K-means clustering on a classifier-induced representation space : application to customer contact personalization

Vincent Lemaire, Fabrice Clérot, Nicolas Creff

Orange Labs, 2 avenue P. Marzin, 22300 Lannion
vincent.lemaire@orange.com

Abstract. When the marketing service has to contact customers to propose them a product, the probability that these customers will buy this product is calculated beforehand. This probability is calculated using a predictive model. The marketing service contacts the clients having the highest probability of buying the product. In parallel and before the commercial contact it may be interesting to realize a typology of the customers who will be contacted. The idea is to propose differentiated campaigns by group of customers. This article shows how it is possible to build such a typology so that it respects the nearness of the customers with respect to their appetency score.

1 Introduction

1.1 Industrial Problem

Data mining consists in methods and techniques which allow the extraction of information and knowledge from data. Its use makes it possible to establish correlations between data and, for example within the framework of customer relationship management, to define types of customer's behavior.

One common task is to find the relationships or correlations between a set of input or explanatory variables and one target variable. This knowledge extraction is often based on the building of a model which represents these relationships. Faced with a classification problem, a probabilist model (B) estimates the probabilities of occurrence of each target class for all instances of the database given the values of the explanatory variables. These probabilities, or scores, are used for example in customer relationship management to evaluate the probability that a customer will buy a new product (appetency).

The scores are then exploited by marketing services to personalize the customer relationship. Customers are sorted out according to the value of their score, and only the most appetent customers (named "top scores"), i.e. those having the strongest probability to buy the product, are contacted.

In parallel or before the commercial contact, it can be interesting to construct a typology of the customers who will be contacted. This typology is often constructed using a clustering method (G). The idea is to propose marketing campaigns differentiated by customer segments. A sales leaflet is built for every group of customers after analysis of the characteristics of the group: age, CSP, detained offers. For practical reasons (time constraints) the analysis of the group generally amounts to the analysis of the center (or representative customer) of the group. It is important note that this clustering is supposed to have a long lifetime, comparable to the marketing strategy time-scales, and that the same clustering will be re-used for successive marketing campaigns.

Marketing services will then use, for each "top score customer", two pieces of information: the score given by the probabilist model (B) and the characteristics of this customer given by a partitioning method (G). But since there is no link between B and G two problems are generally observed (on Orange campaigns):

1. there is no link, no proximity, between the scores of customers belonging to the same cluster: a cluster can contain customers with a high appetency and customers with a low appetency. The analysis of the center of the group returns an erroneous sales leaflet (as seen above, building a new clustering on the "top scores" after every scoring step is not a viable option).
2. the created clusters are not stable in time when the classifier is deployed successively during several months on the same campaign perimeter (See both criteria section 4.3).

So to resolve the aforementioned problems this article proposes to construct a typology by means of a partitioning method taking into account the knowledge stemming from the classifier which calculates the scores. The purpose is to elaborate a clustering method which preserves the nearness of customers having the same scores.

The second section of this article describes the process which led to choose the algorithm of the k-means as the clustering algorithm. Provided with the choice of the algorithm, section 3 details how to use a classifier-dependent metric, which depends on the classifier used to calculate the scores, during the clusters calculation. Section 4 will present the results obtained before concluding with the last section.

2 Choice of a technique among the various methods of clustering based on partitioning

Clustering is the process of partitioning a database in groups called clusters. The purpose of clustering is to find groups of similar elements in the sense of a similarity measure. There are thus two main elements to be chosen: the method of groups creation and the metric used during the groups creation.

Notations which will be used below in this paper are:

- a training database, $\mathcal{D}$, containing N instances, M explanatory variables and one target variable which has J modalities (the classes to be predicted are noted C_j) ;
- every data instance, D, is a vector of numerical or categorical values $D = (D_1, D_2, ..., D_M)$;
- k is used to designate the desired number of groups.

2.1 Introduction

There are four principal partitioning method which can be used to cluster the elements of a database: a gravity center (the empirical average): the k-means [1]; a geometrical median: the k-medians [2]; a center containing the most frequent modes: the k-modes [3]; a medoid (medoids are representative objects of a data set or a cluster with a data set whose average dissimilarity to all the objects in the cluster is minimal): the k-medoids [4].

The choice of one of these algorithms depends on: (i) the nature of the data to which it must be applied; (ii) the desired result (mean, medoid ...); (iii) the available time and therefore the complexity of the algorithm.

In addition, each of these algorithms depends on the initial selected "center", the value of k, the criterion used to evaluate the quality of the partitioning (cohesion of obtained clusters), the similarity measure and the data representation used at the input of the algorithm.

These points are discussed below in the industrial context of the study.

2.2 Influence of the nature of the initial data

In this study we are in a specific industrial context. Data are from the Orange information system. The explanatory variables which are placed at the input of the classifier (B) used to calculate the appetency probabilities are numerical or categorical variables (with a large number of modalities) and there are missing values. The reader can find a description of these data in [5]. This kind of data representation orients the choice of the partitioning technique towards the technique of the k-prototypes [6] which is a mix of the k-means and k-modes methods. However, the data may also contain a certain number of atypical customers (or erroneous data) which in this case would lead to the choice of k-medoid method that is inherently less sensitive to outliers.

2.3 Influence of the desired result

The result of partitioning should allow marketers to build a sales pitch by cluster. A sales pitch is a structured set of arguments that has the characteristics of a product / service as benefits to the customer. It requires detailed knowledge of the product (characteristics), but also of the needs and motivations of the customer. Therefore one would like the "center" of the clusters to represent a "real" customer and not an average customer. It is difficult to extract knowledge from, for example, the average of two genders, several terminal and tariff plans. This desideratum tipped the choice of the partitioning method in favor of the k-medoid method.

2.4 Influence of the metric

A number of factors must be taken into account when choosing the metric. On the one hand the form of clusters obtained depends on the metric used. On the other hand each of the algorithms described above is dedicated to minimize a particular metric: k-means the L2 norm, k-median the L1 norm [7]... Although clustering algorithms based on partitioning work with almost any type of distance function (or similarity measure) the same guarantees are not obtained considering the metric used. For example the Huygens theorem which shows that the sum of intraclusters inertia and interclusters inertia is constant is valid only if one uses the Euclidean distance. In our case we want to adapt the metric to the one which is naturally induced by the classifier (B) used to calculate the appetency probabilities. This adaptation is described in section 3 below. At this point of the article and for understanding the rest of this section, we just indicate that a weighted L1 norm will be used.

2.5 Influence of the algorithmic complexity

The algorithmic complexities of the different partitioning methods vary greatly depending on the partitioning method itself but also on the implementation. Readers can find in [8] different implementations of k-median, in [4] different implementations of the k-medoids (PAM (Partitioning Around Medoids), CLARA (Clustering Large Applications) and CLARANS (Clustering Large Applications based upon RANdomized Search)). From lowest to highest complexity the algorithms are the k-means, k-mode, k-medoids and finally the k-median.

The marketing campaign involved in this study use databases containing hundreds of thousands of customers, each potentially described by several (tens of) thousands of explanatory variables. After training the classifier (B, which performs a step of variable selection) and retaining only customers with the highest probabilities, databases of tens of thousands of customers described by several hundred variables are obtained. These are databases that are used to build the partitioning. Therefore some of the classical algorithms mentioned above are difficult to use because of the volumetry.

2.6 Influence of the pretreatment

The classifier used by Orange (in the framework of this study) to calculate the appetency probabilities is Khiops$^{\text{TM}}$(within the PAC platform [9]). Khiops[1] incorporates a Naive Bayes classifier [10] after an optimal pretreatment step on the explanatory variables. Khiops discretizes numeric variables and construct modalities groupings for categorical variables. At the end of the pretreatment process numeric and categorical variables are recoded: each attribute m is recoded in a qualitative attribute values containing I_m recodings. Each instance of data is then recoded as a vector of discrete modalities : $D = D_{1i_1}, D_{2i_2}, ..., D_{Mi_M}$. D_{mi_m} represents the recoding value of D_m on the m attribute, with the discrete mode index i_m. After application of the Naive Bayes classifier, the initial explanatory variables are all represented in numerical form as a vector of $M * J$ components: $P(D_{mi_m}|C_j)$.

This pretreatment eliminates the choice of an algorithm like the k-modes, since all variables after the pretreatment step are numeric. It also reduces the advantage of the k-medians / k-medoid regarding the “outliers” because after this type of pretreatment not outliers in terms of a single variable value are present in the data (outliers in terms of variable combinations can still exist).

[1] www.khiops.com

2.7 Influence of missing values

In our case the pretreatment step using Khiops eliminates the missing values. Before the discretization and the grouping of modalities, the missing values for numerical attributes are replace by the values $-\infty$ and those for the categorical attributes are considered as a supplementary value. Then Khiops discretizes numeric variables and construct modalities groupings for categorical variables. Then the K-means algorithm described below is applied on data without missing values.

2.8 Discussion

The above discussion shows the constraints which affect the choice of a the partitioning algorithm most adapted to our industrial context. For example, the computational complexity and nature of the preprocessing performed makes the k-means algorithm very suited to our problem but makes the algorithm less suitable because of the use of a L1 norm and the desire to have real customers as cluster centers.

The k-median algorithm seems more appropriate to the metric used and the nature of the data after preprocessing but its computational complexity makes it unsuitable for our data.

The k-medoid algorithm also seems very appropriate but its complexity remains too high (several hours of computing for small databases data even with optimized algorithms such as CLARANS). Other algorithms [11] slightly modify the algorithm of k-medoid to make it closer to the k-means in terms of complexity but need to store the matrix of distances between customers.

Finally the approach taken in this study is to use the k-median algorithm by taking an approximation of the median as a prototype under the assumption of independent variables and adding a final step after convergence. The assumption of independent variables allows the use of the "component-wise median" [12], a fast version of the median calculation. The step performed after the convergence of the algorithm consists in replacing each prototype by the "real" customer (from this cluster) that is closest to the prototype. The proximity between the customer and the true prototype of the cluster is calculated using a distance L1 norm. This step may slightly degrade the results of the partitioning but it can reach all the objectives given in section 1.1 above.

3 K-means based on Classifier-induced representation space

3.1 Introduction

This section shows that it is possible to insert knowledge coming from the classifier (B) in the metric to be used for the elaboration of a k-means. In our case (the Khiops software) the classifier is obtained from the Averaging of Selective Naive Bayes Classifiers. The purpose is to build a new representation called "supervised representation" (or "classifier-induced representation") so that two instances close in this supervised representation according to the L1 metric should have similar scores (similar appetency probabilities).

The following section describes this supervised representation space for the naive Bayes classifier. Section 3.3 presents how weights are associated with the explanatory variables and how these weights modify the distance.

3.2 Distance depending on the target class

From the naive Bayes predictor and using the log formulation, one has for each target class:

$$log(p(C_j|D)) = \sum_{m=1}^{M} log\left(p(D_{mi_m}|C_j)\right) + log(p(C_j)) - log(p(D)) \quad (1)$$
$$\text{with } D = (D_m)_{m=1,\dots,M} \text{ an instance}$$

The Bayesian decision corresponds to the target class C_j maximizing the above formula. We define the distance between two instances, d^1_{NB} as follows:

$$d^1_{NB}(D, D') = \sum_{m=1}^{M} \sum_{j=1}^{J} \left| log\left(p(D_{mi_m}|C_j)\right) - log\left((p(D^{'}_{mi_m}|C_j)\right) \right| \quad (2)$$

Each instance can then be encoded in a new representation space as a vector of $M * J$ components, as shown in equation 3 for $J = 2$:

$$\begin{array}{c}(log(p(D_{i1_1}|C_1)), log(p(D_{i1_1}|C_2)), \dots, \\ \dots, log(p(D_{Mi_M}|C_1)), log(p(D_{Mi_M}|C_2)))\end{array} \quad (3)$$

The proposed distance is the L1 norm for this classifier-induced representation. Two instances close in the sense of this representation will be close in the sense of their behavior for the class to predict. Indeed if we define the distance between the predicted class distributions as follows:

$$\Delta^1(D, D') = \sum_{j=1}^{J} |log(p(C_j|D)) - log(p(C_j|D'))| \quad (4)$$

and use the following majorization:

$$\Delta^1(D, D') \leq \left[d^1_{NB}(D, D') + J\,|log(p(D)) - log(p(D'))|\right] \quad (5)$$

two instances of the same overall probability close in the sense of d^1_{NB} will be close in the sense of predicting the target class probabilities (two instances with close recoding in the supervised representation will have similar probabilities to have been generated by the recoding model).

3.3 Distance weighting

The building phase of the weights of the variables used by the Naive Bayes classifier is fully described in [13]. It includes two key steps: a step of variable selection (Section 3.5 of [13]) and an averaging step (Section 6.2 of [13]). The variable selection step allows the classifier to avoid unnecessary variables or explanatory variables unrelated to the classification problem. The averaging step allows weighting the variables so that the equation 1 becomes:

$$\begin{aligned} log(p(C_j|D) = & \sum_{m=1}^{M} W_m log\left((p(D_{mi_m}|C_j)\right) \\ & +log(p(C_j)) - log(p(D)) \end{aligned} \tag{6}$$

where W_m is the weight of the variable m whatever is the target class.

Every instance is then recoded on a vector with $M * J$ components but where each component is weighted. The distance (equation 2) is then weighted according to the variables weights and the majorization presented in equation 5 remains true.

3.4 Discussion - Modified k-means algorithm

From here the representation coming from the passage of the initial training database towards a representation where every instance is represented on a vector of $M * J$ components (as shown in the equation 3) is called “supervised representation”; where each variable is weighted with its weight W_m.

The result presented above (equation 5) provides the guarantee that if the k-means algorithm is used on the supervised representation with the L1 norm, we obtain clusters where two individuals close in the sense of the distance, d_{NB}^1, will be close in the sense of their probability to belong to the target class.

The modified k-means algorithm proposed in this article is called “modified” because it uses (i) a supervised representation of the data, (ii) the L1 norm, (iii) an approximation of the median, (iv) a step of post-processing to select real customers as centers. These four changes are expected to achieve the original objectives of the study as presented in the introduction to this article.

This algorithm assumes that the training data and test data have not different distributions. If this assumption is not relevant the reader may be interested by the following references :[14], [15], [16].

4 Experimental results

4.1 Preamble

Initialization: Most of the initialization methods mentioned in [17] have been tested. In our case (supervised representation and L1 norm) no significant difference has been found between the results. Results presented below are obtained using a random initialization of the prototypes.

Cross validation: In each of the experimental phases (and for all values of k) databases were split into 10 bags to achieve a cross-validation. The results present in tables and figures are the mean test AUC (Area Under ROC Curve). The score of membership to the target class of an example is defined as the proportion of elements of the target class of the cluster of this example. In case where the number of target classes is greater than 2 the test AUC expectancy is given.

4.2 First experimental phase

A first experimental phase was conducted in order to (i) measure the impact of supervised representation on the k-means algorithm and (ii) measure the difference between the results obtained from the k-medoid algorithm PAM (that works directly on the "true" customers) and the step of post-designation included in the modified k-means algorithm.

Khiops software was tested using (i) native data and (ii) data preprocessed to obtain their supervised representation. The tested values of k are in the range of 1 to $\sqrt{N}$ for instance $k \in \mathcal{A} = \{1, 2, ..., 9, 10, 20, 40, 80, ..., \sqrt{N}\}$. To compare the results with those obtained using PAM the volumetry was limited by using "small" databases from the UCI [18]. The sum of the squares errors (SSE) has not been used to evaluate the results because it is inappropriate here as two different representations (native and supervised) have been tested. The test AUC was then chosen because it gives an indication of purity of the clusters in the sense of a target class.

Table 1 compares the results obtained with (i) the supervised representation to the results obtained with (ii) the native representation for (a) PAM and (b) the modified algorithm on the databases Iris and Phoneme.

Table 1. Phase 1 - AUC : Mean test results (the suffix '-s' indicates the use of the supervised representation)

	Iris	Phoneme	Shuttle	Letter
PAM	0.959	0.926	-	-
PAM-s	0.951	0.935	-	-
K-means	0.946	0.910	0.902	0.711
K-means-s	0.966	0.919	0.929	0.787
J	2	5	7	26
N	150	2554	58000	20000
M	4	256	9	16

For the databases Letters and Shuttle PAM did not provide a result in an acceptable time (for the different tested values of k and the 10-fold cross-validation) therefore only results for k-means are presented. This table present a mean test results calculated using individual values obtained as a function of k and the 10-fold (f) cross validation ($AUC = \frac{1}{|\mathcal{A}|10} \sum_{k \in A} \sum_{f=1}^{10} AUC(k, f)$). This mean result corresponds to the area under the Learning Curve which has been recently used as test measure in challenges

[19]. Only several representative results (increasing in size of (J, N, M)) of the tests are presented in this paper but the interested reader can find more details in [20]. This table shows that the use of a supervised representation exhibits good behavior and gives interesting results.

Figures 1 et 2 show the obtained results on the dataset Abalone ($N = 4177, J = 28$) and Titactoe ($N = 958, J = 2$) using only the supervised representation. In these two figures the red (+), blue (●) and black (■) curves represent respectively the classification results obtained for PAM clustering and the modified k-means clustering proposed above (both acting on the supervised representation) and the Averaging of Selective Naive Bayes Classifiers (SNB) for the Khiops software (acting on the native representation).

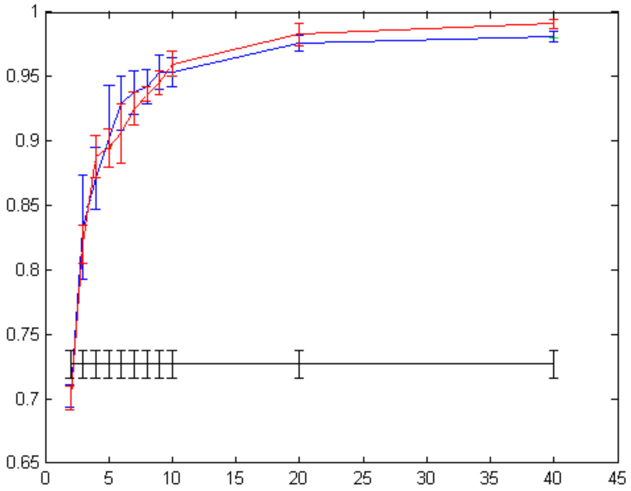


Fig. 1. Abalone: Test AUC versus k

These illustrative results and those presented in [20], show that the modified k-means algorithm using supervised representation induced by the naive Bayes classifier Khiops is very competitive. We also observe that, for high values of k, it can even achieve a better performance classification than the SNB.

4.3 Second experimental phase

Several databases have been available to us for this phase. Three bases of 200,000 customers from March, May and August 2009 for a churn problem for an Orange product were used. These databases contained around 1000 variables. The database of March was used to construct the classifier (B). The March top scores were used to construct

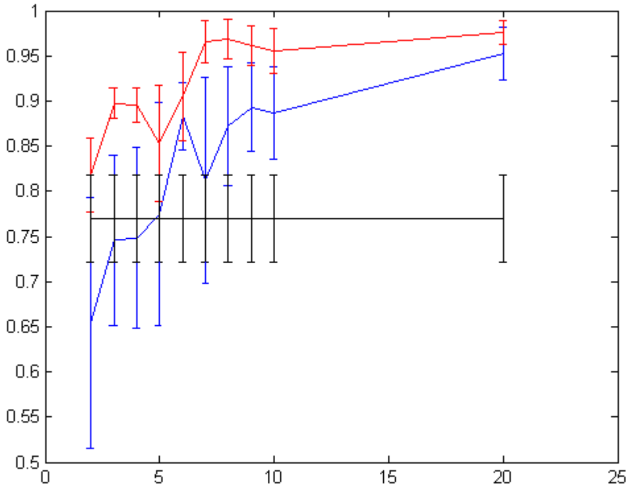


Fig. 2. Titactoe: Test AUC versus k

the partition in k groups using the modified k-means algorithm. The databases of May and August correspond to the test sets. The evaluation criteria were calculated for each month (March, May and August).

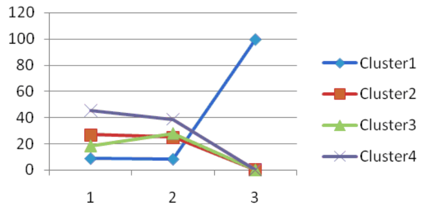
Usually the value of k is fixed using a cross validation process. In that case since we are interested by supervised criterion, the criteria describe in [21] will be appropriate. But in our industrial context, users of the clustering algorithm want to set the value of k according to their own requirements and expertise. After consultation with the concerned Orange entity, three k values were tested: 4, 10, and 20. For space reasons only the results with k = 4 are presented below; the conclusions remain valid for $k = 10$ or $k = 20$ (the complete results are available in [20]).

At the time of the tests, a commercial software solution could be used within the company to achieve this type of campaign. But it was rarely used because the groups obtained were too different from month to month. The modified k-means algorithm proposed in this paper was therefore evaluated using a criterion of stability along two dimensions :

- The first dimension is the percentage of customers in each cluster. For each month T, the percentage of customers in each customer is measured. The operation is repeated the following months using new customers (without a new elaboration of the clustering). On a monthly basis the proportions of customers belonging to a cluster should remain the same so that we can consider the solution as stable according to this criterion.
- The second dimension is the evolution of the distribution of the target class within the clusters. For each the month T, the percentage of the target class is measured for each cluster. The operation is repeated the following months using new customers.

If the allocation of customers remains the same from month to month then we can consider the clustering method to be stable over time.

The results on these two dimensions are presented Figures 3 to 6. The x-axis represents the months (T = 1 = March, T = 2 = May, T = 3 = August) and the y-axis percentages. In Figures 3 and 4 the percentages sum to 100% and correspond to the top scores. On the other hand in Figures 5 and 6 the percentages do not sum to 1 because they represent the proportion of customers in each cluster with the label churn = 1.



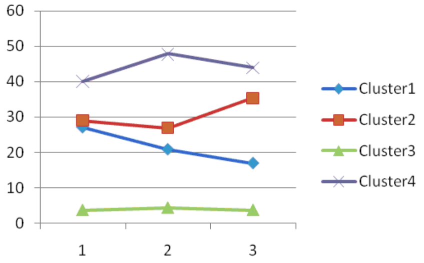
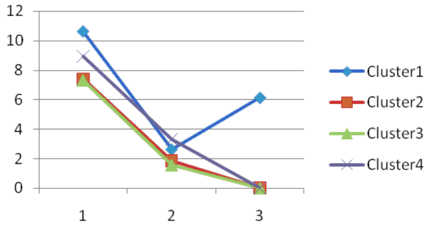


Fig. 3. Percentage of customers per cluster with the current software

Fig. 4. Percentage of customers per cluster with the proposed algorithm



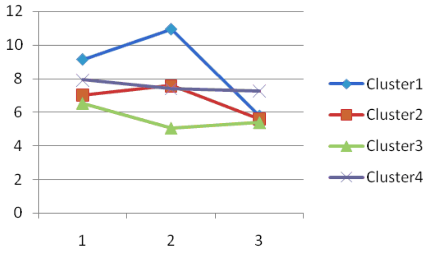


Fig. 5. Percentage of customers (churn=1) with the current software

Fig. 6. Percentage of customers (churn=1) with the proposed algorithm

These four figures show that modified k-means algorithm acting on the supervised representation reaches its goal: the clusters found using supervised representation, which depend on the classifier built in the month T, are much more stable over time (Figures 4 and 6 as compared to Figures 3 and 5). We also know that customers in a cluster have similar churn scores.

4.4 Discussion - A constraint clustering with score proximity

Supervised representation coming from supervised pretreatment (supervised discretization or supervised grouping) allows the use of the result presented equation 5 in the case of the classifier is the naive Bayes. This equation provides a guarantee that if we use the k-means algorithm, using the L1 norm and the supervised representation (equation 3) we obtain clusters where two individuals close in the sense of the supervised representation will be close in the sense of their probability of belonging to the class target.

However equation 5 indicates only $\Delta^1(D, D') \leq d^1_{NB}(D, D')$. So if two instances D and D' are far in the supervised space we have only the guarantee that the distance between their scores will be smaller. The distance between the scores of two distant instances in the supervised representation can be large.

It would be interesting, in the supervised representation, to force the k-means algorithm to cluster only instances that are further away from a threshold value (denoted by ϵ). An algorithm like Xmeans [22] could be used to cut a cluster where the maximum distance between two instances is greater than ϵ. This constraint would give the guarantee to have no cluster with a diameter greater than ϵ. This guarantee could improve the modified k-means algorithm proposed in this paper and automatically set the value of k.

We can also note that the supervised representation built before the clustering could be used with other clustering methods. The Kohonen maps which respect the topology of the space of variables and allow intuitive visualization of the data could be used.

5 Conclusion

This article has shown how to build a typology respecting the knowledge coming from an initial classifier. It was shown that it is possible to elaborate a supervised representation using a naive Bayes classifier. This supervised representation allows a partition that preserves the proximity of samples with the same probability to belong to the target classes. This technique has been used successfully in a customer scoring application. The experimental results show good behavior in terms of measured AUC but also in terms of stability of the typology over time.

The modified k-means algorithm has been operationally deployed and is now used by the Orange business unit which raised the initial problem.

References

1. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability. Volume 1. (1967) 281–297
2. Bradley, P.S., Mangasarian, O.L., Street, W.N.: Clustering via concave minimization. In: Advances in Neural Information Processing Systems -9, MIT Press (1997) 368–374
3. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Discov. **2** (September 1998) 283–304

4. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley (1990)
5. Guyon, I., Lemaire, V., Boullé, M., Dror, G., Vogel, D.: Analysis of the KDD cup 2009: Fast scoring on a large orange customer database. JMLR: Workshop and Conference Proceedings **7** (2009) 1–22 Data available on `http://www.kddcup-orange.com`.
6. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: Pacific Asia Knowledge Discovery and Data Mining Conference, Singapore: World Scientific (1997) 21–34
7. Jajuga, K.: A clustering method based on the l1-norm. Computational Statistics & Data Analysis **5**(4) (1987) 357–371
8. Har-peled, S., Mazumdar, S.: Coresets for k-means and k-median clustering and their applications. In: In Proc. 36th Annu. ACM Sympos. Theory Comput, Chicago, Illinois, USA (2003) 291–300
9. Féraud, R., Boullé, M., Clérot, F., Fessant, F., Lemaire, V.: The orange customer analysis platform. In: Proceedings of the 10th Industrial Conference on Data Mining, Berlin, Germany, Springer Verlag (2010) 584–594
10. Langley, P., Iba, W., Thompson, K.: An analysis of Bayesian classifiers. In: Proceedings of the tenth National Conference on Artificial Intelligence, San Jose, California, USA, MIT Press (1992) 223–228
11. Park, H.S., Jun, C.H.: A simple and fast algorithm for k-medoids clustering. Expert Syst. Appl. **36**(2) (2009) 3336–3341
12. Kashima, H., Hu, J., Ray, B., Singh, M.: K-means clustering of proportional data using l1 distance. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. (dec. 2008) 1 –4
13. Boullé, M.: Compression-based averaging of selective naive Bayes classifiers. Journal of Machine Learning Research **8** (2007) 1659–1685
14. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Self-taught clustering. In: Proceedings of the 25th international conference on Machine learning. (2008) 200–207
15. Zliobaite, I.: Learning under concept drift: an overview. CoRR **abs/1010.4784** (2010)
16. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset Shift in Machine Learning. The MIT Press (2009)
17. Meila, M., Heckerman, D.: An experimental comparison of several clustering and initialization methods. In: Machine Learning. (1998) 386–395
18. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (1998) `http://archive.ics.uci.edu/ml/` las visit : 09/15/2010.
19. Guyon, I., Cawley, G., Dror, G., Lemaire, V.: Results of the Active Learning Challenge. In: JMLR W&CP, Workshop on Active Learning and Experimental Design, collocated with AISTATS, Sardinia, Italy. Volume 10. (2010) 1–26
20. Creff, N.: Clustering l'aide d'une reprsentation supervisée. Master's thesis, Epita, 14-16 rue Voltaire 94276 Kremlin Bictre Cedex (2011)
21. Ferrandiz, S., Boullé, M.: Bayesian instance selection for the nearest neighbor rule. Machine Learning **81**(3) (December 2010) 229–256
22. Pelleg, D., Moore, A.W.: X-means: Extending k-means with efficient estimation of the number of clusters. In: Proceedings of the Seventeenth International Conference on Machine Learning. ICML '00, San Francisco, California, USA, Morgan Kaufmann Publishers Inc. (2000) 727–734