Supervised pre-processings are useful for supervised clustering

O. Alaoui Ismaili[1,2], V. Lemaire[1], and A. Cornuéjols[2]

[1] Orange Labs, AV. Pierre Marzin 22307 Lannion cedex France
(oumaima.alaouiismaili, vincent.lemaire)@orange.com
[2] AgroParisTech 16, rue Claude Bernard 75005 Paris
antoine.cornuejols@agroparistech.fr

Abstract Over the last years, researchers have focused their attention on a new approach, *supervised clustering*, that combines the main characteristics of both traditional clustering and supervised classification tasks. Motivated by the importance of pre-processing approaches in the traditional clustering context, this paper explores to what extent supervised pre-processing steps could help traditional clustering to obtain better performance on supervised clustering tasks. This paper reports experiments which show that indeed standard clustering algorithms are competitive compared to existing supervised clustering algorithms when supervised pre-processing steps are carried out.

1 Introduction

Over the last decade, the world has seen a real explosion of data due mainly to the web, social networks, etc. To exploit these high-dimensional sets of data, clustering and classification algorithms are efficient.

Clustering is an unsupervised learning approach that allows one to discover global structures in the data (i.e. clusters). Given a dataset, it identifies different data subsets which are hopefully meaningful (see Figure 1. a). The discovered clusters are deemed interesting if they are heterogeneous (i.e. their inter-similarity is low) while instances within each cluster share similar features (high intra similarity). This clustering problem has motivated a huge body of work and has resulted in a large number of algorithms (see e.g Jain et al. (1999)). Clustering has thus been used in numerous real-life application domains (e.g. marketing (Berry et al. (1997), CRM (Berson et al. (2000))).

In contrast, classification is a supervised learning approach that consists to learn the link between a set of input variables and an output variable (*target class*). The main goal of this approach is to construct a learning model which is able to predict class membership for new instances (see Figure 1. b).

Recently, researchers have focused their attention on the combination of characteristics of both clustering and classification tasks with the goal to dis-

cover the internal structure of the target classes. This research domain is called *Supervised clustering* (for instance see Al-Harbi et al. (2006) and Eick et al. (2004)). The main idea is to construct or modify clustering algorithms in order to find clusters where instances are very likely to belong to the same class. Formally, *Supervised clustering* is seeks clusters where instances in each cluster share characteristics (homogeneity) and class label. The generated clusters are labeled with the majority class of their instances. Figure 1 illustrates the difference between clustering, classification and supervised clustering.

Generally, clustering tasks require an unsupervised pre-processing step (for example, see Milligan et al. (1988) or Celebi et al. (2013) for the k-means algorithm) in order to yield interesting clusters. For instance, this step might be aimed at preventing features with large ranges from dominating the distance calculations. Now, given the importance of pre-processing for the traditional clustering algorithms, it is natural to ask: could *supervised pre-processing* help standard clustering algorithms to reach good performance in a supervised clustering context? In other words, does a combination of a supervised pre-processing step and a standard clustering algorithm produce a good supervised clustering algorithm, meaning exhibiting high prediction accuracy (supervised criterion) while at the same time uncovering interesting clusters in the data set.

The remainder of this paper is organized as follows. Section 2 briefly describes related work about supervised clustering. Section 3 presents classical unsupervised preprocessing methods and two supervised pre-processing approaches. Section 4 first compares the performance, in terms of prediction accuracy, when using a clustering technique combined with an *unsupervised pre-processing* step and a clustering technique combined with a *supervised pre-processing* step. A comparison between traditional clustering using a supervised pre-processing step with the techniques of supervised clustering algorithms is then carried out. Finally, a conclusion with future work is presented in the last section.

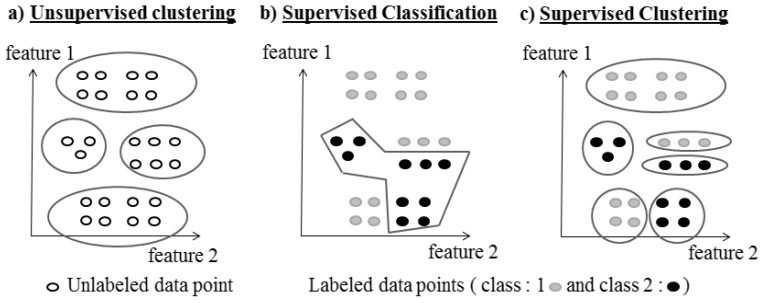

Figure 1. Classification processes

2 Related Work

In the last decade, many researchers focused their attention to build or modify standard clustering algorithms to identify class-uniform clusters where instances within each cluster are homogeneous. Several algorithms are developed to achieve that objective (e.g Aguilar et al. (2001), Sinkkonen et al. (2002), Qu et al. (2004), Finley et al. (2005) and Bungkomkhun (2012)).

In this section, we present two methods proposed by Al-Harbi et al. (2006) and Eick et al. (2004) which modify the K-means algorithm. The experimental results of these algorithms will be compared in Section 4.2 to the results obtained by using a standard K-means algorithm preceded by a supervised pre-processing step.

Al-Harbi et al. (2006) developed a K-means algorithm in such a way to use it as a classifier algorithm. First of all, they replaced the Euclidean metric used in a standard K-means by a weighted Euclidean metric. This modification is carried out in order to be able to estimate the distance between any two instances that have the same class label. The vector of weights is chosen in such a way to maximize the confidence of the partitions generated by the k-means algorithm. This confidence is determined by calculating the percentage of correctly classified objects with respect to the total number of objects in the data set. To solve this problem of optimization, they used Simulated Annealing (a generic probabilistic metaheuristic for the global optimization problem). This iterative process is repeated until an optimal confidence is obtained. In this algorithm, the number of clusters is an input.

Eick et al. (2004) introduced four representative-based algorithms for supervised clustering: *SRIDHCR*, SPAM, TDS and *SCEC*. In their experimentation, they used the first one (i.e *SRIDHCR)*). The greedy algorithm *SRIDHCR* (or Single Representative Insertion/Deletion Steepest Decent Hill Climbing with Randomized Start) is mainly based on three phases. The first one is the initialization of a set of representatives that is randomly selected from the dataset. The second is the primary cluster creation phase, where instances are assigned to the cluster of their closest representative. The third one is the iteration phase where the algorithm is run r times: In each time 'r', the algorithm tries to improve the quality of clustering, for instance, by adding a non-representative instance or by deleting a representative instance. To measure this quality, they use a supervised criterion. It takes into account two points: (i) The impurity of the clustering which defined as a percentage of misclassified observations in the different clusters and (ii) a penalty condition which used in a manner to keep a lowest number of clusters. In this greedy algorithm, the number of clusters is an output.

3 Pre-processing

The following notation is used below:

Let $D = \{(X_i, Y_i)\}_1^N$ denote a training dataset of size N, where $X_i = \{X_{i1}, ..., X_{id}\}$ is a vector of d features and $Y_{i\in\{1,...,N\}} \in \{C_1, ..., C_J\}$ is the target class of size J. Let K denote the number of clusters.

3.1 Unsupervised pre-processing

A pre-processing step is a common requirement for clustering tasks. Several unsupervised pre-processing approaches have been developed depending on the nature of features: continuous or categorical. In this paper, we have used the most common unsupervised pre-processing approach, that is normalization (see e.g. Milligan et al. (1988)).

For continuous features, to the best of our knowledge, data normalization is the most frequently used. It acts to weight the contribution of different features with the aim of making the distance between instances unbiased. Formally, normalization scales each continuous feature into a specific range such that one feature cannot dominate the others. The common data normalization approaches are: *Min-Max*, *statistical* and *rank* normalization.

- *Min-Max Normalization* (NORM): If the minimum and maximum values are given for each continuous feature, it can be then transformed to fit in the range $[0, 1]$ using the following formula: $X'_{iu} = \frac{X_{iu} - min_{i=1,...N} X_{iu}}{max_{i=1,...N} X_{iu} - min_{i=1,...N} X_{iu}}$. Where X_{iu} is the original value of feature u . If minimum and maximum values are equal, then X'_{iu} is set to zero.

- *Statistical Normalization* (SN): This approach transforms data derived from any normal distribution into a standard normal distribution $N(0, 1)$. The formula that allows this transformation is: $X'_{iu} = \frac{X_{iu} - \mu}{\sigma}$ where μ is the mean of the feature u, σ is its standard deviation.

- *Rank Normalization* (RN): The purpose of rank normalization is to rank continuous feature values and then scale the feature into $[0, 1]$. The different steps of this approach are: *i*) Rank feature values u from lowest to highest values and and then divide the resulting vector into H intervals, where H is the number of intervals. *ii*) Assign for each interval a label $r \in \{1, ..., H\}$ in increasing order, *iii*) If X_{iu} belong to the interval r, then $X'_{iu} = \frac{r}{H}$.

For categorical features, among the existing approaches of unsupervised pre-processing, we use in this study the ***Basical Grouping Approach (BGB)***. It aims at transforming feature values into a vector of Boolean values. The different steps of this approach are: *i*) group feature values into g groups with "at best" equal frequencies, where g is a parameter given by the user, *ii*) assign for each group a label $r \in \{1, ..., g\}$, *iii*) use a full disjunctive coding.

3.2 Supervised pre-processing

In this paper, we suggest that one way to help a standard algorithm to reach a good performance in terms of prediction accuracy is to incorporate information given by the target class in a pre-processing step. To prove this, we

proposed two supervised pre-processing approaches called *Conditional Info* and *Binarization*. These approaches are based on two steps: (1) supervised representation and (2) recoding. The first one is a common step for the two approaches. It aims at giving information about variables distribution conditionally to a target class. There are several methods that could achieve the above objective. In this study, we have used the *MODL* (a bayes optimal pre-processing method for continuous and categorical features) approach. It seeks to estimate the univariate conditional density ($P(X|C)$). To obtain this estimation a supervised discretization method is used for continuous features (Boullé (2006)) and a supervised grouping method is used for categorical ones (Boullé (2005)).

To exploit the information given by the first step, a recoding phase is then used as second (common) step. In this paper, we present two ways of recoding (i.e. C.I and BIN). The following methods are compared in Section 4.

• ***Conditional Info (C.I)***: Each feature from the instance X_i is recoded in a qualitative attribute containing I_J recoding values. The resulting vector for this instance is $X_i = X_{i1_1}, \ldots, X_{i1_J}, \ldots, X_{id_1}, \ldots X_{id_J}$. Where $X_{id_1}, \ldots X_{id_J}$ represent the recoding values for the feature d with respect to the number of a class label ($X_{id_J} = log(P(X_{id}|C_J))$. As a result, the initial vector containing d features (continuous and categorical) becomes a vector containing $d \times J$ real components: $log(P(X_{im}|C_j)), j \in \{1,...,J\}, m \in \{1,...,d\}$.

The most remarkable point in this pre-processing process is that if two instances are close in term of distance, they are close also in term of their class membership. A detailed description of this process exists in (Lemaire V. (2012)). Besides, the recoding step provides, for each feature, an amount of information related to the target class. That is by calculating $log(P(X_{im}|C_j))$. This recoding allows one to obtain a new feature space of apriori-fixed size which corresponds to the total number of class labels in the dataset. The similarity between instances is interpreted as a Bayesian distance: $Dist(X_i, X_j) = \sum_{m=1}^{d} \sum_{l=1}^{J} \left[log(P(X_{im_l}|C_l)) - log(P(X_{jm_l}|C_l))\right]^2$.

However, it does not allow keeping the notion of instances: two different instances belonging to different intervals (or groups of modalities) can have equal values of $log(P(X_{im}|C_j))$.

• ***Binarization (BIN)***: In this process, each feature is described on t Boolean features. Where t is a number of intervals or groups of modalities generated by MODL or an other supervised approach. The synthetic feature takes 1 as a value if the real value of the original feature belongs to the corresponding interval or group of modalities and is zero otherwise.

The recoding step of this approach is based on the full disjunctive coding. It transforms each feature into a vector of Boolean features. The size of the vector depends on the number of intervals or groups of modalities associated with each feature. Hence, the size of the new feature space mainly depends on the number of intervals or groups of modalities for all features. Besides, the similarity between instances is determined such that similar instances belong to the same interval or group of modalities.

4 Experimentation

In this section, we present and compare first the average performance of both supervised and unsupervised pre-processing approaches using the k-means algorithm. Then, we compare and discuss the average performance of both supervised pre-processing and other supervised clustering algorithms. These experiments are intended to assess the ability of supervised pre-processing to provide better results than unsupervised pre-processing and also to evaluate the competitiveness of a traditional clustering algorithm (k-means) preceded by a supervised pre-processing step compared to some supervised algorithms in a supervised clustering context.

4.1 Protocol

To test the validity of our assumption, we choose to use the standard K-means algorithm (MacQueen (1967)) which is traditionally viewed as the most popular algorithm in unsupervised clustering. To reduce at best the problem that the K-means algorithm does not guarantee to reach a global minimum: i) the k-means++ algorithm (Arthur et al. (2007)) is used to initialize centers, ii) the algorithm is realized 100 times. At this stage, it is important to define what the best partition is. To be consistent with the definition of *supervised clustering*, we search a criterion that allows us to choose the closest partition to the one given by the target class. In fact, the main aim is to get a compromise between intra similarity and prediction. The intra similarity criterion is guaranteed by the K-means algorithm (trade-off between inertia inter and intra cluster) and the class membership of instances inside each cluster is verified by the chosen criterion; knowing that a supervised / unsupervised pre-processing step is used. For this, we use the Adjusted Rand Index (ARI) (Hubert et al. (1985)) criterion to select the best partition. It is computed by comparing the partition of the target class labels with the partition of the k-means algorithm. For pre-processing approaches, we use those presented above in section 3. Table 1 presents a list of these approaches.

Table 1. The used pre-processing approaches

Unsupervised pre-processing			**Supervised pre-processing**		
Name	Num features	Cat features	Name	Num features	Cat features
RN-BGB	RN	BGB	BIN-BIN	BIN	BIN
CR-BGB	CR	BGB	C.I-C.I	C.I	C.I
NORM-BGB	NORM	BGB			

To evaluate and compare the behavior of different pre-processing approaches in term of their capacity to help traditional clustering in a supervised context, some tests are performed on different databases of the UCI repository (Blake et al. (1998)). Table 2 presents the databases used in this study.

Table 2. Datasets from UCI used in experiment (Var= Variable, Cat= Categorical and Num= Numerical)

Dataset	N	# Var	# Cat	# Num	Dataset	N	# Var	# Cat	# Num
Auto-import	205	26	11	15	Heart-stat-log	270	13	3	10
Breast cancer	699	9	0	9	Iris	150	4	0	4
Contraceptive	1473	9	7	2	Pima	768	8	0	8
Glass	214	10	0	10	Vehicle	846	18	0	18

In order to compare the obtained results with some supervised clustering algorithms, we do: (i) 10×5 fold cross classification (like in Al-Harbi et al.(2006) experiment) for Auto-import, Breast cancer, Contraceptive and Pima datasets. These datasets are also modified in the same way as in Al-Harbi et al. (2006), (ii) 5×10 fold cross classification (like in Eick et al. (2004) experiment) for Glass, Heart-stat-log, Vehicle and Iris datasets.

4.2 Results

Part 1: Comparing supervised and unsupervised pre-processing

Table 3 presents the average performance of the K-means algorithm in term of predictions (Accuracy (ACC) criterion), using each pre-processing approach (section 3.2) for 6 datasets. In this case, the number of clusters is selected following the next procedure. First, the value of K is varied from 1 to 64. Then, For each value of K, a x-fold (see section 4.1) cross validation is performed and the mean value of the ARI is calculated. Finally, the optimal value of K corresponds to the closest partition to the one given by the target class (higher value of ARI versus the value of K in train dataset). Based on this value of K, the ACC is calculated from the corresponding partition in a test dataset. The results in this table show that: (1) supervised pre-processing approaches have most of the time a better performance than unsupervised pre-processing approaches, (2) Binarization (BIN) and Conditional Info (C.I) are close with a small preference for BIN.

In the case where K is given (K is equal to the cardinality of the target class), we obtain also the same result. For example, Figures 2 and 3 present respectively the case where K is an output and where K is an input for Auto-Import and the Glass dataset. This result shows clearly the influence of supervised pre-processing steps (the two first boxplots) on the K-means performance (using the accuracy (ACC) criterion).

Part 2: Comparing supervised pre-processing to other supervised clustering algorithms

We compare the obtained results using the standard k-means algorithm preceded by a supervised pre-processing step (BIN or C.I) to a supervised k means

Table 3. Average performance of k-means algorithm in term of predictions using several pre-processing approaches. (H= Heart, C= Contraceptive, P= Pima, I= Iris, V= Vehicle and B= Breast)

		K	ARI Train	ACC Test
H	**RN-BGB**	**2**	**0.422**	**0.815 ± 0.071**
	SN-BGB	2	0.365	0.796 ± 0.074
	NORM-BGB	3	0.241	0.754 ± 0.077
	BIN-BIN	2	0.452	0.813 ± 0.069
	C.I-C.I	2	0.451	0.807 ± 0.079
I	RN-BGB	3	0.675	0.851 ± 0.087
	SN-BGB	3	0.641	0.833 ± 0.099
	NORM-BGB	3	0.726	0.879 ± 0.080
	BIN-BIN	**3**	**0.872**	**0.929 ± 0.069**
	C.I-C.I	3	0.836	0.899 ± 0.092
C	RN-BGB	2	0.069	0.627 ± 0.025
	SN-BGB	2	0.052	0.604 ± 0.025
	NORM-BGB	3	0.067	0.616 ± 0.030
	BIN-BIN	**3**	**0.093**	**0.630 ± 0.027**
	C.I-C.I	3	0.075	0.621 ± 0.026
V	RN-BGB	7	0.196	0.546 ± 0.036
	SN-BGB	8	0.157	0.507 ± 0.049
	NORM-BGB	8	0.159	0.510 ± 0.044
	BIN-BIN	5	0.256	0.558 ± 0.039
	C.I-C.I	**5**	**0.283**	**0.589 ± 0.033**
P	RN-BGB	2	0.132	0.671 ± 0,038
	SN-BGB	2	0.177	0.705 ± 0.034
	NORM-BGB	5	0.135	0.673 ± 0.041
	BIN-BIN	3	0.148	0.694 ± 0.039
	C.I-C.I	**2**	**0.244**	**0.736 ± 0.034**
B	RN-BGB	2	0.898	0.973 ± 0.012
	SN-BGB	2	0.850	0.959 ± 0.016
	NORM-BGB	2	0.854	0.962 ± 0.015
	BIN-BIN	**2**	**0.904**	**0.974 ± 0.011**
	C.I-C.I	2	0.870	0.961 ± 0.036

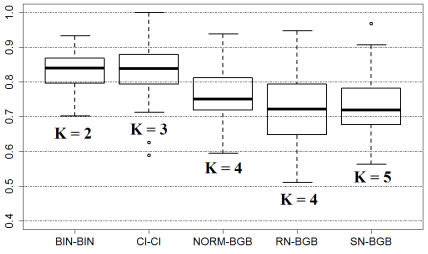

Figure 2. Auto-import: Average performance of k-means (k is an output) using supervised pre-processing (the two first boxplots) and unsupervised pre-processing (the three last boxplots).

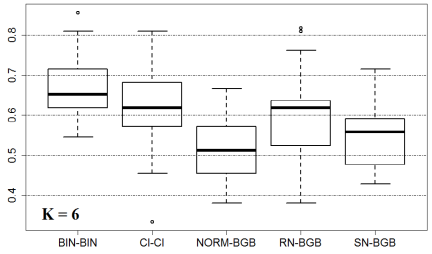

Figure 3. Glass: Average performance of the k-means (k is an input) using supervised pre-processing (the two first boxplots) and unsupervised pre-processing (the three last boxplots).

algorithm proposed by Eick or Al-Harbi. The results for the later algorithms are available in Eick et al. (2004) and Al-Harbi et al. (2006), respectively. Table 4 presents a summary of the average performance of the used methods in term of predictions in the case where K is estimated (Eick) and where K is given (Al-Harbi). The results obtained in the experiments using a standard k-means preceded by a supervised pre-processing are competitive with the mean results of Eick or Al-Harbi (who performed a single x-fold cross validation). We also observe that a standard k-means with a supervised pre-processing

step tends to conserve a lower number of clusters (in Glass dataset, $k = 34, 7$ and 6 for respectively Eick, Binarization and Conditional Info approaches).

Table 4. Comparing with Eick and Al-Harbi algorithms

Comparing with Eick algorithm: (K is an output)						
	Glass dataset		Heart dataset		Iris data set	
	K	ACC Test	K	ACC Test	K	ACC Test
Eick algorithm	34	0.636	2	0.745	3	0.973
K-means with BIN	6	0.664 ± 0.070	2	0.813 ± 0.069	3	0.929 ± 0.068
K-means with C.I	5	0.627 ± 0.080	2	0.808 ± 0.079	3	0.898 ± 0.091
Comparing with Al-Harbi algorithm: (K is an input)						
	Auto-import dataset		Breast dataset		Pima data set	
	K	ACC Test	K	ACC Test	K	ACC Test
Al-Harbi algorithm	2	0.925	2	0.976	2	0.746
K-means with BIN	2	0.830 ± 0.051	2	0.974 ± 0.012	2	0.672 ± 0.041
K-means with C.I	2	0.809 ± 0.102	2	0.961 ± 0.035	2	0.735 ± 0.033

5 Conclusion

This paper has presented the influence of a supervised pre-processing step on the performance of a traditional clustering (especially K-means) in term of predictions. The experimental results showed the competitiveness of a traditional clustering using a supervised pre-processing step comparing to unsupervised preprocessing approaches and other methods of supervised clustering from the literature (especially Eick and Al-Harbi algorithms). Future works will be done (i) to compare supervised pre-processing approaches to others supervised clustering algorithms from the state of the art, (ii) to combine supervised pre-processing presented in this paper with supervised K-means and (iii) to define a better supervised pre-processing approach to combine the advantages of BIN and C.I without their drawbacks.

References

Aguilar-Ruiz, J.S., Ruiz, R., Santos, J.C.R., Girldez, R. (2001): SNN: A supervised clustering algorithm. *In Monostori, L., Vncza, J., Ali, M., eds.: IEA/AIE. Volume 2070 of Lecture Notes in Computer Science., Springer (pp.207–216)*

Al-Harbi, S. H., Rayward-Smith, V. J. (2006): Adapting k-means for supervised clustering. *In: Journal of Applied Intelligence. 24(3) (pp.219-226)*

Arthur, D., Vassilvitskii, S. (2007): K-means++: The advantages of careful seeding. *In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '07 (pp.1027–1035)*

Berry M, Linoff G (1997): Data mining techniques for marketing, sales, and customer support. *John Wiley and Sons, New York*

Berson A, Smith S, Thearling K (2000): Building data mining applications for CRM. *In: McGraw-Hill New York*

Bache, K., Lichman, M. (2013): UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/. *Irvine, CA: University of California, School of Information and Computer Science.*

Boullé, M.(2005): A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research. (pp.1431-1452)*

Boullé, M.(2006): MODL: a Bayes optimal discretization method for continuous attributes. *Journal of Machine Learning 65(1) (pp.131–165)*

Bungkomkhun, P. (2012): Grid-based supervised clustering algorithm using greedy and gradient descent methods to build clusters. *In: National Institute of Development Administration*

Celebi E. M., Hassan A. Kingravi, Patricio A. Vela (2013): A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm. *Journal of Expert Systems with Applications 40(1) (pp.200-210)*

Eick C.F., Zeidat N., Zhao Z.(2004): Supervised clustering algorithms and benefits. *In: Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on, Boca. (pp.774–776)*

Finley, T., Joachims, T. (2005): Supervised clustering with support vector machines. *In: Proceedings of the 22Nd International Conference on Machine Learning. ICML '05, New York, NY, USA, ACM (pp.217–224)*

Hubert, L., Arabie, P. (1985): Comparing partitions. *Journal of classification 2(1) (pp.193–218)*

Jain, A. K., Murty, M. N., Flynn, P. J. (1999): Data Clustering: A Review. *In: ACM computing surveys (CSUR). 31(3) (pp.264-323)*

Jirayusakul, A., Auwatanamongkol, S. (2007): A supervised growing neural gas algorithm for cluster analysis. *In: International Journal of Hybrid Intelligent Systems. 4(2) (pp.129–141)*

Kotsiantis, S. B. (2007): Supervised Machine Learning: A Review of Classification Techniques. *In: IOS Press, Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering.(pp. 3-24)*

Lemaire, V., Clérot, F., Creff, N. (2012): K-means clustering on a classifier-induced representation space : application to customer contact personalization. *Annals of Information Systems, Springer, Special Issue on Real-World Data Mining Applications (pp. 139-153).*

Milligan, G., Cooper, M. (1988): A study of standardization of variables in cluster analysis. *In: Journal of Classification, Springer-Verlag. 5(2) (pp.181-204)*

MacQueen, J.B.(1967): Some methods for classification and analysis of multivariate observations. *In Cam, L.M.L., Neyman, J., eds.: Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. University of California Press (pp.281–297)*

Qu, Y., Xu, S.(2004): Supervised cluster analysis for microarray data based on multivariate gaussian mixture. *In: Journal of Bioinformatics, Oxford Univ Press. 20(12)(pp.1905–1913)*

Sinkkonen, J., Kaski, S., Nikkil, J. (2002): Discriminative clustering: Optimal contingency tables by learning metrics. *In Machine Learning: ECML 2002, Springer. Volume 2430 (pp.418–430)*