Frugal and Online Affinity Propagation

Xiangliang Zhang, Cyril Furtlehner, Michèle Sebag

Frugal and Online Affinity Propagation

Xiangliang Zhang, Cyril Furtlehner, Michèle Sebag

Laboratoire de Recherche en Informatique, CNRS UMR 8623 & INRIA-Ile de France
Bâtiment 490, Université Paris-Sud, 91405 - Orsay Cedex
{xlzhang,furtlehn,sebag}@lri.fr

Résumé : A new Data Clustering algorithm, Affinity Propagation suffers from its quadratic complexity in function of the number of data items. Several extensions of Affinity Propagation were proposed aiming at online clustering in the data stream framework. Firstly, the case of multiply defined items, or weighted items is handled using Weighted Affinity Propagation(WAP). Secondly, Hierarchical AP achieves distributed AP and uses WAP to merge the sets of exemplars learned from subsets. Based on these two building blocks, the third algorithm performs Incremental Affinity Propagation on data streams. The paper validates the two algorithms both on benchmark and on real-world datasets. The experimental results show that the proposed approaches perform better than K-centers based approaches.
Mots-clés : Data Clustering, Data Streaming, Affinity Propagation, K-centers

1 Introduction

Data Clustering, one major task in Unsupervised Learning, is concerned with structuring data items into clusters, enforcing the similarity of items belonging to a same cluster and their dissimilarity w.r.t. items in other clusters. While Unsupervised Learning has been acknowledged a core task of Machine Learning since the beginnings of the field, its theoretical foundations are less mature than those of Supervised Learning.

Many fundamental advances in Data Clustering however have been proposed since the mid 2000s. Ding et al. have highlighted the relationship between K-means and Principal Component Analysis (Ding & He, 2004). Based on this relationship, Meila has proposed a stability criterion for assessing clusters and shown the uniqueness of good optima for K-means (Meila, 2005, 2006). In the meanwhile, various criteria have been proposed to set the number K of clusters, e.g. based on Information Theory (Sugar & James, 2003), ROC curve (Jahanian *et al.*, 2004) or Dynamic Local Search (Karkkainen & Franti, 2002). Simultaneously, the topic of distance learning has been considered along different perspectives, e.g. related to accurate K-nearest neighbors (Weinberger *et al.*, 2005), or enforcing good margins (Hertz *et al.*, 2004), or correlated to information gain (Hillel & Weinshall, 2007).

The present paper is concerned with a new clustering approach, called Affinity Propagation(AP) and proposed by Frey & Dueck (2007a). This approach is suited to domains

where no artefact item (e.g. the barycenter of a set of items) can be constructed although a similarity or a distance function can be defined ; such domains involve e.g. molecular biology (the barycenter of a set of molecules is hard to define) or scheduling problems. In such spaces, data clustering is viewed as a combinatorial optimization problem : assuming the number K of clusters to be given, the goal is to select K items or *exemplars* in the dataset, such that the average distance from an item to the nearest exemplar, is minimal. This combinatorial optimization problem is tackled using a message passing algorithm, like belief propagation, detailed in section 2.

AP involves the acquisition of the similarity matrix, and the message passing algorithm. While the message passing algorithm converges with $N \log N$ complexity, the similarity matrix is computed with quadratic complexity, thus hindering the scalability os the approach. In Frey & Dueck (2007a), the similarity matrix is assumed to be given beforehand, or to involve a small fraction of the item pairs.

The goal of the paper is to address the limitation related to the quadratic complexity ; in order to do so, three extensions of the AP algorithm are proposed. Firstly, AP is extended to handle duplicated items in a transparent way, resulting in the Weighted AP (WAP) algorithm. Secondly, WAP is used to achieve Hierarchical AP, merging the exemplars independently learned from subsets of the whole dataset. Lastly, an incremental AP algorithm is defined, aimed to Data Streaming (section 3).

The paper is organized as follows. Section 2 presents the AP algorithm, and describes the first two proposed extensions, Weighted and Hierarchical AP. Section 3 describes the AP-based data streaming algorithm proposed, called STRAP. Section 4 describes the comparative validation of the proposed algorithms on benchmark problems. A real-world application, the clustering of 237,087 jobs submitted to a grid system, is finally considered. The paper concludes with a discussion and some perspectives for further research.

2 Affinity propagation and scalable variants

For the sake of self-containedness, this section first describes the AP algorithm, referring the reader to Frey & Dueck (2007a) and Frey & Dueck (2007b) for a comprehensive introduction. Two AP extensions are thereafter described, respectively handling the case of weighted items, and the merge of partial solutions.

2.1 Affinity propagation

Let $\mathcal{E} = \{e_1, \ldots e_N\}$ define a set of items, and let $d(i,j)$ denote the distance or dissimilarity between items e_i and e_j. Letting K denote a positive integer, the K-center problem consists of finding K items in $\mathcal{E}$, referred to as exemplars and denoted $e_{i_1}, \ldots, e_{i_K}$, such that they minimize the sum, over all items e_j, of the minimal squared distance between e_j and $e_{i_k}, k = 1 \ldots K$.

The Affinity Propagation approach proposes an equivalent formalization of the K-center problem, defined in terms of energy minimization. Let $\sigma(i)$ associate to each item e_i the index of its nearest exemplar, then the goal is to find the mapping σ maximizing

the functional $E[\sigma]$ defined as :

$$E[\sigma] = \sum_{i=1}^{N} S(i, \sigma(i)) - \sum_{i=1}^{N} \chi_i[\sigma] \tag{1}$$

where $S(i,j)$ is set to $-d(i,j)^2$ if $i \neq j$, and is set to a small constant $-s^*$, $s^* \geq 0$ called *preference* otherwise. The second term in the energy function represent a consistency constraint[1] : if e_i is an exemplar for others, it has to be its own exemplar ($\sigma(\sigma(e_i)) = \sigma(e_i)$), with

$$\chi_i[\sigma] = \begin{cases} \infty & \text{if } \sigma(\sigma(i)) \neq \sigma(i) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Aside from the consistency constraints, the energy function thus enforces a tradeoff between the distortion, i.e. the sum over all items of the squared error $d(i, \sigma(i))^2$ committed by assimilating item e_i to its nearest exemplar $e_{\sigma(i)}$, and the cost of the model, that is $s^* \times |\sigma|$ if $|\sigma|$ denotes the number of exemplars retained. Eq. (1) thus does not directly specify the number of exemplars to be found, as opposed to K-centers. Instead, it specifies the penalty s^* for allowing an item to become an exemplar ; note that for $s^* = 0$, the best solution is the trivial one, selecting every item as an exemplar.

The resolution of the optimization problem defined by Eq. (1) is achieved by a message passing algorithm, considering two types of messages : availability messages $a(i,k)$ express the accumulated evidence for e_k to be selected as the best exemplar for e_i ; responsibility messages $r(i,k)$ express the fact that e_k is suitable to be the exemplar of e_i.

All availability and responsibility messages $a(i,k)$ and $r(i,k)$ are set to 0 initially. Their values are iteratively adjusted[2] by setting :

$$r(i,k) = S(i,k) - \max_{k', k' \neq k} \{a(i,k') + S(i,k')\} \tag{3}$$

$$r(k,k) = S(k,k) - \max_{k', k' \neq k} \{S(k,k')\} \tag{4}$$

$$a(i,k) = min\{0, r(k,k) + \sum_{i', i' \neq i, k} \max\{0, r(i',k)\}\} \tag{5}$$

$$a(k,k) = \sum_{i', i' \neq k} \max\{0, r(i',k)\} \tag{6}$$

The exemplar $\sigma(i)$ associated to the i-th item is finally given by :

$$\sigma(i) = argmax\{r(i,k) + a(i,k), k = 1 \ldots N\} \tag{7}$$

The algorithm is stopped after a maximal number of iterations or when the exemplars did not change for a given number of iterations.

[1] A soft-constraint AP(SCAP) was proposed by Leone *et al.* (2007) to relax the hard constraint that the selected exemplar by other items has to be its own self-exemplar. This SCAP algorithm unveils the hierarchical cluster structure in the data sets instead of regularly shaped clusters.

[2] Numerical oscillations are avoided by using a relaxation mechanism ; empirically, the actual value is set to the half sum of the old and new values (Frey & Dueck, 2007a).

As could have been expected, Affinity Propagation is not to be seen as a universally efficient data clustering approach. Firstly, as mentioned in the introduction, linear and robust algorithms such as K-means should be preferred to AP in domains where artefact items can be constructed[3]. Secondly, if the desirable number K of clusters is small, then the combinatorial problem can be tackled by brute force (considering all N^K possible solutions). Lastly, and most importantly, AP suffers from a quadratic computational complexity in the number N of items (as all dissimilarities $d(i,j)$ must be computed), hindering its direct use in large-scale applications. As mentioned in the introduction, the computation cost of similarity matrix is not accounted for in Frey & Dueck (2007a). The next subsection aims to address this limitation.

2.2 Weighted and Hierarchical AP

Two possibilities can be considered in order to reduce the computational complexity of AP. The first one, left for further study, is based on uniformly sampling the dissimilarity matrix, computing the actual value $d(i,j)$ for a fraction of the pairs of items and setting $d(i,j)$ to the default value ∞ otherwise. The second one, considered in this paper, is based on a hierarchical extension of AP, splitting the whole dataset into $\sqrt{N}$ subsets, each including $\sqrt{N}$ items, and further clustering the sets of exemplars extracted from every subset.

2.2.1 Weighted AP

In order to do so, a preliminary step is to extend AP in order to deal with multiply-defined items. Let the dataset $\mathcal{E}$ be defined as in section 2.1, and let n_i be the number of copies of item e_i (in the default case, $n_i = 1$ for all i). The S matrix involved in the energy criterion (Eq. (1)) is thus naturally modified as follows. With no difficulty, the penalty $S(i,j)$ of selecting e_j as exemplar of e_i is multiplied by n_i ; as e_i actually represents a set of n_i identical copies, the penalty is n_i times the cost of selecting e_j as exemplar for each one of these copies.

Likewise by consistency with Eq. (1), the penalty $S(i,i)$ of selecting e_i as exemplar for itself is set to $s^* + (n_i - 1)\varepsilon_i$. Indeed, let item e_i be unfolded as a set of n_i (almost) identical copies $\{e_{i_1}, \ldots, e_{i_{n_i}}\}$, and let us assume that one of them, say e_{i_1} is selected as exemplar. One thus pays the *preference* penalty s^*, plus the sum of the dissimilarities between e_{i_1} and the other copies in e_i, modelled as $(n_i - 1)\varepsilon_i$. Constant ε_i thus models the average dissimilarity among the n_i copies of e_i.

Formally, let $\mathcal{E}' = \{(e_1, n_i), \ldots, (e_L, n_L)\}$, and define S' as :

$$S'(i,j) = \begin{cases} -n_i d^2(i,j) & \text{if } i \neq j \\ s^* + (n_i - 1) \times \varepsilon_i & \text{otherwise} \end{cases}$$

It is then straightforward to show that the combinatorial optimization problem defined

[3]Selecting the best set of artefacts out of τ independent runs of K-means usually enforce a high-quality distortion, with complexity $\tau \times K \times N$.

as : find σ minimizing

$$E'[\sigma] = \sum_{i=1}^{L} S'(i, \sigma(i)) - \sum_{i=1}^{L} \chi_i[\sigma]$$

is equivalent, for $\varepsilon_i = 0$, to the optimization problem defined by Eq. (1) for $\mathcal{E}$ made of the union of n_i copies of e_i, for $i = 1 \ldots L$.

2.2.2 Hierarchical AP

The WAP algorithm above is then used to cluster the sets of exemplars constructed from disjoint subsets of the whole dataset. Formally, let $\mathcal{E}$ be divided into $\sqrt{N}$ subsets of equal size, noted $\mathcal{E}_i, i = 1 \ldots \sqrt{N}$.

Let $\{e_{i_1,}, \ldots e_{i_{K_i}}\}$ be the set of exemplars extracted from $\mathcal{E}_i$, with n_{i_j} the number of items in $\mathcal{E}_i$ having e_{i_j} as nearest exemplar.

Consider the weighted AP problem defined from $\mathcal{E}' = \{(e_{i_j}, n_{i_j}), i = 1 \ldots \sqrt{N}, j = 1 \ldots K_i\}$.

Note that the construction of $\mathcal{E}'$ is in $\mathcal{O}(N^{\frac{3}{2}})$. Letting K be an upper bound on the number of exemplars learned from every subset $\mathcal{E}_i$, WAP thus achieves the hierarchical clustering of the exemplars extracted from all $\mathcal{E}_i$ with complexity $\mathcal{O}(N^{\frac{1}{2}} \times K^2)$.

Further work is concerned with examining and bounding the energy loss entailed by solving the WAP problem defined from $\mathcal{E}'$ with complexity $\mathcal{O}(N^{\frac{1}{2}} \times (N + K^2))$ instead of the initial AP problem defined from $\mathcal{E}$ with complexity $\mathcal{O}(N^2)$.

3 Incremental AP and Data Streaming

This section describes the proposed extension from AP and Weighted AP to Data Streaming. Data Streaming, one of the hottest topics in Data Mining (Fan *et al.*, 2004; Aggarwal *et al.*, 2003; Guha *et al.*, 2000), aims to provide a compact description of the data flow (Muthukrishnan, 2005) and/or the frequent patterns or anomalies thereof. It imposes an additional constraint on Data Mining techniques, the fact that each data item can be seen only once due to the fast rate of acquisition.

The general schema proposed to extend AP to Data Streaming (called STRAP , Alg. 1) involves four main steps besides the initialization.

1. The first bunch of data is used by AP to compute the first exemplar-based model.
2. Each new item is compared to the exemplars ; if the new item is too dissimilar wrt the current exemplars (section 3.1), it is put in the reservoir.
3. The restart criterion is triggered if the reservoir size exceeds some threshold, or if some drift in the data distribution is detected (section 3.2).
4. If it is triggered, WAP is restarted with the current exemplars and the reservoir ; new exemplars are thus obtained and the associated model is computed (section 3.3).
5. The process goes to step 2.

At every time step, the current model of the data flow is represented by the exemplars and their distribution. The performance of the process is measured from the average distortion and the overall size of the model, detailed in section 3.4.

Algorithm 1 WAP-based Data Streaming

Datastream $e_1, \ldots e_t, \ldots$ **; fit threshold** ϵ
Init
  AP$(e_1, \ldots, e_T) \rightarrow$ Exemplar-based Model
  Reservoir = $\{\}$
for $t > T$ **do**
  Compute Fit(e_t, current model) — section 3.1
  if $Fit > \epsilon$ **then**
    Update model — section 3.3
  else
    Reservoir $\leftarrow e_t$
  end if
  — section 3.2
  if Restart criterion **then**
    Update model by WAP — section 3.3
  end if
end for

3.1 WAP-based Modelling

While AP only aims to provide the exemplars best representing the dataset according to the energy criterion (Eq. 1), STRAP might need some additional information in order to see whether a new item should be allocated to some exemplar or rather considered to be an outlier at this point.

The proposed model, inspired from DBSCAN (Ester, 1996), characterizes each exemplar e_i from a 4-tuple $(e_i, n_i, M_i, \Sigma_i)$, where :

n_i is the number of items associated so far to exemplar e_i ;
M_i is the sum of the distances between these items and e_i ;
Σ_i is the sum of the squared distances between these items and e_i.

This model, as DBSCAN , enables an additive, computationally efficient update when a new item is associated to any exemplar. It supports three alternative measures in order to evaluate the relevancy between some new item e and any exemplar e_i :

- **Energy-based**. The first criterion simply measures the distance $d(e, e_i)$ between the new item e and exemplar e_i. Item e is associated to the nearest exemplar provided that the associated squared distance is less than the energy threshold s^* (section 2.1), i.e. the cost of turning the new item in an exemplar *per se*. Otherwise, item e is put in the reservoir.
- **Prior-based**. Considering the set of items associated to each exemplar and the associated distances, the set of such distances is modelled as a Gaussian[4] distribution centered on $\mu_i = \frac{M_i}{n_i}$ with variance $\sigma_i = \sqrt{\frac{\Sigma_i}{n_i} - \frac{M_i^2}{n_i^2}}$. The relevancy between item e and exemplar e_i is then measured as :

$$F(e, e_i) = Pr(d(e, e_i)|\mathcal{N}(\mu_i, \sigma_i))$$

[4]Naturally, considering that the set of distances follows a Gaussian distribution is a coarse approximation, since distances are necessarily greater than 0 and smaller than the smallest distance to the other exemplars.

Let e_i^* be the exemplar maximizing $F(e, e_i)$; item e is associated to exemplar e_i^* except if $F(e, e_i^*)$ is lower than some user-given threshold ϵ, in which case the new item is put in the reservoir.

– **Posterior-based**. The above measure does not take into account the actual number of items associated to exemplars. The third relevancy criterion is thus defined as :

$$F_B(e, e_i) = Pr(d(e, e_i)|\mathcal{N}(\mu_i, \sigma_i)) \times Pr(e_i)$$

where $Pr(e_i)$ is the fraction of items associated to exemplar e_i ($Pr(e_i) \propto n_i$). As in the previous case, item e is associated to exemplar e_i^* maximizing the criterion, except if $F_B(e, e_i^*)$ is lower than some user-given threshold ϵ_B, in which case the new item is put in the reservoir.

3.2 Restart criterion

The core difficulty in Data Streaming is to deal with outliers and detect some drift in the item generative process. As a matter of fact, when a poor fit with the current exemplars is observed, there is in general no easy way to tell outliers from items generated after the new process distribution.

In the case of drift i.e. when the generative process has changed, the stream model must be updated. While in some application domains, the model update can be smoothly achieved through updating the clusters and their centers (e.g. in continuous spaces), AP-relevant domains requires the definition of new exemplars. Therefore the data streaming process needs a restart criterion, in order to decide whether the construction of new exemplars from the current ones and the reservoir should be launched.

Two restart criteria have been considered. The first one is most simply based on the size of the reservoir criterion. When the reservoir is filled with items, the construction of new exemplars based on the current exemplars and the items in the reservoir is launched. In this case, some care must be exercised (section 3.3) in order to ensure that i) the number of new exemplars does not grow beyond control ; ii) relevant exemplars are not sacrificed to outliers.

The second criterion is based on a change point detection test. Let us consider the flow of items e_t, and the sequence $p_t = max_i F(e_t, e_i)$ of their relevancy measure wrt the current exemplars. If the item generative process is drifting, then sequence p_t should display some change ; the restart criterion is triggered upon detecting such a change.

The so-called Page-Hinkley change-point-detection test (Page, 1954; Hinkley, 1970, 1971) has been selected as it minimizes the expected detection time for a prescribed false alarm rate. Formally, the PH test is controlled after a detection threshold λ and tolerance δ, as follows :

$$\bar{p}_t = \frac{1}{t} \sum_{\ell=1}^{t} p_\ell \tag{8}$$

$$m_t = \sum_{\ell=1}^{t} (p_\ell - \bar{p}_\ell + \delta) \tag{9}$$

$$M_t = max\{|m_\ell|, \ell = 1...t\} \tag{10}$$

$$PH_t = (M_t - m_t) > \lambda \tag{11}$$

In this latter case, it might happen that the reservoir is filled before the restart criterion is triggered. In such case, the new item put in the reservoir replaces the oldest one ; a counter keeping track of the removed reservoir items is incremented.

3.3 Model update

In the case where a new item e is associated to an existing exemplar e_i, model $(e_i, n_i, M_i, \Sigma_i)$ is most simply updated[5], by incrementing n_i, adding $d(e, e_i)$ (respectively, $d(e, e_i)^2$) to M_i (resp. Σ_i).

Upon triggering of the restart criterion, Weighted AP is launched on the set of weighted items involving i) the current exemplars $e_i, i = 1 \ldots N$ together with their size n_i ; ii) the reservoir items noted $e'_j, j = 1 \ldots M$, with $n'_j = 1$. The question is how to adjust the penalties $S(e_i, e_i)$ and the distances $S(e_i, e'_j)$ in order to prevent the number of final exemplars from increasing beyond control, and to avoid sacrificing relevant exemplars to many outliers.

After section 2.2.1, it comes :

$$\begin{array}{ll} S(e_i, e_i) & = s^* + \Sigma_i \\ S(e'_j, e'_j) & = s^* \\ S(e_i, e_j) & = -n_i d(e_i, e_j)^2 \\ S(e_i, e'_j) & = -n_i d(e_i, e'_j)^2 \\ S(e'_j, e_i) & = -d(e_i, e'_j)^2 \end{array}$$

Let $f_1, \ldots f_K$ denote the exemplars constructed by WAP. The next point is to construct the associated model from the previous model $\{(e_i, n_i, M_i, \Sigma_i)\}$ and the reservoir items, granted that the items originally involved in the extraction of exemplars e_i are no longer available.

Formally, let f be a new exemplar, let $e_1, \ldots e_m$ (respectively $e'_1, \ldots, e'_{m'}$) be previous exemplars (resp. reservoir items) associated to f. With no difficulty, the number n of items associated to f is set to $n_1 + \ldots + n_m + m'$.

The sum of distances of the items to f is estimated after an Euclidean model as follows. Let e be an item associated to e_1. After the Euclidean model, e is viewed as a random item $e_1 + X\vec{v}$, where $\vec{v}$ is a random vector in the unit ball, and X is a random variable with distribution $\mathcal{N}(\mu_1, \sigma_1)$. One has :

$$\begin{array}{ll} ||f - e||^2 & = ||f - e_1||^2 + ||e_1 - e||^2 - 2\langle f - e_1, X\vec{v}\rangle \\ & = d(f, e_1)^2 + d(e_1, e)^2 - 2X\langle f - e_1, \vec{v}\rangle \end{array}$$

Taking the expectation, it comes $E[d(f, e)^2] = d(f, e_1)^2 + \frac{1}{n_1}\Sigma_1$. Accordingly,

$$\Sigma = \sum_{i=1}^{m} \left(n_i d(f, e_i)^2 + \Sigma_i\right) + \sum_{i=1}^{m'} d(f, e'_i)^2$$

[5]Further work is concerned with using relaxation-based update mechanism, in order to decrease the influence of the oldest items associated to the exemplar.

Along the same ideas, assuming that items associated to a given exemplar are independent, M is approximated to

$$M = M_0 + \sum_{i=1}^{m} (n_i d(f, e_i)) + \sum_{i=1}^{m'} d(f, e'_i)$$

where M_0 is the M value of f before other exemplars and items are merged to it.

3.4 Evaluation criterion

An evaluation criterion, inspired from the energy criterion (Eq. 1), is proposed in order to assess the STRAP algorithm. This criterion measures the trade-off between the average distortion and the average size of the model.

The average size of the model is defined after the total number of exemplars constructed, divided by the number of restarts + 1. The distortion D is computed as follows :

- If some new item e is associated to exemplar e_i, D is incremented by $d(e, e_i)^2$;
- Otherwise, e is put in the reservoir ; after the next restart, the average square distance $\bar{d^2}$ of the reservoir items to the new exemplars is computed, and D is incremented by $\bar{d^2}$ times the number of items put in the reservoir since the last restart[6].

4 Experimental Validation and Discussion

In this section, we first compare the distortion of AP with the best distortion of 20 independent runs of K-centers on the same time series benchmarks. We then assessed Hierarchical AP on the two largest benchmark data sets. Finally, Hierarchical AP is evaluated on a real world data set. The distortion is defined as

$$D([\sigma]) = \sum_{i=1}^{N} d(i, \sigma(i))^2 \tag{12}$$

Hierarchical AP is validated by comparing with K-centers. For showing the performance of WAP, we use both AP and WAP for clustering. Formally, letting N be the total size of the dataset $\mathcal{E}$, $\mathcal{E}$ is partitioned into $\sqrt{N}$ subsets of equal size noted $\mathcal{E}_i$.

- Hierarchical AP proceeds as follows :
  1. On each subset $\mathcal{E}_i$, the preference s_i^* is set to the median of the pair differences in the subset ; AP(WAP) is run and defines a set of K_i exemplars noted e_{i_j}, each of those represents n_{i_j} items in $\mathcal{E}_i$.
  2. Letting $\mathcal{E}'$ denote the set of (e_{i_j}, n_{i_j}) for $i = 1 \ldots \sqrt{N}, j = 1 \ldots K_i$, AP(WAP) is launched on $\mathcal{E}'$ with various values of the preference s^*. The associated number of final exemplars and distortion are reported.
- Simultaneously :

[6] This procedure is meant to handle the case of items removed from the reservoir, when the restart criterion is based on the change point detection test, section 3.2.

1. K-centers is applied on each subset $\mathcal{E}_i$, with K set to the average of K_i over $i = 1 \ldots \sqrt{N}$.
2. The best K-centers out of 120 independent runs in terms of the distortion on $\mathcal{E}_i$ are kept ; their union defines the set of centers $\mathcal{C}$.
3. K-centers is applied to the total set $\mathcal{E}$, with the constraint that the centers must belong to $\mathcal{C}$. For various values of K, K-centers is run independently 20 times, and the best distortion is kept. The independent launch times of K-centers are set to make its running time comparable with WAP.

– Finally, the three curves $(K, \text{distortion}(K))$ are compared.

4.1 Validation on benchmarks

13 benchmark datasets kindly provided by E. Keogh have been considered (Keogh *et al.*, 2006), ranging over diverse application domains, e.g. images, videos, texts. On each data set, the distance considered is the Euclidean one and the "ground truth" clusters are defined by the classes.

Two experimental settings have been considered. In the first one (A), the number K of centers is set to the number of classes and the preference s^* is tuned so as the number of exemplars is K. In the second one (B), the preference is set to the median squared distance among pairs of items and K is set to the number of exemplars thus obtained with AP.

TAB. 1 – Comparison of K-centers (best of 20 runs) and AP when K is set to the number of classes and the preference s^* is tuned so as the number of exemplars is K

Data	K	N	D	Distortion		Distortion of Hierarchical clustering		
				KC	AP	KC	AP	WAP
1	6	600	60	24014	**23719**	/	/	/
2	2	200	150	4422	*4422*	/	/	/
3	3	930	128	78326	*78326*	/	/	/
4	14	2250	131	189370	**183265**	198658	190496	**189383**
5	6	442	427	151351	**149615**	/	/	/
6	15	1125	128	20220	**19079**	20731	20248	**20181**
7	50	905	270	93749	**85558**	/	/	/
8	4	200	275	10054	10072	/	/	/
9	4	112	350	24443	24447	/	/	/
10	2	121	637	65783	67104	/	/	/
11	7	143	319	25596	**25274**	/	/	/
12	2	200	96	6424	*6424*	/	/	/
13	37	781	176	547	**356**	/	/	/

In both cases, the distortion obtained by AP is compared with the best distortion of K-centers out of 20 independent runs. Table 1 reports on the results obtained for

experimental setting (A) : K is the given number of classes, N is the number of items in the dataset, D the dimension. The distortion of batch clustering, on the whole data set, is reported in the left part of Table 1. The performance of Hierarchical AP on the two largest data sets is also shown in the right part of Table 1.

TAB. 2 – Comparison of K-centers (best of 20 runs) and AP when K depends on AP

Data	K	K_AP	Distortion		K_HAP	Distortion of Hierarchical clustering		
			KC	AP		KC	AP	WAP
1	6	35	18528	**17522**	/	/	/	/
2	2	12	858	**813**	/	/	/	/
3	3	47	44088	**42593**	/	/	/	/
4	14	168	100420	**88282**	39	172359	164175	**160415**
5	6	41	90798	**83795**	/	/	/	/
6	15	100	12682	**9965**	23	21525	**20992**	21077
7	50	62	87426	**78996**	/	/	/	/
8	4	9	4529	4651	/	/	/	/
9	4	13	15315	**14662**	/	/	/	/
10	2	17	37826	**35466**	/	/	/	/
11	7	16	20480	**19602**	/	/	/	/
12	2	14	2254	**2172**	/	/	/	/
13	37	70	412	**216**	/	/	/	/

These results suggest that AP is more appropriate for complex datasets, where the underlying structure of the domain involves many clusters. As could have been expected, Hierarchical AP uses less information than batch clustering and entails a slightly higher distortion.

In Hierarchical AP, WAP performs better than AP given the same set of exemplars learned from the subsets. WAP merges the exemplars considering their potential ability of being a bigger exemplar by passing weighted messages. AP, by contrast, fairly groups the exemplars.

Table 2 reports on the results obtained for experimental setting (B), when the number of clusters is set as K_AP. K_AP is the number of clusters obtained with AP when the preference s^* is set to the median distance. K_AP is larger than the K given by the data. In the Hierarchical AP validation, K_HAP is the final number of clusters using AP for subset clustering and then exemplars clustering. The preference s^* used by WAP is tuned to have also K_HAP final clusters. The K of K-centers in the exemplars clustering is also set to be K_HAP. In the subset clustering, K of K-centers is set to be N_{all}/N_s, where N_{all} is the total number of clusters learned from all subsets, and N_s is the number of subsets. Left part of Table 2 is the results of batch clustering and right part is the result of hierarchical clustering.

Hierarchical AP significantly decreased the clustering computation time compared with batch clustering, in spite of a slightly higher distortion. On the 4-th dateset, Hierarchical AP spent only 3 seconds while batch AP clustering spent 128 seconds. On

the 6-th dateset, Hierarchical AP spent 1.4 seconds while batch AP clustering spent 21 seconds.

4.2 Validation on real-world data

This validation considers a real-world dataset, the set of jobs submitted to the EGEE grid system[7], which will be described first.

4.2.1 Job stream

The considered dataset describes the states of the arrived jobs from 2006-03-14 to 2007-02-06, including 237,087 jobs each described by five attributes :

1. the time when a job arrived at a queue ;
2. the time when a job began to execute ;
3. the time when the job is finished ;
4. the identifier of the user who submitted the job ;
5. the identifier of queue by which the job was transited.

In the data preprocessing step, new features were derived from these initial ones and the user identifiers were removed. Finally, a job is described by the following features :

1. the duration of waiting time in a queue ;
2. the duration of execution ;
3. the number of jobs waiting in the queue when the current job arrived ;
4. the number of jobs being executed after the transition of this queue when the current job arrived ;
5. the identifier of queue by which the job was transited.

This representation makes it impossible to consider job artefact ; the behavior might be significantly different from one queue to another and the expert is willing to extract representative actual jobs as opposed to virtual ones (e.g. executed on queue 1 with weight .3 and on queue 2 with weight .7).

The dissimilarity of two jobs x_i and x_j is the sum of the Euclidean distance between the numerical description of x_i and x_j, plus a weight w_q if x_i and x_j are not executed on the same queue.

TAB. 3 – Parameters and running time of subset clustering on real-world jobs

Algorithm	parameter	running time	N. of exemplars
K-centers	K = 15	10 mins	7290
AP	s^* = median(S)	26 mins	8444
WAP	s^* = median(S)	10 mins	7531

[7] http ://www.eu-egee.org/

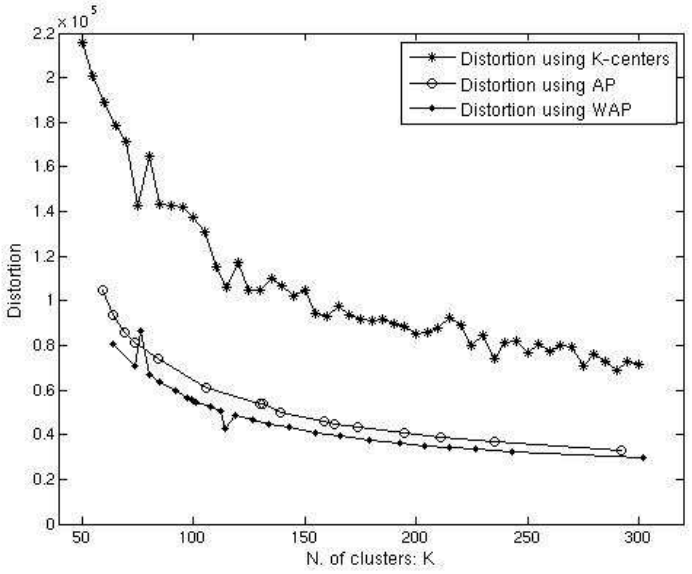

FIG. 1 – Distortion of hierarchical AP and K-centers on real-world jobs

The validation of hierarchical AP was conducted on this real-world dataset. The whole data, 237,087 jobs, is divided into 486 subsets and each subset includes 486 jobs. We used K-centers, AP and WAP respectively on each subset to get exemplars. WAP is used on subset clustering because there are around 30% duplications in the real-world data. The parameters, the number of exemplars and running time are shown in Table 3. K-centers is independently launched 120 times to make its running time comparable with WAP. The best results which have lowest distortion are reported. All the experiments were conducted on a Intel 2.66GHz Dual-Core PC with 2 GB memory by Matlab codes.

K-centers, AP and WAP are applied on the set of exemplars learned from the subsets. The distortions on different number K of clusters are shown in Fig. 1.

The Fig. 1 shows that WAP-based hierarchical clustering has lower distortion than AP-based and K-centers based hierarchical clustering. The proposed approach scales down the computation complexity of large-size data with roughly one third of the distortion when compared with K-centers.

5 Conclusion and Perspectives

In this paper we have explored the possibility of using the affinity propagation algorithm to perform online clustering of data stream and set a general approach for this purpose. Frey & Dueck (2007a) have shown that AP performs better than K-centers

clustering especially on sufficiently complex problems. Considering the possible huge amount of data flow which is supposed to be treated in real applications (e.g. job error detection in grid computing), the main step is to adapt the scalability of AP.

To overcome the N^2 complexity of AP (caused by the computation of the similarity matrix), we firstly proposed the Weighted AP by aggregating the similar items into one single item.

The second algorithm achieves hierarchical clustering, by building exemplars from subsets of the initial dataset and aggregating them using WAP. Experimental validation demonstrates that hierarchical AP is competitive with K-centers on large datasets.

The third proposed algorithm, STRAP , achieves data streaming based on Hierarchical AP. Further research is concerned with experimental validation of STRAP and bounding the distortion loss due to the distributed computing of exemplars from different subsets.

Références

AGGARWAL C., HAN J., WANG J. & YU P. (2003). A framework for clustering evolving data streams. In *Proceedings of the International Conference on Very Large Data Bases(VLDB)*, p. 81–92.

DING C. & HE X. (2004). K-means clustering via principal component analysis. In *International Conference on Machine learning (ICML)*, p. 225–232.

ESTER M. (1996). A density-based algorithm for discovering clusters in large spatial databases with noisethe uniqueness of a good optimum for k-means. In *Proceedings of Second International Conference on Knowledge Discovery and Data Mining(KDD)*, p. 226–231.

FAN W., WANG H. & YU P. (2004). Active mining of data streams. In *SIAM International Conference on data mining SDM'2004*.

FREY B. & DUECK D. (2007a). Clustering by passing messages between data points. In *Science*, volume 315, p. 972–976.

FREY B. & DUECK D. (2007b). Supporting online material of clustering by passing messages between data points. In *Science*, volume 315, http ://www.sciencemag.org/cgi/content/full/1136800/DC1.

GUHA S., MISHRA N., MOTWANI R. & O'CALLAGHAN L. (2000). Clustering data streams. In *IEEE Symposium on Foundations of Computer Science*, p. 359–366.

HERTZ T., BAR-HILLEL A. & WEINSHALL D. (2004). Boosting margin based distance functions for clustering. In *International Conference on Machine learning (ICML)*, p. 50–58.

HILLEL A. & WEINSHALL D. (2007). Learning distance function by coding similarity. In *Proceedings of the 24th International Conference on Machine learning(ICML)*, p. 65–72.

HINKLEY D. (1970). Inference about the change-point in a sequence of random variables. In *Biometrika*, volume 57, p. 1–17.

HINKLEY D. (1971). Inference about the change-point from cumulative sum tests. In *Biometrika*, volume 58, p. 509–523.

JAHANIAN H., ZADEH H., HOSSEIN-ZADEH G. & SIADAT M. (2004). Roc-based determination of number of clusters for fmri activation detection. In *Proceedings of SPIE Medical Imaging*, volume 5370, p. 577–586.

KARKKAINEN I. & FRANTI P. (2002). Dynamic local search for clustering with unknown number of clusters. In *16th International Conference on Pattern Recognition (ICPR)*, volume 2, p. 240–243.

KEOGH E., XI X., WEI L. & RATANAMAHATANA C. A. (2006). The ucr time series classification/clustering homepage : www.cs.ucr.edu/ eamonn/time_series_data/.

LEONE M., , SUMEDHA & WEIGT M. (2007). Clustering by soft-constraint affinity propagation : Applications to gene-expression data. *Bioinformatics*, **23**, 2708.

MEILA M. (2005). Comparing clustering - an axiomatic view. In *International Conference on Machine learning (ICML)*, p. 577–584.

MEILA M. (2006). The uniqueness of a good optimum for k-means. In *International Conference on Machine learning (ICML)*, p. 625–632.

MUTHUKRISHNAN S. (2005). Data streams : Algorithms and applications. In *Found. Trends Theor. Comput. Sci.*, volume 1, p. 117–236 : Now Publishers Inc.

PAGE E. (1954). Continuous inspection schemes. In *Biometrika*, volume 41, p. 100–115.

SUGAR C. & JAMES G. (2003). Finding the number of clusters in a dataset : An information-theoretic approach. In *Journal of the American Statistical Association*, volume 98, p. 750–763.

WEINBERGER K. Q., BLITZER J. & SAUL L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *NIPS*, p. 1473–1480 : MCambridge, MA : MIT Press.