results of multiple clusterings in order to obtain “robust” data partitions, as supervised ensemble techniques such as bagging \cite{1} do.

Formally, given N different available partitions of the data X, a clustering ensemble $\Pi = \{P_1, P_2, \ldots, P_N\}$, is defined, where $P_i = \{C_{i1}, C_{i2}, \ldots, C_{i\xi_i}\}$ is a single partition with $\xi_i$ clusters.

The goal consists in defining a final “consensus” partition $P^*$, result of a combination of the N partitions of $\Pi$. Determination of $P^*$ always involves the use of a “quality” measure for partitions: authors agree on the use of the Mutual Information index as a measure of how well final partition reproduces information in original data \cite{5} or shared in cluster ensemble \cite{1}.

Pseudobagging technique \cite{5} is one of these cluster ensemble techniques: one of the specificities of Pseudobagging is the use of Mutual Information and Inertia as quality measures, useful both either in an intermediate step of the Pseudobagging process and for the assessment of goodness of the final results.
Computation of Mutual Information index requires the estimation from observed data of joint density of all the attributes considered: in the multivariate continuous case, where numerical integration is required, this involves some computational handicaps. These issues have been addressed in several works [6], [9]: in this paper, the computation of the index on discretized data is proposed, as other interesting methods still are very difficult to apply when dealing with more than a few variables.

Experimental results related to robustness of Mutual Information index using different discretization methods are presented. Tests have been run using a set of data coming from a waste water treatment plant [4] and Data Analysis Intelligent System GESCONDA [4], which has been specifically designed and developed at Technical University of Catalonia (UPC) and implements Pseudobagging, among others.

The paper starts describing clustering ensemble methods and Pseudobagging. Then questions on Mutual Information computation with numerical attributes are addressed, and methods chosen for discretization are described. A description of how MI could be used for weighting voting follows. Finally, experimental results are presented. The paper ends with some conclusions and future work.

2. Cluster Ensemble Methods and Pseudobagging

Cluster ensemble methods are the “unsupervised version” of more general multi-classifiers systems, that improve the accuracy of a learning algorithm by conveniently “combining” the results of a multiple or recursive application of it [1].

In unsupervised framework, the lack of knowledge about true class labels makes impossible to express a prediction error and consequently a loss function, that along with its minimization allows increasing accuracy in final classification and comparing the effectiveness of different methods: so some “criteria” or new objective function to evaluate final partition has to be defined. In clustering ensemble methods these proposed criteria take into account two main properties: first of all, goodness of fit to the original observed data, the more the partition resembles the original data, the better; then specificity of problem suggest also to consider the consistency with the clustering ensemble, involving the agreement between P* and single P_i partitions in the ensemble (P* as a consensus partition).

Recent works [1],[11] agree in the definition of the normalized average mutual information of a partition as the objective function to be maximized: this function is a measure of the shared information of a single partition with all the others, according to the Mutual Information index. However, this function is impractical for direct optimization so proposed techniques mostly use empirical validation and heuristics.

There are different cluster ensemble techniques using Mutual Information to assess quality of partition. Fred and Jain [1] introduce the concept of evidence accumulation clustering that maps the individual data partitions in a clustering ensemble into a new similarity measure between patterns, summarizing inter-pattern structure among clusterings; a final data partition is obtained by applying the single-link method to the new similarity matrix. Strehl and Ghosh [11] explore the concept of consensus between data partitions, using graph-theoretical approaches for consensus decisions, based on a cluster matching paradigm.

Pseudobagging [5] adapts classical bagging supervised technique to an unsupervised context: partitions are obtained making use of repeated runs of k-means (notice that other partitioning techniques are possible), with random initial seeds.
Randomizer effects, that could lead to partitions of very different qualities, are mitigated by combining partitions; a simple voting scheme is used to combine different partition results, each partition voting with a single vote. In the unsupervised context, combining labels from several iterations requires a pre-process of relabelling all partitions with regards to a reference one, since the labels are automatically produced in the clustering process and the same class may change label from one iteration to another (making differences only apparent); that's why before applying voting, best partition is chosen, on the basis of Mutual Information Index or Inertia, and labels of this partition are used to re-label correspondent ones in other partition to be combined. Finally, quality of resulting consensus partition is judged, with MI or Inertia again.

3. First implementation of Pseudobagging in GESCONDA

Pseudobagging technique has been implemented in GESCONDA, an Intelligent Data Analysis System, designed and developed at Technical University of Catalonia (UPC) [2], for knowledge management mostly in environmental databases. This system has been developed in Java and integrates both statistical and artificial intelligence data mining techniques.

First implementation of Pseudobagging was very useful to test main features of algorithm [5]: increased quality of bagged final classification and results became stable using just a few partitions to combine. This first implementation, however, presented some drawbacks. First, it used a simple majority voting scheme, that is to say that each partition has the same weight when voting, which is quite inefficient, taking into account that different partitions have different quality that is already taken into account in re-labelling step of the Pseudobagging process. Moreover, the computation of the MI index on previously discretized continuous attributes can change depending on the underlying discretization method used. Following paragraphs address these issues and present methodological solutions chosen to improve the quality of the proposal.

4. Computation of Mutual Information with Continuous Attributes

Original MI index was proposed to measure distance from independence between two variables: extension to multivariate case is intended as a measure of simultaneous interaction or a sort of a multi-way similarity: if 0, variables “do not interact” [9]. Formally, given $X_1, ..., X_K$ variables, the mutual information MI is calculated as:

$$MI(X_1, ..., X_K) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} f(x_1, ..., x_K) \log \frac{f(x_1, ..., x_K)}{f_1(x_1) \ldots f_K(x_K)} \, dx_1 \ldots dx_K$$

where $f(x_1, ..., x_K)$ represents the observed joint density function between $X_1, ..., X_K$ and each $f_k(x_k)$ the univariate density function of $X_k$, with $k=1,2, ..., K$. When the variables are categorical, it is possible to add the frequencies of the modalities of those variables as estimations of the probabilities. In this case, every $X_k$ can take a set of possible discrete value $D_k=\{x_{k1}, ..., x_{kt}\}$ and the formula becomes:
\[ MI(X_1, X_K) = \sum_{x_1 \in D_1} \ldots \sum_{x_K \in D_K} p(x_1, \ldots, x_K) \log \frac{p(x_1, \ldots, x_K)}{p_1(x_1) \ldots p_K(x_K)} \]

where \( p(x_k) = P(X_k = x_k) \), probability approximated by observed relative frequency of \( x_k \) in the dataset and \( p(x_1, x_2, \ldots, x_k) \) is the joint probability of “\( X_1 = x_1 \) and \( X_2 = x_2 \) ... and \( X_K = x_K \)” approximated similarly.

Problems in computation of MI arise when working with continuous variables: how to estimate the joint density function? Many methods have been developed to solve this density estimation problem, using a kernel density approach or a segmentation of variables spaces [9][12][7][6]: a completely satisfactory method to deal the question when a big number of numerical attributes is involved is difficult to find or, at least, very difficult to implement. Managing this computation by previously discretizing numerical attributes seems to be a good solution from a practical point of view, as long as discretization process doesn't imply relevant information loss: that’s why it’s important to evaluate the “robustness” of MI index with respect to different discretization methods used.

There are a lot of different methods [8], we concentrate on three methods: equidistant method, for its simplicity and diffusion could be considered a baseline; boxplot-based discretization method [4], which has given very good results and for this reason already was implemented in GESCONDA; an entropy-based discretization method, using Fayad-Irani algorithm [10][8].

4.1. Equidistant discretization

Equal width interval binning is perhaps the simplest method to discretize data and surely one of the most used. It divides, after sorting observed values, the range of the variable to be discretized into \( l \) equally sized bins, where \( l \) is a parameter introduced by user. Formally, if a variable \( X \) has boundary values \( x_{\min} \) and \( x_{\max} \), then discretization is obtained determining first of all the bin width \( \delta = (x_{\max} - x_{\min}) / l \), and then constructs cutpoints at \( (x_{\min} + i\delta) \), where \( i = 1, \ldots, l-1 \). Its main drawback is “sensitivity” to outliers, that may drastically skew the range; also, this may produce disequilibrium in the sense of producing intervals with very different frequencies.

4.2. Boxplot-based discretization

The Boxplot based discretization is presented in [3]: it's a supervised method which uses the same idea of boxplot graphical representation to induce cutpoints for partitioning numerical attributes on the basis of a previous classification. Given a partition \( P \) with \( \zeta \) classes, method consists in:

1. for each class \( C_j, j = 1, \ldots, \zeta \), computing minimum (\( m_j \)) and maximum (\( M_j \)) value for \( X_k \), numerical variable to be discretized; a set \( M \) with all these values is obtained \( M = \{m_1, M_1, \ldots, m_\zeta, M_\zeta\} \)
2. determining set \( Z = \{z_1, z_2, \ldots, z_\zeta\} \) of cutpoints by sorting \( M \)

The bins obtained do not have the same length and the set of intersecting classes is constant all along every interval and changes from one to another. An interesting
property of this method is that it produces a discrete variable which has maximum
association with partition $P$.

4.2.1. Fayyad-Irani Entropy Discretization

This supervised algorithm is a generalization of a binary discretization algorithm based
on the minimization of information entropy, calculated on the basis of the class labels:
it finds the best split so that the bins are as pure as possible, i.e. the majority of the
values in a bin have the same class label. For details, better referring to the original
article [10]. The core binary discretization technique is based on determining cutpoint
$T$ by the minimization of the class information entropy of partition induced by $T$:

$$E(X_k, T, S) = \frac{|S_1|}{|S|} \text{Ent}(S_1) + \frac{|S_2|}{|S|} \text{Ent}(S_2)$$

where $S$ is the set of instances in observed dataset, $S_1$ is the subset including all
instances with a lower value of $X_k$ than $T$ and $S_2 = S - S_1$; finally, $\text{Ent}(S_i)$ is the entropy
of a subset $S_i$ with reference classification.

The process is recursively applied to partitions obtained until the stopping criterion
is met. The use of the “Minimal Description Length Principle” to decide when stop
splitting is the novel idea presented by the authors of this algorithm.

5. Determining final partition in Pseudobagging: the voting scheme

In the first implementation of Pseudobagging process [5], the final class is assigned
based on a most frequent class according to a basic count: every partition $P_i$ in $\Pi$ gets
one vote, regardless its quality and the most voted class is the one assigned. This can
drive to a biased Pseudobagging class assignment that does not correspond with the
reality.

A pooled voting scheme is clearly better: quality index, either the Inertia or MI,
already computed for Pseudobagging during relabelling phase, could be used to assign
a number of votes to each partition. Two factors are considered:

• Normalized Index: this factor reflects how much the current execution is
  contributing to the total sum of quality Indexes, in a percentage coefficient.

• Number of distributed votes: this factor reflects the total number of votes to
  distribute among all partitions. Higher number of votes results in higher precision.

Each partition is assigned a number of votes according to normalized index
coefficient, that is to say in the same proportion to total number as its quality is
providing to total quality: thus, if a given execution is contributing the double than
other to total quality, it will get a double number of twice as much votes.

6. Application

Data analyzed in this paper comes from the wastewater Treatment Plant (WWTP) of
Girona, Spain. It is a sample of 396 observations taken from September the first of
1995 to September the 30th of 1996 [4]. Each observation refers to a daily mean: the
state of the Plant is described through a set of 25 variables, considered the more relevant upon expert’s opinions. We completely relied on GESCONDA for experiments; version of system being the one with new features implementing pooled voting and discretization methods.

Two main experiments were executed: one to test quality increase of final partition determined by Pseudobagging when using pooled voting instead that simple majority scheme; second test was to prove robustness of MI and of the overall Pseudobagging process towards different computation methods, e.g. different type of discretization of numerical attributes.

Experiments design was strongly affected by results of previous works on the same data [4][5], namely that four is the right number of classes for these data and that ten partitions are enough to get a good classification on these data with Pseudobagging.

To set up both experiments, 20 executions of K-means on WWTP data were performed (P1, ..., P20), every execution providing a partition of the 396 days in 4 classes. Then, using increasing sequence of these partitions, 20 cluster ensemble were built Π1=(P1), Π2=(P1,P2)... Π20=(P1,...P20).

6.1. Evaluating pooled voting

For first experiment on evaluation of pooled voting, Pseudobagging was applied to the 20 cluster ensembles Π1=(P1), Π2=(P1,P2)... Π20=(P1,...P20. Firstly, a simple voting scheme was used to find a consensus partition for each of them; then, a pooled voting scheme was used, producing a new series of 20 partitions. So, at the end, 3 series of partitions were available: one including the single k-means executions and other two series with "consensus" partitions resulting from Pseudobagging with the two different voting scheme.

For each Pi partition in one of these three series and considering the K attributes in the dataset, the Mutual Information Index MI(X1,...XK,Pi) was computed, using equidistant discretization technique for numerical attributes, as a quality measure.

![Figure 1. Quality of final partition according to voting scheme](image-url)
Results (Figure 1) clearly show very good performance of pooled voting, with MI indexes always higher than the ones obtained with simple voting scheme. More, when a “bad” partition enters the ensemble (as partitions 6, 10 and 12) pooled voting is not so strongly affected as simple voting.

In these experimental results, the five initial ensembles considered are composed by higher quality partitions, while low quality ones enter in the following ensembles: this explains the big boost of pooled voting MI series at the beginning and the progressive tendency to go down; when a very good MI partition occurs again, the change in tendency doesn’t take place because of the occasionality of the fact and of the dimension of the ensemble, that “dilute” the impact.

6.2. Computing Mutual Information with Continuous Attributes: evaluation of effects of discretization method

Firstly, to evaluate effects of different discretization methods when computing MI index, for each partition $P_i$ of the 20 single k-means executions, $MI(X_1,\ldots,X_K,P_i)$ was calculated using different methods, resulting in 3 series of values; for supervised discretization techniques, the same partition $P_i$ was used for determining classes.

First thing to remark (Figure 2) is that lower values of MI appear when computing with Fayyad-Irani discretization method. Second remark is the fact that the distribution of the three series is rather similar, that in the Pseudobagging context, where MI index is used to rank partitions of the ensemble, implies that different techniques analyzed produce similar results.

The availability of a reference partition for these data, obtained in previous works [4] and validated by experts, suggested the computation of the “accuracy” of partitions used in the experiment. This accuracy was calculated using a feature of GESCONDA, which provides, among other things, the percentage of correctly classified observations.
So, in order to evaluate the overall effect of discretization of MI in the entire Pseudobagging process, partitions obtained using Pseudobagging on $\Pi_1, \ldots, \Pi_{20}$ implementing different discretization method were compared through accuracy index.

![Comparison of Pseudobagging results accuracy using different discretization methods and with single k-means executions](image)

Figure 3 shows first of all good properties of Pseudobagging: high variability in accuracy series of single k-means partitions is contrasted by accuracy of pseudobagged partitions, more stable whatever the method. More, there isn’t a technique that produces better results than others. So, this experimental results show that in Pseudobagging framework, different discretization methods to compute Mutual Information do not affect final results.

7. Conclusions and future work

In this work special attention is given to Mutual Information index in the context of Cluster ensemble methods: more specifically, in Pseudobagging technique [5]. This technique in its first implementation already made use of MI in a specific step to assess quality of partition: this work proposes the use of the same index also to “weight” voting scheme; results of application to real data prove quality increase in final partition.

Discretization of continuous attributes is one of the main tools used to deal with computational difficulties arising in the continuous multivariate case. Results of application of Pseudobagging techniques using three different discretization methods on data coming from a Waste Water Treatment Plant are discussed, to assess MI computation robustness and effect of different discretization on Pseudobagging results.
Experimental results of this case-study doesn’t give strong evidence for a discretization technique against others and show that quality of Pseudobagging is not affected by the choice of the methods for computing Mutual Information index.

Future work will first of all validate results of this work with other datasets; then other interesting ways of computing MI [7] and new possibilities to find final consensus partition will be explored.

References