Pseudobagging: Improving class discovery by adapting bagging techniques to clustering algorithms

Karina Gibert[1], Isaac Pinyol[2], Luís Oliva[3], Miquel Sànchez-Marré[3]

[1]Statistics and Operations Research Department
Technical University of Catalonia
Campus Nord-UPC
Barcelona
karina.gibert@upc.edu

[2]Artificial Intelligence Research Institute
Spanish Council for Scientific Research
Campus UAB
Bellaterra
ipinyol@iiia.csic.es

[3]Software Department
Technical University of Catalonia
Campus Nord-UPC
Barcelona
{loliva,miquel}@lsi.upc.edu

Abstract

In this paper adaptation of bagging techniques to increase stability of classes discovered by a clustering algorithm is addressed. Experiments on real data coming from a Waste Water Treatment plant are performed using K-means, which is one of the most used algorithms for classes discovery.

Repeated use of Kmeans on the same set of data may produce different results since the algorithm starts by generating as random seeds as classes required by the user and they are used as class-centroids at first iteration. Since the results of Kmeans are order dependent, the classes found depend of the position of the initial seeds and results may substantially change from one run to another.

In this paper, the adaptation of bagging techniques to repeated runs of Kmeans are used to get convergence.

keywords: bagging, clustering, inertia, entropy, mutual information, convergence, separability, class discovery, waste water treatment plant.

1 Introduction

Clustering algorithms are the most suitable techniques for discovering existing classes in an unknown domain. Clustering algorithms are unsupervised techniques, devoted to discover the unknown structure of a domain. There are many families of algorithms. Among them, the partition algorithms, based in random seed initialization (like initial election of seeds in the KMeans algorithm), present some problems to obtain trustworthy results. Resulting classes highly depend on the location of the initial seeds and different runnings of the algorithm with same data, may produce different results.

Recommendations on some initialization techniques is found in literature for decreasing this random effect [2] [4] [1] . But, these techniques present other problems such as local optimum or biased solutions.

In this paper the adaptation of bagging techniques, commonly used in supervised environments, is proposed for getting stable classes. Basically, bagging techniques compute a final partition through the combination of several ones made on the same data.

In this paper, the idea of bagging techniques is used to mitigate the effects, in class discovery, of the random initialization of the clustering algorithms. The proposed methodology is general for any other clustering algorithm, provided it requires random initialization, or even to combine results provided for different clustering algorithms.

Main idea is to repeat clustering with random seeds and to get different classifications of the objects. Our guess is that combining these results will mitigate the random effect and will produce more stability in the discovered classes.

On the other hand two criteria are proposed for evaluation of the goodness of a classification. One

of them is based on the separability of classes and the other one on Shannon's measure of mutual information.

Proves using pseudobagging with increasing lenght sequences of classifications are done over a real set of data coming from a waste water treatment plant and quality of resulting classifications are assessed using the criteria proposed in this paper. All implementations and tests were run on the Data Analysis Intelligent System GESCONDA [6], which has been specifically designed and developed by the authors.

The structure of the paper is the following: First the Kmeans and the standard bagging techniques are introduced. Then pseudobagging is described and technical details provided. Finally the real application domain from which data was taken is presented together with experimental results. The paper ends with some conclusions and future work.

2 Basic concepts

In this section the clustering algorithm which was used (KMeans) is described as well as the general principles of bagging techniques.

2.1 K-Means

Maybe, KMeans [8] is the most known non-hierarchical clustering algorithm and its also the most used in data mining because of its simplicity and intuitive structure, as well as for its ef"ciency and good results that usually gives in specific environments[1].

```
Algorithm KMeans
Input:
    E: Set of instances
    K: Integer (number of classes)
     : Real (minimum increment)
Output:
    P: Partition
    Generate K prototypes, choosing K random instances
    for each e E
        Assign e to the nearest prototype
    End for each
    S1 := Sum the square of distances(instance,prototype)
    do
        Recalculate prototypes
        Reassign instances to the nearest prototype
        S2:=S1
        S1:=Sum the square of distances(instance,prototype)
    while S1-S2
End of Algorithm
```

However, as it can be seen the algorithm starts by generating K random seeds and they are used as class centroids atfirst iteration. This determines thefi rst partition and, as a consequence, the resultingfinal classes, which highly depend on the position of the initial random seeds. Two runs of Kmeans with same data use to provide different results, unless the real underlying class-structure is very strong and clear. This is a common situation of all clustering techniques order-dependent, based on random initialization.

2.2 Bagging

Purely, bagging is a technique that belongs to supervised learning which combines the results obtained in different supervised classifications of the same set of objects to generate afinal grouped partition of the data.

There exist many different criteria to recognize the class of a set of instances, providing a myriad of supervised learning algorithms and rule induction algorithms. Classes predicted for every instance are usually different depending on the supervised algorithm used. Bagging is devoted to solve the biases on class recognition of different algorithms [4].

Bagging techniques, basically consists on two steps: First, several class predictions for every instance of the set, by using different supervised classification algorithms are obtained; then different predictions are compared, for every single instance, and the most frequent predicted class isfnally assigned to the instance.

3 Pseudobagging: Adapting bagging techniques to unsuper vised clustering

In this work bagging principles are transferred to improve clustering problem, classically unsupervised, where the underlying structure of the domain is unknown and needs to be discovered.

[1] Hierarchical clustering also use to be very popular but, since they have quadratic complexity, they are prohibitive in data mining applications with big sets of data.

As said before, repeated runs of a clustering algorithm with random initialization produce different partitions of the same set of data. In this paper, Kmeans is particularly used as underlying clustering algorithm, but the proposed methodology is general for any other clustering algorithm, provided it requires random initialization, or even to combine results provided for different clustering algorithms.

Basically, a clustering process is iteratively performed over data and then, the best partition is selected as a reference partition to perform the bagging. The *pseudobagging* algorithm implemented can be described as it follows:

Algorithm *Pseudobagging*
Input:
E: Set of instances
$(C'_1, \ldots, C'_r)$: Set of classifications
Output:
P: Partition

Find the Reference Partition C_a
$(C'_1, \ldots, C'_r)$= Relabel $(C_1, \ldots, C_r)$ according C_a
for each *$e \in E$*
Get $C'_1(e), \ldots, C'_r(e)$
Assign e to the most frequent class
End for each

End of Algorithm

Specially in the case of clustering algorithms that are order dependent with random initialization, as it is the case of Kmeans, we think that using pseudobagging techniques, could give better results that just taking a single run of the algorithm as definite classification. Because of the randomness, some classifications would be better and some other worse, and the goal is to find the real underlying class structure of the domain. The idea is to use bagging techniques for combining the results of several runs of the algorithm (producing a set of classifications with some variability) to obtain a final classification that would balance between good and bad ones.

Our proposal introduces two steps into the general bagging procedure specifically oriented to adapt bagging techniques to the context of unsupervised clustering, which are described bellow.

3.1 Finding the reference partition

First of all, one of the classifications obtained with the (unsupervised) clustering algorithm is identified as a reference partition, according to some criteria, and it is then used as the real class label of the instances to proceed with the bagging steps classical in supervised classification. The main difference here is that in real bagging, the class label is certainly known, while in pseudobagging the class label is calculated over data and may include predictive errors. However, this reference partition is only used to standardize comparisons among the results of other runs and it is not used to compute predictive errors and so on, which would not be correct.

In this work, three criteria for identifying the reference partition have been implemented in GESCONDA:

1. **First**: Just choose the results of the first running as reference partition. This is a very fast method that can produce bad results if this first classification is very far from real classes.

2. **Inertia**: The reference partition is the one that optimizes the I coefficient introduced in next section. This coefficient is a measure of the goodness of a classification in terms of separability. Using this method would mean choosing the classification with highest value of I.

3. **Mutual Information**: The chosen classification is the one that provides the most quantity of information. Using Shannon's Mutual Information calculation the chosen classification would be the one with highest value of entropy, as presented in next section.

Details about Inertia and Mutual Information criteria are provided in next section.

After determining a reference partition, relabelling according to that partition is required as it is explained bellow.

3.2 Relabelling

In classical bagging, the class of every instance e is previously known and different classifiers are used to predict this class. The labels used for each class are maintained constants all along the process and label l is always used for same class.

In clustering context the situation is radically different. Although all the classifications $(C_1, \ldots, C_r)$ have the same number of classes, each of them is built independently of the others

and emerging classes are directly labelled by the clustering algorithm. As a consequence, a single class discovered simultaneously by C_i and C_j, can be differently labelled in each classification (i.e. a class numbered as 1 in C_i may be named 4 in C_j and class 1 of C_j don't have any relation with class 1 in C_i). Because of that, it is required to build a correspondence table, by identifying the labels used for the same classes in different classifications.

First, a reference classification C_a is chosen to build up a correspondence table. Classification C_a would be the base to calculate the correspondences with other classifications. Each class c, belonging to another classification C_i, would be assigned to the most similar class c' of C_a. That is, c is referred to c' if the prototype of class c has the minimum distance to the prototype of c'. When possible, the distance used for building these correspondences, should be the same that has been used to produce the set of classifications $C_1 \dots C_r$.

Performing such a correspondence table, all the classifications will share a common set of labels to refer the classes and assignments of an instance to every class can be correctly counted.

4 Measuring the Goodness of a Partition

As said before, pseudobagging requires to measure the goodness of a partition in order to select which is the best one to mark it as reference partition.

When dealing with supervised environments it is quite easy to measure the goodness of a partition provided by a classifier, as soon as part of the data is saved from the training set and uses as a validation test, simply comparing the real class with the predicted one.

However, in non supervised environments no information at all is available to know the real class of an instance, and comparison between real class and predicted one is no more possible. As a consequence ratios of misclassification cannot be provided either. That is the reason why other tools that assess the goodness of the classes are required. At least some way to know how well done are the classes at a structural level should be found. In this work, two measures are proposed for this purposes: Inertia, and Shannon's mutual information.

4.1 Some notation

Here, some notation is introduced to clarify the development from now on. Data is represented in a matrix of k columns $(X_1, \dots, X_k)$ and n rows where the row $i = (x_{i1}, \dots, x_{iK})$ represents the instance i of the data set. The vector P of length n will be the categorical variable that represents the classification being evaluated. Then, the instance i belongs to the class c_i.

4.2 Inertia

The coefficient I is, in fact, a measure of the separability of the classes and homogeneity inside classes. This measure is based on the classical F statistic, and calculates the ratio between the inertia within classes (which increases with the heterogeneity of a class) and the inertia between classes(which increases with distinguishability among the classes). Here, a multivariate formulation is presented. It is important to notice here that I coefficient can be only calculated over numerical variables.

Let n_c be the number of elements of class c. Let ξ be the number of classes of P and K and the number of variables. Let $\overline{c} = (\overline{x}_{c1}, \dots, \overline{x}_{cK})$ be the centroid of the class c, where $\overline{x}_{ck} = \frac{\sum_{\forall i \in c} x_{ik}}{n_c}$ and let $\overline{x} = (\overline{x}_1, \dots, \overline{x}_K)$ be the global centroid of the whole set of data, where $\overline{x}_k = \frac{\sum_{\forall i} x_{ik}}{n}$. Let $d(i,j)^2 = \sum_{k=1}^{K} (x_{ik} - x_{jk})^2$ be the quadratic Euclidean distance between objects i and j. Then, the inertia within classes (S_w^2) is defined as

$$S_w^2 = \frac{\sum_{\forall c \in P} (n_c - 1) S_c^2}{n - \xi} \tag{1}$$

where

$$S_c^2 = \frac{\sum_{\forall i \in c} d(i, \overline{c})^2}{n_c - 1} \tag{2}$$

The inertia between classes (S_ξ^2) is

$$S_\xi^2 = \frac{\sum_{\forall c} d(\overline{c}, \overline{x})^2}{n - \xi} \tag{3}$$

Finally, the total inertia ratio (I) is written as

$$I = \frac{S_\xi^2}{S_p^2} \tag{4}$$

Coefficient I grows when the variance between classes increase (S_ξ^2) and the variance within every class decrease (S_w^2). That is, the more separated are the classes and the more compact, the greater is the value of I.

4.3 Mutual Information

This approach is based on Shannon's mutual information. Originally, the mutual information between two variables $X_k, X_{k'}$ is defined as:

$$I(X_k, X_{k'}) = \\ = \int_x \int_{x'} f_{k,k'}(x, x') \log\left(\frac{f_{k,k'}(x,x')}{f_k(x) f_{k'}(x')}\right) dx\, dx' \quad (5)$$

being $f_k(x)$ the density function of $X_k, \forall k$ and $f_{k,k'}(x, x')$ the bivariate density for X_k and X'_k.

In the praxis, this is easy to calculate when $X_k, X_{k'}$ are qualitative, since in this case it is possible to add the frequencies of the modalities of those variables, as estimations of the probability functions. Let $D_k, D_{k'}$ be the set of values taken by two qualitative variables $X_k, X_{k'}$.

$$I(X_k, X_{k'}) = \\ = \sum_{x \in D_k} \sum_{x' \in D_{k'}} P(x, x') \log\left(\frac{P(x,x')}{P(x)P(x')}\right) \quad (6)$$

where,

$$P(x) = \frac{card\{i : x_{ik} = x\}}{n}$$

,

$$P(x') = \frac{card\{i : x_{ik'} = x'\}}{n}$$

$$P(x, x') = \frac{card\{i : x_{ik} = x \wedge x_{ik'} = x'\}}{n}$$

In the general multivariate formulation, and taking P as a reference partition:

$$M(X_1, \ldots, X_K, P) = \\ = \sum_{x \in D_1} .. \sum_{x \in D_K} \sum_{c \in P} F(x_1, \ldots, x_K, c) \quad (7)$$

where

$$F(x_1, \ldots, x_K, c) = \\ = P(x_1, \ldots, x_K, c) \log\left(\frac{P(x_1 \ldots x_K, c)}{P(x_1 \ldots x_K)P(c)}\right) \quad (8)$$

The same expression is also useful for discrete numerical variables. Numerical ones should be discretized before, but I coefficient can be used instead.

4.4 Structural validation

The two measures introduced before are also suitable for evaluating the quality of a resulting partition obtained by any clustering process, taking into account different aspects, either the separability of the classes or the quantity of information contained in data. The greater are the values of I and M, the best is the partition.

5 Application

The main goal of wastewater treatment plants is to guarantee the outflow water quality (referred to certain legal requirements), in order to restore the natural environmental balance, which is disturbed by industry waste or domestic wastewater.

The process used to achieve this goal is highly complex; on the one hand, because of the intrinsic features of wastewater treatment processes; on the other hand, because of the bad consequences of an incorrect management of the plant [7], [3].

A very brief description of the process in the plant is presented: the waste water flows sequentially through three processes which are commonly known as pretreatment, primary and secondary (see [9] for a detailed description of the process). Figure 1 depicts its general structure: *(i):* In the *pretreatment*, an initial separation of gross solids, oils and greases from wastewater is performed. *(ii) Primary* treatment consists of leaving the wastewater in a primary settler for some hours. Suspended solids will deposit down the settler and could be removed from the water. *(iii) Secondary* treatment occurs inside a biological reactor. A population of microorganisms (biomass) degrades the organic matter solved in the wastewater. A secondary settler is used to separate the treated water from the biomass. The primary and secondary settler outputs (solids

and biomass) produce a kind of mud which is the input of another set of processes in the WWTP called *sludge line*.

Data analyzed in this paper comes from the wastewater Treatment Plant of Girona (in Spain). It is a sample of 396 observations taken from September the first of 1995 to September the 30th of 1996. Each observation refers to a daily mean, and it is identified by the date itself.

The state of the Plant is described through a set of 25 variables, considered the more relevant upon expert's opinions. They can be grouped as: *(i)* Input (measures taken at the entrance of the plant): Q-E: Inflow wastewater (daily m^3 of water);FE-E Iron pre-treatment (g/l); pH-E; SS-E: Suspended Solids (mg/l); SSV-E: Volatile suspended solids (mg/l); COD-E: Chemical organic matter (mg/l); BOD-E: Biodegradable organic matter (m/l). *(ii)* After Settler (measures taken when the wastewater comes out of the first settler): PH-D: pH; SS-D: Suspended Solids (mg/l); SSV-D: Volatile suspended solids (mg/l); COD-D: Chemical organic matter (mg/l); BOD-D: Biodegradable organic matter (m/l). *(iii)* Biological treatment (measures taken in the biological reactor): Q-B: Biological reactor-flow; V30: Index at the biological reactor (measures the sedimentation quality of the mixed liquor, in ml/l); MLSS-B: Mixed liquor suspended solids at the biological reactor; MLVSS-B: Mixed liquor volatile suspended solids: MCRT-B: Mean cell residence time at the biological reactor. *(iv)* Output (when the water is meeting the river): PH-S: pH ; SS-S: Suspended Solids (mg/l); SSV-S: Volatile suspended solids (mg/l); COD-S: Chemical organic matter (mg/l); BOD-S: Biodegradable organic matter (m/l). *(v)* Other variables: QR-G: Recirculated Flow ; QP-G: Purged flow; QA-G: Air inflow

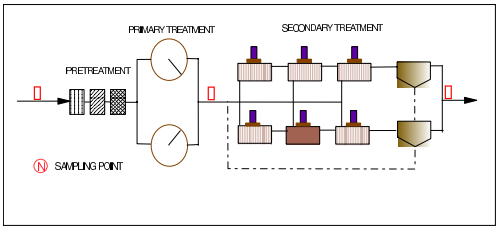

Figure 1: Wastewater treatment plant scheme.

Figure 1 shows where all the measures are taken along the plant. This set is heterogeneous and usually there are missing values and the sensors may provide noisy data.

5.1 Experimental results

Using the software GESCONDA we have performed 60 executions of KMeans over WWTP data set. Every execution provided a classification of the 396 days in 4 classes (from previous works [5],[3] it is known that this is the right number of classes), namely $C_1, \ldots C_{60}$.

Pseudobagging of C_1, C_2, C_3, for example, would consist of selecting the reference partition, crossing the remaining two partitions versus the reference one to perform relabelling and assigning to every day the most frequent class (under new labels).

In this work, pseudobagging was applied to the sequences: $C_1 \ldots C_j, \forall j = 2 : 60$ providing 59 new classifications, each of them obtained with pseudobagging a longer sequence of single classifications. The criteria used in this experiment to select the reference partition was taking the first run, which in fact is the criteria behaving the worst among the three implemented in GESCONDA. This allows a worse case analysis, in fact, and we could suppose that better selection of reference partition (either using I or M criteria) would provide even better results.

For each of the 59 pseudobagged new classifications, the two proposed measures I and M are computed, in order to evaluate the separability among the proposed classes. Figures 2 and 3 show the values of both coefficients either for the single classifications provided by Kmeans and for the pseudobagged new classifications obtained with increasing sequences of single Kmeans classifications.

Since this data only contains numerical variables, the M criteria is evaluated after a discretization of the variables.

The first thing to remark is that I coefficient has higher variability than M, as shown in fig. 2 and 3 with the curves associated to the 60 single classfications provided by Kmeans. Secondly, the measures I and M are not measuring the same aspect of the goodness of the classification and, as it can be seen, best values for I do not coincide with best values for

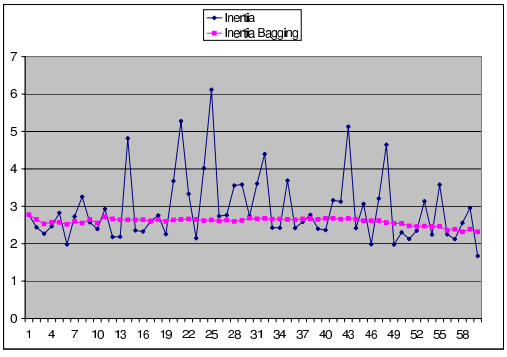

Figure 2: Values of I for single kmeans classification (60 iterations) and for 59 pseudobagged increasing sequences of single kmeans classifications.

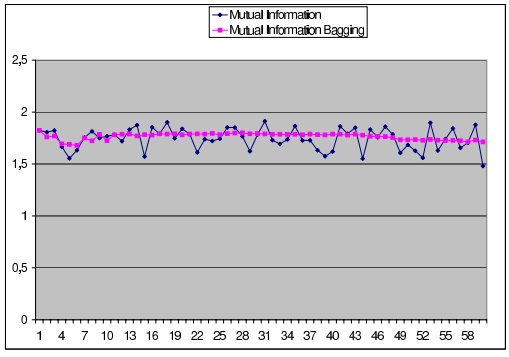

Figure 3: Values of M for single kmeans classification (60 iterations) and for 59 pseudobagged increasing sequences of single kmeans classifications.

M (correlation(I,M)=0,353, which is quite low, indicating that high values of I are not necessarily associated with high values of M). In fact, I increases for more separable classes, while M increases if the information provided by the variables to understand the class variable is greater, which is not exactly the same. The best partition would probably be the one with greater pair (I,M).

Superimposed to the values of single Kmeans run, the curve with the values of I and M for the pseudobagging results are displayed in both figures. Fist thing to remark is that variability when using pseudobagging is much lower. So, first conclusion is that quality of classification becomes quickly stable when using pseudobagging, either using I or M. Also, for this particular application both series converge to an asymptote after the tenth iteration, what means that performing ten runs of Kmeans should be enough to get a pseudobagging classification. This means that bagging provides a good compromise among the single runs, and results would not signifficantly improve by adding more iterations of kmeans for performing the pseudobagging.

6 Conclusions and Future work

In this paper it is shown that bagging techniques can be easily adapted to face non supervised class discovery problems, contributing to mitigate the non deterministic behavior of the random initialization of kmeans.

One single execution of KMeans can produce very good classifications as well as very bad ones, depending on the random initial seeds. From a structural point of view, two measures were introduced to evaluate the goodness of the clustering with different criteria. One, based on inertias and separability of classes, the other based on Shannon's quantity of information.

This work empirically proves that application of pseudobagging techniques converges to a stable quality of the final classification; that assymptote is arround ten iterations and with more classifications, both inertia and mutual information show a constant behavior, what is a desirable situation.

On the other side a very particular implementation of pseudobagging has been used in this work, where the initial classification is picked for building the correspondences in the relabelling step, and a simple voting method is used to decide the class of an instance. In the future, more sophisticated methods for final assignment of class are going to be explored, as well as other algorithms that allow us to make the correspondences between classes.

It is also interesting to take into account that the pseudobagging is based in comparisons with a reference partition that, on the contrary of bagging, is not certain and propagation of uncertainty to the quality of final results needs a depth analysis.

Acknowledgements: This research has been partially fi-

nanced by the project TIC-2004-01368.

References

[1] E. Bauer and R. Kohavi. An empirical comparision of voting classification algorithm: Bagging, boosting and variants. *Machine Learning*, 36(1/2):105–139, 1999.

[2] P. Bühlmann. Bagging, subagging and bragging for improving some prediction algorithm. *Recent Advances and Trends in Nonparametric Statistics*, 2003.

[3] J. Comas. Knowledge discovery by means of inductive methods in wastewater treatment plant data. *AI Communications*, 14(1):45–62, 2001.

[4] S. Duboit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinormatics*, 19(9):1090–1099, 2003.

[5] K. Gibert, M. Sánchez-Marrè, and X. Flores. Cluster discovery in environmental databases using gesconda: The added value of comparisons. *AI Communications*, 18(4):319–331, 2005.

[6] K. Gibert and M. Sánchez-Marrè. Gesconda: An intelligent data analysis system for knowledge discovery and management in environmental databases. *Environmental Modelling and Software*, 21:(in press).

[7] J.M. Gimeno, Béjar J., and U. M. Sànchez-Marrè, Cortés. Discovering and modelling process change: An application to industrial processes. In *Practical Applications of Data Mining and Knowledge Discovery*.

[8] D.J. Hand. *Principles of Data Mining*. MIT Press, Cambridge, Massachusetts, 2001.

[9] Metcalf and Eddy. *Wastewater engineering treatament. Disposal and reuse.* McGraw-Hill, 2003. 4th Ed. revised by George Tchobanoglous, Franklin L. Burton NY.US.