Towards an approach using metric learning for interactive semi-supervised clustering of images

Viet Minh Vu
IFI, VNU, Hanoi, Vietnam
Email: vn.vietvu@gmail.com

Hien Phuong Lai
ICTLab, USTH, Hanoi, Vietnam
UMI UMMISCO 209, IRD, Vietnam
FCSE, Thuyloi University
Email: lhienphuong@gmail.com

Muriel Visani
L3I, Université de La Rochelle, France
ICTLab, USTH, Hanoi, Vietnam
Email: muriel.visani@univ-lr.fr

***Abstract*—The problem of unsupervised and semi-supervised clustering is extensively studied in machine learning. In order to involve user in image data clustering, we proposed in [1] a new approach for interactive semi-supervised clustering that translates user feedback (expressed at the level of individual images) into pairwise constraints between groups of images, these groups being formed thanks to the underlying hierarchical clustering solution and user feedback. Recently, the need for appropriate measures of distance or similarity between data led to the emergence of distance metric learning approaches. In this paper[1], we propose a method incorporating metric learning in the existing system to improve performance and reduce the computational time. Our preliminary experiments performed on the Wang dataset show that metric learning methods improve the performances and computational time of the existing system.**

***Index Terms*—interactive semi-supervised clustering, pairwise constraints, metric learning, image analysis.**

I. Introduction

This article deals with content-based image analysis, more specifically with interactive semi-supervised clustering, which can be used *e.g.,* by image navigation or retrieval systems. We proposed an interactive semi-supervised clustering method [1] in order to reduce the semantic gap between high level concepts expressed by the user and the low level features extracted from the original images. In an incremental and interactive context, this method involves the user in the clustering phase, so that he/she can interact with the system to improve the results provided by the automatic semi-supervised clustering model. The system converts supervised information provided by the user into pairwise constraints between groups of images, and proceeds iteratively semi-supervised reclustering by penalizing these constraints. First of all, the system builds an unsupervised clustering model using the BIRCH algorithm proposed in [3] so as to organize input images in a hierarchical clustering structure in which similar images are automatically grouped into compact and representative clusters. Then, unsupervised clustering results are presented to the user via an interactive interface. The user can specify positive and negative images for each cluster. He/she can also drag and drop an image from one cluster to another cluster. In [1], six strategies for deducting pairwise constraints between groups of images based on this user feedback were studied and tested. Taking into account deducted constraints, the system reorganizes the hierarchical data structure using a specifically designed semi-supervised clustering method. This interactive process is repeated iteratively until the user is satisfied.

The measures of similarity/dissimilarity between observations play an important role in human cognitive systems as well as artificial recognition and retrieval systems. The main task of all clustering algorithms is to determine the cluster to which an observation belongs. It means that we need a measure of similarity/dissimilarity between points in the dataset. Euclidean distance is widely used in clustering algorithms. But this geometric distance is not always good, *e.g.* when the data groups are non-spherical or heterogeneous in the data space. Indeed, the Euclidean distance is isotropic, while some directions must be considered in priority in some situations. We therefore need a parameterizable measure. Learning Mahalanobis distance from the distribution of observations in the data space is an interesting solution. The main idea of metric learning algorithms is to learn (sometimes incrementally) a set of parameters which control a particular distance function. This idea is compatible with incremental interactive systems where new supervised information is provided in each interactive iteration in the form of user feedback. The metric is learned based on the new supervised information in order to make the clustering result more satisfying for the user.

In this article, we propose several methods which incorporate Mahalanobis distance learning in the existing interactive semi-supervised clustering system proposed in [1] in order to improve the performance and reduce the computational time. These methods are presented in Section II. The privileged approach in our application context as well as the analysis of experimental results are presented in Section III. Finally, Section IV concludes this article.

II. Scientific context

Automatic machine learning systems (except some deep learning systems) traditionally use low-level features extracted from original images using algorithms for edge detection (Canny, Sobel, Prewith, *etc.* [4]), for corner detection (Harris, SUSAN, *etc.* [5]), for blob detection (Laplacian of Gaussian (LoG) [6], Difference of Gaussians (DoG) [7], Determinant of

[1] A french version of this paper [2] was published in CIFED 2016.

Hessian (DoH) [6], etc.) as well as for local features extraction, such as SIFT [8], SURF [9], color SIFT (rgSIFT, CSIFT, *etc.*) [10], etc.

When a non-expert user observes the results provided automatically based on such features, he/she usually uses higher level semantic concepts for analyzing if two images are similar or different. There is thus a semantic gap between the perception of the user and results provided by the system. The interactive semi-supervised clustering model proposed in [1] and different metric learning methods presented in this section aim at reducing this semantic gap.

A. Interactive semi-supervised clustering (Lai et al., 2014)

This section presents the interactive semi-supervised clustering system we proposed in [1], which involves the user in the clustering phase in order to reduce the semantic gap and improve the unsupervised clustering results. Based on a formal and experimental comparison [11], this system uses rgSIFT [10] features and BIRCH unsupervised clustering. But, it could be used with any set of image features and hierarchical clustering algorithm.

More specifically, rgSIFT is a color SIFT descriptor we use together with the ”Bag of Visual Words” (BoVW) for its high performance. In our system, the number of visual words has been heuristically fixed to 200.

BIRCH unsupervised clustering is used for the initial clustering. A CF-tree is first incrementally built from all the images in the database. Its leaves (CF-entries) are then grouped using K-means. In each interactive iteration, the user visualizes the clustering results and provides feedback through an interactive interface (see next paragraph). After collecting user feedback, re-clustering is performed so as to enhance the clustering results compared to the user's attempts. Our re-clustering is applied using a variant of HMRF-Kmeans [12] performed on the set of CF-entries (leaf nodes) instead of individual images. It thus relies on a constraint deduction engine, feeding the system with pairwise constraints between CF-entries deduced from user feedback with individual images. This process is iterated until the user is satisfied with the clustering solution.

1) Interaction model: The user is involved in each interactive iteration to improve the clustering result. The interaction model is as follows. The interactive interface is shown in Figure 1. In the principal plane (obtained by Principal Component Analysis (PCA) [13]), all clusters are represented by their most representative image called ”prototype” (according to a selected criterion, for example the Silhouette Width measure [1]). By clicking a prototype, the user can view the prototype image, the 10 most representative images and the 10 least representative images which have not yet received a user feedback. The user can then specify positive and negative feedback by clicking on the images that are relevent (respectively irrelevant) for their assigned cluster. He/she can also drag and drop an image from one cluster to another cluster. When an image is moved from the cluster A to the cluster B, it is considered as a negative for cluster A and positive for cluster B. The idea of collecting feedback on the most representative and least representative images of each cluster is, similar to active learning, to maximize the information collected by the system while minimizing the user's effort.

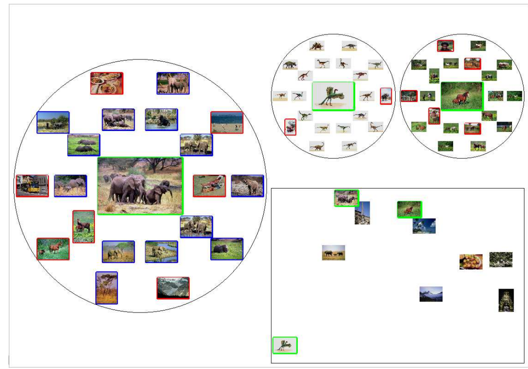

Fig. 1. Interactive interface proposed by [1].

2) Constraint deduction strategies: In each interactive iteration, the system receives user feedback as a list of positive and negative images for each cluster the user interacted with. All positive images should remain in their cluster, while negative images should move to another cluster. As a result, in each cluster, we consider that there are MustLink constraints between each pair of positive images and CannotLink constraints between each pair of negative image and positive image. CannotLink constraints may exist between images of a CF-entry, or there may be simultaneously MustLink and CannotLink constraints between images of two CF-entries CF_i and CF_j. Such CF-entries need to be split into several purer CF-entries. The algorithm for splitting CF-entries is based on intermediate sets of data: seeds. A seed consists of images which belong to the same cluster and are linked by MustLink constraints. We proposed a constraint deduction engine so as to deduce, from the list of pairwise MustLink and CannotLink constraints between images, a list of pairwise MustLink and CannotLink constraints between CF-entries (tree leaves). The idea is to be more efficient when re-clustering the data while incorporating more global feedback information from local interactions with individual images. Then, a variant of the semi-supervised clustering algorithm HMRF-Kmeans is applied using directly pairwise constraints between CF-entries. In this algorithm, constraint violations are not forbidden, they are only penalized. For more details, please refer to [1].

B. Metric learning

The need for appropriate distance/similarity measures is omnipresent in machine learning, pattern recognition and data mining, in particular for classification and clustering problems. But finding a good measure for a specific problem is generally difficult. This need led to the emergence of metric learning, which aims at learning an adapted metric from input data.

1) Mahalanobis distance: The Mahalanobis distance allows to calculate the distance between two data points in a d dimensional space by taking into account the covariance of its d variables. Practically, for a set of data points $\mathcal{X} = \{x_i\}$ drawn from an underlying distribution with mean vector μ and covariance matrix $\Sigma = \frac{1}{|\mathcal{X}|}(\mathcal{X} - \mu)(\mathcal{X} - \mu)^T$, the Mahalanobis distance from a vector x to this distribution is: $D_\Sigma(x, \mu) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}$. The principal objective of most Mahalanobis distance learning methods is to learn the covariance matrix Σ.

2) Different types of metric learning approaches: In many semi-supervised learning methods, supervised information is organized under the form of pairwise constraints: MustLink constraints $\mathcal{M} = \{(x_i, x_j)\}$ (where x_i, x_j must be similar) and CannotLink constraints $\mathcal{C} = \{(x_i, x_j)\}$ (where x_i, x_j must be dissimilar). Mahalanobis distance learning generally aims at estimating the metric parameters that best satisfy these constraints, by solving the following optimization problem:

$$\min_{\Sigma} \mathcal{J}_{obj}(\Sigma, \mathcal{M}, \mathcal{C}) + \lambda \mathcal{R}(\Sigma) \tag{1}$$

where the objective function $\mathcal{J}_{obj}(\Sigma, \mathcal{M}, \mathcal{C})$ is a function of the covariance matrix Σ and of the sets of constraints $\mathcal{M}, \mathcal{C}$ (using a penalty for violating these constraints). $\mathcal{R}(\Sigma)$ is a regularization term which has different form for each particular problem, weighted by a hyper-parameter $\lambda \geq 0$.

There are two kinds of metric learning approaches: global approaches and local approaches.

Global approaches

As mentioned above in the general form of optimization problem, mathematically, metric learning generally focuses on linear measures because they are easier to optimize and less likely to be over-trained. In practice, it is easier to draw convex formulations with the guarantee of finding the global optimum.

MMC (*Mahalanobis Metric Learning for Clustering with Side Information* - [14]) is the first approach learning the Mahalanobis distance, which aims at maximizing the sum of distances between dissimilar points while minimizing the sum of distances between similar points. LMNN (*Large Margin Nearest Neighbors* - [15]) is also one of the most widely used methods. The idea is that, the k-nearest neighbors of a training example should belong to the class of this example ("*target neighbors*"), while the other neighbors should be far from the given example (*"imposteurs"*).

In some cases, there are non-linear structures in the data that linear measures cannot represent. Non-linear methods learn a global distance of the form $D(x_i, x_j) = \| f(x_i) - f(x_j) \|_2$ with a specified function f. In general, learning a non-linear transformation is difficult. Contrary to linear transformation which can be expressed as a matrix of parameters, non-linear transformation is not easily parameterized. In order to learn such transformations, it is necessary to limit the form of non-linear mapping function f to a particular parameterized class, *e.g., kernelized* linear transformations. Several authors have proposed kernelization methods based on the Kernel Principal Component Analysis (KPCA - [16]), a non-linear extension of PCA. In short, KPCA projects implicitly the data into a non-linear space induced by a kernel, then executes a dimensionality reduction in this space. A metric learning algorithm can then be used to learn a metric in this non-linear space. [16] shown that KPCA is theoretically robust for metric learning without constraints. Another useful trick (involving a non-linear preprocessing of the feature space) is based on the kernel density estimation and allows to process both numerical and categorical attributes [17]. Wang et al. [18] proposed an approach using multiple kernel framework for metric learning.

Local approaches

The above methods learn (linearly or non-linearly) global metrics. However, if the data is heterogeneous, only one metric may not represent properly the data complexity. Using several local measures may be more effective, *e.g.,* one metric for each class/cluster. In particular, MPCKMeans (*Metric with Pairwise Constraints KMeans* - [19]) is able to learn individual measures for each cluster (see next section). It has been shown that local approaches significantly outperform global approaches for some problems, but they require more annotated examples, and thus more training time and memory. We chose to integrate MPCKMeans in our existing interactive semi-supervised clustering system because of its effectiveness, and because it has several common points with the HMRF-KMeans algorithm we used in the baseline algorithm (without metric learning): both are based on EM algorithm and use pairwise constraints.

III. Proposed method and experimental evaluation

A. Proposed method

The baseline method in [1] is an adaptation of the HMRF-KMeans algorithm [12], applied to the CF-Entries of the BIRCH tree [3] in an interactive context. The Euclidean distance is used everywhere (in the construction of the CF-tree, in the division of CF-Entries and in the incremental interactive clustering). However, as discussed above, the Euclidean distance has several drawbacks, so we need a new metric which best corresponds to the demand of the user in the interactive context and can accelerate the convergence. In our system, HMRF-KMeans algorithm is replaced by the metric learning algorithm MPCKMeans in the interactive re-clustering step. Our proposed method differs from the baseline method on two points: (1) Replacement of HMRF-KMeans by MPCKMeans, in order to learn one (or several) Mahalanobis distance(s) that better fit our multidimensional data. (2) Use of the Mahalanobis distance in the early steps of construction and division of the BIRCH tree.

The MPCKMeans algorithm adapted to the existing system receives as an input the set of CF-Entries $\mathcal{X} = \{CF_1, ..., CF_m\}$, together with MustLink and CannotLink constraints between pairs of CF-Entries: $\mathcal{M} = \{(CF_i, CF_j)\}$, $\mathcal{C} = \{(CF_i, CF_j)\}$. We also use two constants $\omega, \overline{\omega}$ to weight the penalty cost for violating constraints in $\mathcal{M}$ and $\mathcal{C}$ respectively.

We note $\mathcal{X}_h$ as the set of points of each cluster h. We always have to compute the distance between any two data points (CF_i, CF_j), or the distance between a data point and a cluster h according to the metric of each cluster which is parameterized by the mean μ_h and the covariance matrix Σ_h: $\Sigma_h = \frac{1}{|\mathcal{X}_h|}(\mathcal{X}_h - \mu_h)(\mathcal{X}_h - \mu_h)^T$. The Mahalanobis distance (characterized for each cluster h) is thus calculated based on the inverse of the covariance matrix $A_h = {\Sigma_h}^{-1}$:

$$D^2_{A_h}(CF_i, \mu_h) = (CF_i - \mu_h)^T A_h (CF_i - \mu_h) \quad (2)$$

$$D^2_{A_h}(CF_i, CF_j) = (CF_i - CF_j)^T A_h (CF_i - CF_j) \quad (3)$$

We note l_i the label of each CF-Entry CF_i. The expected results are K disjoint subsets $\{\mathcal{X}_h\}_{h=1}^K$ which are created iteratively by minimizing locally the objective function $\mathcal{J}_{obj}$:

$$\begin{aligned} \mathcal{J}_{obj} \quad = \quad & \sum_{CF_i \in \mathcal{X}} \Big(D^2_{A_{l_i}}(CF_i, \mu_{l_i}) - log(det(A_{l_i})) \Big) \\ & + \sum_{(CF_i, CF_j) \in \mathcal{M}} \omega_{i,j} f_M(CF_i, CF_j) \mathbb{1}[l_i \neq l_j] \\ & + \sum_{(CF_i, CF_j) \in \mathcal{C}} \overline{\omega_{i,j}} f_C(CF_i, CF_j) \mathbb{1}[l_i = l_j] \end{aligned} \quad (4)$$

In equation 4, the first term measures the distortion between a CF-entry CF_i and the center of its cluster μ_{l_i}, followed by the log-determinant of the covariance matrix of the cluster l_i which is a normalization constant of the Gaussian distribution with covariance matrix ${A_{l_i}}^{-1}$ that plays a role of regularization term in the equation (1). The second and third terms represent the penalties for violating pairwise constraints between CF-entries CF_i and CF_j. $\mathbb{1}$ is the indicator function: $\mathbb{1}[true] = 1$ and $\mathbb{1}[false] = 0$. Intuitively, the penalty for violating a MustLink between distant points should be greater than between nearby points (equation (5)), and the penalty for violating a CannotLink between two points that are nearby should be greater than between distant points (equation (6)). In equation (6), CF'_{l_i}, CF''_{l_i} are the two most distant entries in all the dataset according to the metric of the cluster l_i.

$$f_M(CF_i, CF_j) = \frac{1}{2} D^2_{A_{l_i}}(CF_i, CF_j) + \frac{1}{2} D^2_{A_{l_j}}(CF_i, CF_j) \quad (5)$$

$$f_C(CF_i, CF_j) = D^2_{A_{l_i}}(CF'_{l_i}, CF''_{l_i}) - D^2_{A_{l_i}}(CF_i, CF_j) \quad (6)$$

The pseudo-code of the MPCKMeans algorithm adapted to our context is presented in Algorithm 1.

B. Experimental protocol

In order to fairly compare the clustering results of different interactive methods even though the images presented for feedback to the user differ from one method to another, a user agent is implemented to simulate the behavior of the user. The user agent interact with the system under the same conditions a human agent would ("seeing" the same clusters and images). As an oracle, the user agent always gives its feedback following the ground truth containing the class label of each image (previously manually annotated by a human). All the experiments presented in this article are tested with the Wang image database containing 1000 images of 10 different categories. Each experiment is launched 5 times with the same configuration, and the average measures are provided. A PC with *Ubuntu 15.04, 64 bit, CPU Intel 2.4GHz, RAM 4GB, g++ 4.9.2* is used for these experiments.

Using the BoVW model for rg-SIFT with 200 words, we compare the baseline system in ([1]) with different extensions using the metric learning Algorithm 1.

As illustrated in Table I, in our Mahalanobis distance learning framework, we can choose to learn diagonal or full covariance matrices. We can also learn one local metric for

Algorithm 1 Adapted version of MPCKMeans

Input:

$$\begin{aligned} &\text{A set of CF-Entries } \mathcal{X} = \{CF_1, ..., CF_m\}, \\ &\text{A set of MustLinks } \mathcal{M} = \{(CF_i, CF_j)\}, \\ &\text{A set of CannotLinks } \mathcal{C} = \{(CF_i, CF_j)\}. \end{aligned} \quad (7)$$

Output: K disjoint subsets $\{\mathcal{X}_h\}_{h=1}^K$

Initialization (at iteration $t = 0$):

Create λ neighborhoods $\{N_p\}_{p=1}^{\lambda}$ from $\mathcal{M}$ and $\mathcal{C}$, where each neighborhood consists of points connected by Mustlinks.
if $\lambda \geq K$ **then**
  Initialize $\{\mu_h^{(0)}\}_{h=1}^K$ using *Weighted Farthest-First Traversal* algorithm, starting from the largest neighborhood.
else
  Initialize $\{\mu_h^{(0)}\}_{h=1}^{\lambda}$ with the centroids of the created neighborhoods $\{N_p\}_{p=1}^{\lambda}$.
  For the remaining $K - \lambda$ clusters, initialize randomly.
end if

EM loop (for each iteration t):

while not convergence **do**
  $assign\_cluster(E\_Step)$: Assign each data point CF_i to cluster h^*:

$$\begin{aligned} h^* = \arg\min_h \Big(& D^2_{A_h}(CF_i, \mu_h) - log(det(A_h)) \\ & + \sum_{(CF_i, CF_j) \in \mathcal{M}} \omega_{i,j} f_M(CF_i, CF_j) \mathbb{1}[h \neq l_j] \\ & + \sum_{(CF_i, CF_j) \in \mathcal{C}} \overline{\omega_{i,j}} f_C(CF_i, CF_j) \mathbb{1}[h = l_j] \Big) \end{aligned} \quad (8)$$

  $estimate\_means(M\_Step\_A)$:

$$\{\mu_h^{(t+1)}\}_{h=1}^K \leftarrow \{\frac{1}{|\mathcal{X}_h^{(t+1)}|} \sum_{CF_i \in \mathcal{X}_h^{(t+1)}} CF_i\}_{h=1}^K \quad (9)$$

  $update\_metrics(M\_Step\_B)$:

$$\begin{aligned} X\_cov &= \sum_{CF_i \in \mathcal{X}_h} (CF_i - \mu_h)(CF_i - \mu_h)^T \\ ML\_cov &= \sum_{(CF_i, CF_j) \in \mathcal{M}} \frac{1}{2} \omega_{i,j} (CF_i - CF_j)(CF_i - CF_j)^T \mathbb{1}[h \neq l_j] \\ CL\_cov &= \sum_{(CF_i, CF_j) \in \mathcal{C}} \overline{\omega_{i,j}} \Bigg((CF'_h - CF''_h)(CF'_h - CF''_h)^T \\ & \qquad - (CF_i - CF_j)(CF_i - CF_j)^T \Bigg) \mathbb{1}[h = l_j] \\ A_h &= |\mathcal{X}_h| \Big(X\_cov + ML\_cov + CL\_cov \Big)^{-1} \end{aligned} \quad (10)$$

  $t \leftarrow (t + 1)$
end while

each cluster, or only one global metric for the whole dataset. Moreover, we can use the Euclidean or the Mahalanobis distance for the CF-tree construction and the leaves division.

C. Experimental results

Our system is experimented in an interactive context with feedback provided by the user agent (oracle). In each

Global Approaches	Local Approaches
GLOBAL_DIAGONAL(distE)	LOCAL_DIAGONAL(distE)
GLOBAL_FULL(distE)	LOCAL_FULL(distE)
GLOBAL_DIAGONAL(distM)	LOCAL_DIAGONAL(distM)

TABLE I
EXPERIMENTED METRIC LEARNING METHODS USING MPCKMEANS

interactive iteration, the clustering result is presented to the user agent and its feedback is used to modify the clustering model. The re-clustering step is repeated to best satisfy the user's need. We use the same strategies for deducting constraints from user feedback as in [1]; in short, it ranges from strategy 1 where all user constraints and all deduced constraints from all interactive iterations are used to strategy 6, where we filter the user and deduced constraints so as to keep only the MustLink constraints between the most distant images (resp. the CannotLink constraints between the closest images), taking into account the sizes of the neighbourhoods (sets of objects which are "must-linked").

We do not obtain any result for the LOCAL_FULL method because some covariance matrices are not properly conditioned (lack of data in each cluster). We can overcome this problem in the LOCAL_DIAGONAL method by using the regularization term of the diagonal coviariance matrices. With several strategies for deducting constraints, we do not obtain good results with the LOCAL_DIAGONAL method because it converges to a local minima. On the other hand, the computational time of the GLOBAL_FULL method is very high (up to 4 times higher than when using a diagonal covariance matrix). Thus it is not suitable to our applicative context (on-line user interaction). We focus then on the results of global-diagonal methods (see Figures 2 and 4), where the vertical axis represents the performance measured by a conditional Entropy-based external cluster evaluation measure [20], the V-Measure $\in [0.0, 1.0]$ (comparing the clusters with the classes in the ground-truth) and the horizontal axis represents the number of interactive iterations. Computational times are presented in Figure 3.

1) Analysis of the results of adapted MPCKmeans methods: The *GLOBAL_DIAGONAL (distE)* method using the Euclidean distance for the construction and division of the CF-Tree is presented in Figure 2. In that case, the learnt Mahalanobis distance is used only in the objective function $\mathcal{J}_{obj}$ (see equation (4)). We found that the first strategy which deduces all possible constraints produces the best result. But its high computational cost is a great disadvantage. The second and third strategies have acceptable execution times, but they do not produce good results. The fifth and sixth strategies, which use some selected constraints, give better results. In comparison to the fifth strategy, the sixth strategy optimizes the deduction of constraints and reduces computational time but its result is always worse. On the other hand, the fourth strategy (where we use user constraints between images and cluster positive prototypes of all interactive iterations as well as deduced constraints between images and cluster positive prototypes from the current iteration only) gives an interesting result. It takes only about 37 seconds to execute all the 50

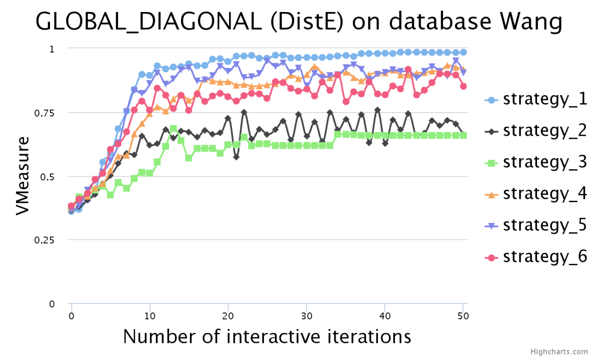

Fig. 2. Metric learning with a global diagonal covariance matrix, using Euclidean distance in the construction and division of CF-Entries

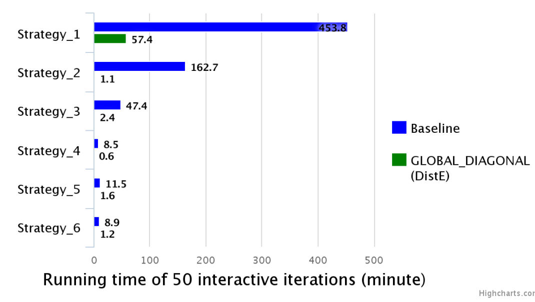

Fig. 3. Computational time of the Baseline and the adapted MPCKMeans GLOBAL_DIAGONAL (DistE) methods

iterations of the interactive re-clustering. The result of this strategy is not very good at the beginning (when the number of constraints is small), but we observe its tendency to increase after each interactive iteration. And its result is comparable to the 5$^{\text{th}}$ strategy after about 30 iterations.

Figure 4 compare the adapted MPCKmeans method using the Euclidean distance (*GLOBAL_DIAGONAL (distE)*) with that using the Mahalanobis distance ((*GLOBAL_DIAGONAL (distM)*)) for the construction and division of the CF-tree. We can see that using the Mahalanobis distance gives better results except in the case of strategy 1 that uses all constraints deducted from all interactive iterations (which is anyway not suitable for our context due to its high computational time).

2) Comparison of the adapted MPCKmeans methods with the Baseline method: In Figure 4, the results of the adapted MPCKmeans methods (GLOBAL_DIAGONAL (DistE) and GLOBAL_DIAGONAL (DistM)) are compared to that of the Baseline method. We can see that metric learning methods outperform the Baseline method for most constraint deduction strategies, at the cost of some instability. This instability, which is smaller with strategy 1, is certainly due to the lower number of constraints given to the system in the other strategies. Additionally, in most of the data-sets, there is always the within-cluster variance: the images in the same cluster may differ. Sometimes when our proposed method deals with this case, it will be heavily penalized. Our parameters $\omega_{i,j}$ for balancing the penalization of violated constraints are also not well tuned, which partially leads to this instability.

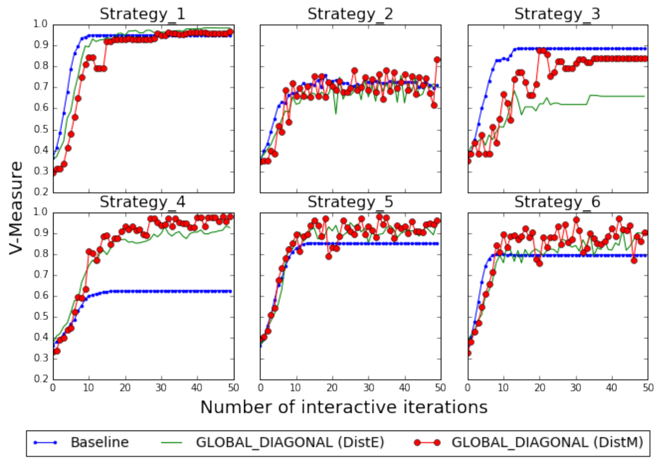

Fig. 4. Performance comparison of our methods with the Baseline method

The computational time of the Baseline and the adapted MPCKmeans GLOBAL_DIAGONAL (DistE)) methods are compared in Figure 3. The results show that the computational time of the metric learning method is much smaller. It can be explained as follows: the time for deducing constraints occupies most of the time in each iteration. In the baseline system, when a CF-entry contains at the same time MustLinks and CannotLinks between its images, it must be divided into serveral coherent CF-Entries, which takes significant time. When applying metric learning methods, the images in the CF-Entries are more coherent because less constraint violations occur between images. Therefore, less CF-Entries are created after each iteration, and thus less pairwise constraints are deduced between CF-Entries. That helps to speed-up considerably the re-clustering step. Moreover, the implementation of the MPCKMeans is optimized by using *Eigen* (http://eigen.tuxfamily.org), a high-level C++ library of template headers for linear algebra, matrix and vector operations.

Thus we obtain a dramatic improvement in computational time; for example with strategy 4, for 50 interactive iterations it decreases from 8,5 minutes to less than 37 seconds.

3) Work in progress: Despite the fact that the metric learning allows to improve the final results, in the first iterations, the Baseline method is still better. Thus, the combination of these two methods is a perspective of our work. For instance, our preliminary results have shown that, when applying the Baseline method in the first iterations for getting a stable clustering solution, then switching to our adapted MPCKmeans, we can accelerate the convergence of this solution towards the user's wishes with improved stability. In addition, experiment our proposed method with the others data-sets is considered in the future work.

IV. Conclusion

In this paper, we present a new interactive semi-supervised clustering approach using metric learning. We are interested in learning the Mahalanobis distance which allows us to modify the weight of each dimension in the representation space (built using BoVW in our context). Specifically, adapted versions of MPCKMeans are used to learn either a global distance, or several local distances. We have shown that, for practical reasons (except when the number of examples in each cluster is sufficiently high), it is preferable to use global distance learning. For computational matters linked to our online interactive learning context, we prefer to learn the Mahalanobis distance with a diagonal covariance matrix than a full matrix. Our experiments, applied in the Wang image database, show that metric learning allows to improve the clustering performance, but at the cost of some instability. We are currently working on combining this new approach with the existing approach in order to reduce the instability while keeping the performance improvement. Another very important result of our study is that metric learning allows to reduce dramatically the execution time, and thus is better suited to interactive applications.

References

[1] H. P. Lai, M. Visani, A. Boucher, and J.-M. Ogier, "A new interactive semi-supervised clustering model for large image database indexing," *Pattern Recognition Letters*, vol. 37, pp. 94–106, 2014.

[2] V. M. Vu, H. P. Lai, and M. Visani, "Vers une approche utilisant l'apprentissage de métrique pour du clustering semi-supervisé interactif d'images," in *CIFED 2016 Colloque International Francophone sur l'Ecrit et le Document.*

[3] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," in *ACM SIGMOD Record*, vol. 25, no. 2. ACM, 1996, pp. 103–114.

[4] S. E. Umbaugh, *Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools, Second Edition*, 2nd ed. Boca Raton, FL, USA: CRC Press, Inc., 2010.

[5] A. R. Willis and Y. Sui, "An algebraic model for fast corner detection." in *ICCV*. IEEE Computer Society, 2009, pp. 2296–2302.

[6] T. Lindeberg, "Scale selection properties of generalized scale-space interest point detectors," *Journal of Mathematical Imaging and Vision*, vol. 46, no. 2, pp. 177–210, 2013.

[7] G. H. Ball and D. J. Hall, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–1157.

[8] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[10] K. van de Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, Sep. 2010.

[11] H. P. Lai, M. Visani, A. Boucher, and J.-M. Ogier, "An experimental comparison of clustering methods for content-based indexing of large image databases," *Pattern Analysis and Applications*, vol. 15, no. 4, pp. 345–366, 2012.

[12] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 59–68.

[13] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[14] E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in neural information processing systems*, 2002, pp. 505–512.

[15] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in neural information processing systems*, 2005, pp. 1473–1480.

[16] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[17] Y. He, W. Chen, Y. Chen, and Y. Mao, "Kernel density metric learning," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 2013, pp. 271–280.

[18] J. Wang, H. T. Do, A. Woznica, and A. Kalousis, "Metric learning with multiple kernels," in *Advances in neural information processing systems*, 2011, pp. 1170–1178.

[19] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 11.

[20] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure." in *EMNLP-CoNLL*, vol. 7, 2007, pp. 410–420.