On-line Evolutionary Exponential Family Mixture

Jianwen Zhang, Yangqiu Song, Gang Chen and Changshui Zhang
State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing 100084, China
{jw-zhang06, songyq99, g-c05}@mails.tsinghua.edu.cn, zcs@mail.tsinghua.edu.cn

Abstract

This paper deals with evolutionary clustering, which refers to the problem of clustering data with distribution drifting along time. Starting from a density estimation view to clustering problems, we propose two general on-line frameworks. In the first framework, i.e., *historical data dependent* (HDD), current model distribution is designed to approximate both current and historical data distributions. In the second framework, i.e., *historical model dependent* (HMD), current model distribution is designed to approximate both current data distribution and historical model distribution. Both frameworks are based on the general *exponential family mixture* (EFM) model. As a result, all conventional clustering algorithms based on EFMs can be extended to evolutionary setting under the two frameworks. Empirical results validate the two frameworks.

1 Introduction

Clustering is a fundamental problem in machine learning and data mining. Conventional clustering algorithms, such as k-means [Hartigan and Wong, 1979] and spectral clustering [Ng *et al.*, 2002], focus on static data and assume all the data are I.I.D. (Independent and Identically-Distributed) samples from one underlying distribution. However, in lots of dynamic applications, data come from different time epochs. Due to concept drifting or noise varying, the distribution of epoch data often drifts along time. For example, contents under the topic "life style" in a Bulletin Board System (BBS) often differ from those of one year ago while not deviating too much. The clustering task on this kind of data raised the problem of *evolutionary clustering* [Chakrabarti *et al.*, 2006]. In this case, the final target is to provide a set of partitions, one for each time epoch. In addition, as the data distributions of adjacent epochs are close to each other, the clustering results of epochs should be smooth along time.

We should distinguish evolutionary clustering from *incremental clustering* [Charikar *et al.*, 1997]. Incremental clustering gives a single partition for all the data, although the data enter into the algorithm sequentially. Two properties are emphasized in incremental clustering, the first is the one-pass manner of the access to data, and the second is the equivalence between the original non-incremental algorithm and the corresponding incremental one.

The necessity of evolutionary clustering lies in two aspects. First, when distribution drifts, applying a conventional clustering algorithm to overall data may not be appropriate. Second, if we apply a conventional clustering algorithm independently to each epoch data, the smoothness of clustering results along time can not be preserved. The second aspect can be realized from two facts. (1) For non-deterministic clustering algorithms relying on initialization, such as k-means, Gussian Mixture Model (GMM), etc., the clustering results of adjacent epochs may be quite different from each other due to local optima, even when the two distributions are almost the same. (2) For deterministic clustering algorithms, such as spectral clustering and agglomerative hierarchical clustering, data noise may lead to different clustering results between adjacent epochs.

Evolutionary clustering can be off-line or on-line[1]. Two off-line methods have been proposed by [Wang *et al.*, 2007] and [Ahmed and Xing, 2008].

The first on-line method is proposed by [Chakrabarti *et al.*, 2006]. In their approach, the smoothness property is ensured by adding a temporal loss to the original loss of static clustering. The temporal loss penalizes the deviation of current clustering result from the historical. Using the approach, they proposed an evolutionary k-means algorithm: each center at epoch i should be matched to the nearest center at $i-1$ as a pair, and distances between all pairs of centers were summed as the temporal loss. As pointed out in [Chi *et al.*, 2007], this heuristic approach could be unstable, i.e., sensitive to small perturbation on the centers. Using the same idea, [Chi *et al.*, 2007] extended spectral clustering to evolutionary setting. Moreover, [Tang *et al.*, 2008] extended evolutionary spectral clustering further to multi-relational clustering. However, in [Chi *et al.*, 2007] and [Tang *et al.*, 2008], data to be clustered at different time epochs should be identical, i.e., data of epochs are "snapshots" of the same set of objects at different time. These kind of methods have difficulties to deal with the scenario when data of different epochs are arbitrary I.I.D.

[1] When doing clustering at epoch i, in off-line setting, the overall data of all epochs are available, while in on-line setting, only the data before epoch i are available.

samples from different underlying distributions. However, in many cases, we desire a good solution which is able to deal with the variation of data size and cluster number.

In this paper, we focus on the on-line setting when data of different epochs need not be identical. Starting from a density estimation view to clustering, we propose two general frameworks. In the first framework, i.e., *historical data dependent* (HDD), current model distribution is designed to approximate both current and historical data distributions. In the second framework, i.e., *historical model dependent* (HMD), current model distribution is design to approximate both current data distribution and historical model distribution. Both frameworks are based on the general *exponential family mixture* (EFM) model. As a result, all conventional clustering algorithms based on EFMs can be extended to evolutionary setting under the two frameworks. Experiments on both synthetic and real data sets demonstrate the validation of the two frameworks.

2 Notations and Preliminaries

$\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ denotes observed data, which are i.i.d samples from an unknown underlying distribution $F(\mathbf{x})$ (with density $f(\mathbf{x})$). About the superscript, $\mathbf{x}^{(i)}$ denotes an item at time epoch i, while $\mathbf{x}^{[t]}$ denotes an item at the t'th step in an iterating algorithm. $\mathbf{E}_f[\cdot]$ is the expectation under distribution f.

2.1 Exponential Family Mixture (EFM)

An exponential family is a probability distribution set $\mathcal{F}_\Psi$, from which each density function can be expressed in the form

$$p_\Psi(\mathbf{x};\boldsymbol{\theta}) = \exp\left\{\langle\boldsymbol{\theta}, T(\mathbf{x})\rangle - \Psi(\boldsymbol{\theta})\right\} p_0(T(\mathbf{x})) \tag{1}$$

where $\boldsymbol{\theta}$, $T(\mathbf{x})$, and $\Psi(\boldsymbol{\theta})$ are called *natural parameter*, *natural statistic*, and *cumulant function*, respectively.

[Banerjee *et al.*, 2005] stated that each exponential family distribution can be uniquely expressed using *Bregman divergence*

$$p_\Psi(\mathbf{x},\boldsymbol{\theta}) = \exp\left\{-d_\phi(T(\mathbf{x}), \boldsymbol{\mu}(\boldsymbol{\theta}))\right\} b_\phi(T(\mathbf{x})) \tag{2}$$

where ϕ and b_ϕ are functions uniquely determined by Ψ, d_ϕ is the Bregman divergence derived from ϕ, and $\boldsymbol{\mu}$ is the *expectation parameter* $\boldsymbol{\mu}(\boldsymbol{\theta}) = \mathbf{E}_{p_\Psi(\mathbf{x},\boldsymbol{\theta})}[T(\mathbf{x})]$. Parameters $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ are linked by

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}\Psi, \text{and } \boldsymbol{\theta}(\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}}\phi. \tag{3}$$

For some widely used exponential families, the specific forms of above parameters can be found in [Banerjee *et al.*, 2005].

A mixture model refers to a parametric distribution model with the following form:

$$p(\mathbf{x};\boldsymbol{\Xi}) = \sum\nolimits_z^C \alpha_z p(\mathbf{x};\boldsymbol{\theta}_z), \text{with} \sum\nolimits_z \alpha_z = 1 \tag{4}$$

where C is the component number, $z \in \mathcal{C} = \{1, \cdots, C\}$ is the component indicator variable, and $\boldsymbol{\Xi} = \{\alpha_z, \boldsymbol{\theta}_z\}_{z=1}^C$ are model parameters. When the components are taken in an exponential family $\mathcal{F}_\Psi$, we get the general *exponential family mixture* (EFM) model. Typical examples of EFMs are GMM, multinomial mixture model (MMM), etc., with different definitions on Ψ or d_ϕ.

2.2 Clustering as Density Estimation

From the view of statistical learning theory, density estimation is to find a model distribution $p(\mathbf{x};\boldsymbol{\Xi})$ minimizing an expected loss (risk) (*Fisher-Wald setting*) [Vapnik, 2000]:

$$\mathcal{L}(\boldsymbol{\Xi}) = -\int \log p(\mathbf{x};\boldsymbol{\Xi}) dF(\mathbf{x}) = -\mathbf{E}_f\left[\log p(\mathbf{x};\boldsymbol{\Xi})\right] \tag{5}$$

on the unknown true distribution $F(\mathbf{x})$. Notice that $\mathcal{L}(\boldsymbol{\Xi})) + \int f(x)\log f(x)dx = KL(f||p)$, where $KL(\cdot||\cdot)$ denotes Kullback-Leibler (KL) divergence between two distributions. So density estimation is equivalent to minimizing the KL divergence between $f(\mathbf{x})$ and $p(\mathbf{x};\boldsymbol{\Xi})$.

A mixture model as Eq. (4) can be adopted to estimate $f(\mathbf{x})$. Based on this mixture model, it's well known that $\mathcal{L}$ is difficult to minimize, and what will be minimized actually is a variational convex upper bound [Beal, 2003]:

$$\begin{aligned}
&\mathcal{L}(p(\mathbf{x};\boldsymbol{\Xi})) = -\int \log\left[\sum\nolimits_z \alpha_z p(\mathbf{x};\boldsymbol{\theta}_z)\right] d\,F(\mathbf{x}) \\
&\le -\int \sum\nolimits_z \left[q_{\mathbf{x}}(z)\log\frac{\alpha_z p(\mathbf{x};\boldsymbol{\theta}_z)}{q_{\mathbf{x}}(z)}\right] d\,F(\mathbf{x}) \\
&= \underbrace{-\int \sum\nolimits_z \left[q_{\mathbf{x}}(z)\log\left(\alpha_z p(\mathbf{x};\boldsymbol{\theta}_z)\right)\right] d\,F(\mathbf{x})}_{\mathcal{E}(q_{\mathbf{x}}(\cdot),\boldsymbol{\Xi})} \\
&\qquad + \underbrace{\int \sum\nolimits_z q_{\mathbf{x}}(z)\log q_{\mathbf{x}}(z) d\,F(\mathbf{x})}_{\mathcal{H}(q_{\mathbf{x}}(\cdot))} = \mathcal{G}(q_{\mathbf{x}}(\cdot),\boldsymbol{\Xi})
\end{aligned} \tag{6}$$

where $q_{\mathbf{x}}(\cdot)$ is a distribution of z determined by $\mathbf{x}$. The "$\le$" is derived from Jensen's inequality, with the "$=$" holding iff $q_{\mathbf{x}}(\cdot) = p(\cdot|\mathbf{x};\boldsymbol{\Xi})$.

The well known EM procedure is used to minimize the variational bound $\mathcal{G}$:

$$\begin{aligned}
&\mathbf{E}\text{-step: } q_{\mathbf{x}}^{[t+1]}(\cdot) \leftarrow \arg\min\nolimits_{q_{\mathbf{x}}(\cdot)} \mathcal{G}(q_{\mathbf{x}}(\cdot), \boldsymbol{\Xi}^{[t]}) \\
&\mathbf{M}\text{-step: } \boldsymbol{\Xi}^{[t+1]} \leftarrow \arg\min\nolimits_{\boldsymbol{\Xi}} \mathcal{E}(q_{\mathbf{x}}^{[t+1]}(\cdot), \boldsymbol{\Xi})
\end{aligned} \tag{7}$$

In $\mathbf{E}$-step, $q_{\mathbf{x}}(\cdot)$ actually gives a solution to clustering. If no additional constraints are enforced upon $q_{\mathbf{x}}(\cdot)$, the optimal solution is

$$q_{\mathbf{x}}^{[t+1]}(\cdot) = p(\cdot|\mathbf{x}, \boldsymbol{\Xi}^{[t]}). \tag{8}$$

We call this case *soft-clustering*, e.g., GMM. If we constrain $\forall z \in \mathcal{C}, q_{\mathbf{x}}(z) \in \{0,1\}$, then the optimal solution is

$$q_{\mathbf{x}}^{[t+1]}(z) = \mathbf{I}_{\left[z = \arg\max_z p(z|\mathbf{x};\boldsymbol{\Xi}^{[t+1]})\right]}, \forall z \in \mathcal{C}. \tag{9}$$

We call this case *hard-clustering*, e.g., k-means. The superiority of soft-clustering is that in each $\mathbf{E}$-step, the upper bound $\mathcal{G}$ is touched by the original loss $\mathcal{L}$, i.e., $\mathcal{L}(\boldsymbol{\Xi}^{[t]}) = \mathcal{G}(q_{\mathbf{x}}^{[t+1]}(\cdot), \boldsymbol{\Xi}^{[t]})$, while in hard-clustering, this property does not hold. However, when $p(\mathbf{x};\boldsymbol{\Xi})$ is an EFM model, using the Bregman divergence expression (Eq. (2)), hard-clustering (Eq. (9)) is efficient to compute.

In $\mathbf{M}$-step, when $p(\mathbf{x};\boldsymbol{\Xi})$ is an EFM model, simply using Lagrangian method [Beal, 2003], we obtain the closed form of the solution: $\forall z \in \mathcal{C}$,

$$\alpha_z^{[t+1]} = \mathbf{E}_f\left[q_{\mathbf{x}}(z)\right] \tag{10}$$

and

$$\boldsymbol{\mu}_z^{[t+1]} = \nabla_{\boldsymbol{\theta}_z} \Psi \mid_{\boldsymbol{\theta}_z^{[t+1]}} = \frac{\mathbf{E}_f \left[q_{\mathbf{x}}^{[t+1]}(z) T(\mathbf{x}) \right]}{\mathbf{E}_f \left[q_{\mathbf{x}}^{[t+1]}(z) \right]}. \quad (11)$$

Then $\boldsymbol{\theta}_z^{[t+1]}$ can be obtained by Eq. (3). In fact, using Eq. (2), we do not need $\boldsymbol{\theta}_z^{[t+1]}$ in the EM iterations.

Typical examples of clustering via EFMs are GMM clustering and k-means.

3 Frameworks

Now consider the setting of on-line evolutionary clustering. At each time epoch i, new data $\mathbf{X}^{(i)} = (\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_{n_i}^{(i)})$ arrives, and a partition on $\mathbf{X}^{(i)}$ is desired. The underlying distribution is denoted as $F^{(i)}(\mathbf{x})$ (with density $f^{(i)}(\mathbf{x})$). For each epoch, $f^{(i)}$ is approximated by an EFM model $p^{(i)}(\mathbf{x}; \boldsymbol{\Xi}^{(i)}) = \sum_z^{C_i} \alpha_z^{(i)} p(\mathbf{x}; \boldsymbol{\theta}_z^{(i)})$. The expectation parameter of the component $p(\mathbf{x}; \boldsymbol{\theta}_z^{(i)})$ is $\boldsymbol{\mu}_z^{(i)}$. The component numbers C_i need not be the same at different epochs.

Following [Chakrabarti *et al.*, 2006; Chi *et al.*, 2007], an first-order Markovian property is assumed for the evolving behavior. Therefore, we only need to consider adjacent epochs i and $i+1$. From now on, they will be simply denoted as epochs "1" and "2".

From the density estimation view to clustering, the loss of static clustering via an EFM is the divergence between data distribution f and the EFM model distribution $p(\mathbf{x}; \boldsymbol{\Xi})$. In evolutionary setting, the data distributions $f^{(1)}$ and $f^{(2)}$ are assumed close to each other. If model distributions $p^{(1)}$ and $p^{(2)}$ are their good estimates respectively, naturally, current model distribution $p^{(2)}$ should neither deviate much from historical data distribution $f^{(1)}$ nor historical model distribution $p^{(1)}$. This viewpoint results in our two general frameworks for on-line evolutionary EFM: *Historical Data Dependent* (HDD) and *Historical Model Dependent* (HMD).

The general form of loss function for HDD is

$$\mathcal{L}_{hdd} = (1-\lambda)\, dist(f^{(2)}, p^{(2)}) + \lambda\, dist(f^{(1)}, p^{(2)}) \quad (12)$$

where the temporal loss $dist(f^{(1)}, p^{(2)})$ ensures current model distribution $p^{(2)}$ dose not deviate much from historical data distribution $f^{(1)}$.

The general form of loss function for HMD is

$$\mathcal{L}_{hmd} = (1-\lambda)\, dist(f^{(2)}, p^{(2)}) + \lambda\, dist(p^{(1)}, p^{(2)}) \quad (13)$$

where the temporal loss $dist(p^{(1)}, p^{(2)})$ ensures current model distribution $p^{(2)}$ dose not deviate much from historical model distribution $p^{(1)}$.

In both frameworks, parameter λ reflects the preference to historical data/model. The dynamic evaluation of λ will be discussed in Sec. 3.3.

3.1 Historical Data Dependent (HDD)

Since the loss of static clustering via an EFM is the KL divergence between true distribution f and the EFM model distribution $p(\mathbf{x}; \boldsymbol{\Xi})$, we also define the temporal loss as $dist(f^{(1)}, p^{(2)}) = KL(f^{(1)} || p^{(2)})$, then we get the specific form of loss for HDD:

$$\mathcal{L}_{hdd} = (1-\lambda)\, KL(f^{(2)} || p^{(2)}) + \lambda\, KL(f^{(1)} || p^{(2)})$$

With constant item ignored, it can be easily written as

$$\mathcal{L}_{hdd}(\boldsymbol{\Xi}^{(2)}) = -\int \left[(1-\lambda) f^{(2)}(\mathbf{x}) + \lambda f^{(1)}(\mathbf{x}) \right] \log p^{(2)}(\mathbf{x}; \boldsymbol{\Xi}^{(2)}) d\,\mathbf{x}$$

Notice that $(1-\lambda) f^{(2)}(\mathbf{x}) + \lambda f^{(1)}(\mathbf{x})$ induces another distribution, denoted by $\tilde{f}_\lambda(\mathbf{x})$. Then we have

$$\mathcal{L}_{hdd}(\boldsymbol{\Xi}^{(2)}) = -\mathbf{E}_{\tilde{f}_\lambda}[\log p^{(2)}(\mathbf{x}; \boldsymbol{\Xi}^{(2)})] \quad (14)$$

Comparing Eq. (14) with Eq. (5), we can see that HDD is essentially to estimate the density of the deduced distribution $\tilde{f}_\lambda(x)$ using an EFM. The same EM procedure as Eq. (8, 9, 10, 11) can be used:

E-step: $\forall z \in \mathcal{C}$, for soft-clustering

$$q_{\mathbf{x}}^{[t+1]}(z) = p(z | \mathbf{x}; \boldsymbol{\Xi}^{(2),[t]}), \quad (15)$$

and for hard-clustering

$$q_{\mathbf{x}}^{[t+1]}(z) = \mathbf{I}_{\left[z = \arg\max_z p(z | \mathbf{x}; \boldsymbol{\Xi}^{(2),[t]})\right]} \quad (16)$$

M-step: $\forall z \in \mathcal{C}$,

$$\alpha_z^{(2),[t+1]} = \mathbf{E}_{\tilde{f}_\lambda}[q_{\mathbf{x}}^{[t+1]}(z)] \quad (17)$$

$$\boldsymbol{\mu}_z^{(2),[t+1]} = \frac{\mathbf{E}_{\tilde{f}_\lambda}[q_{\mathbf{x}}^{[t+1]}(z) T(\mathbf{x})]}{\mathbf{E}_{\tilde{f}_\lambda}[q_{\mathbf{x}}^{[t+1]}(z)]} \quad (18)$$

where $\mathbf{E}_{\tilde{f}_\lambda}[\cdot] = (1-\lambda)\mathbf{E}_{f^{(2)}}[\cdot] + \lambda \mathbf{E}_{f^{(1)}}[\cdot]$. Notice that, for $i = 1, 2$, $\mathbf{E}_{f^{(i)}}[q_{\mathbf{x}}^{[t+1]}(z)]$ and $\mathbf{E}_{f^{(i)}}[q_{\mathbf{x}}^{[t+1]}(z) T(\mathbf{x})]$ are the estimators of $\alpha_z^{(i)}$ and $\boldsymbol{\mu}_z^{(i)}$ on $f^{(i)}$, respectively. Above result means that, in each M-step, the estimator of parameters $\boldsymbol{\Xi}^{(2)}$ on $f^{(2)}$ is adjusted by the same estimator on $f^{(1)}$, to ensure that the estimated model distribution $p(\mathbf{x}; \boldsymbol{\Xi}^{(2)})$ approximates both $f^{(1)}$ and $f^{(2)}$ well.

3.2 Historical Model Dependent (HMD)

In the framework of Eq. (13), an intuitive choice of $dist(p^{(1)}, p^{(2)})$ is also the KL divergence. However, KL divergence between two EFMs can not be exactly calculated, and approximate sampling methods are needed, which are time exhausted. We seek other divergence measures.

Rather than KL divergence, Earth Mover Distance (EMD) [Rubner *et al.*, 1998] is another divergence measure between two distributions, which is frequently used to measure divergence between mixture models. EMD between two mixture models is defined as:

$$d_{\text{EMD}}(p^{(1)}, p^{(2)}) = \min_{\mathbf{w}} \sum\nolimits_{l,z} w_{lz} d(p(\mathbf{x}; \boldsymbol{\theta}_l^{(1)}), p(\mathbf{x}; \boldsymbol{\theta}_z^{(2)}))$$
$$s.t. \quad w_{lz} \geq 0, \sum\nolimits_z w_{lz} = \alpha_l^{(1)}, \sum\nolimits_l w_{lz} = \alpha_z^{(2)} \quad (19)$$

where $d(p(\mathbf{x}; \boldsymbol{\theta}_l^{(1)}), p(\mathbf{x}; \boldsymbol{\theta}_z^{(2)}))$ is a predefined divergence measure between two components. In this paper, KL divergence is adopted, as KL divergence $KL(p(\mathbf{x}; \boldsymbol{\theta}_l^{(1)}) || p(\mathbf{x}; \boldsymbol{\theta}_z^{(2)}))$ between the two components from a same exponential family has a closed form

$$\Psi(\boldsymbol{\theta}_z^{(2)}) - \Psi(\boldsymbol{\theta}_z^{(1)}) - \left\langle \boldsymbol{\theta}_z^{(2)} - \boldsymbol{\theta}_z^{(1)}, \nabla_{\boldsymbol{\theta}} \Psi \mid_{\boldsymbol{\theta}_z^{(1)}} \right\rangle.$$

The loss function of HMD will be written as

$$\mathcal{L}_{hmd} = (1-\lambda)\, KL(f^{(2)}, p^{(2)}) + \lambda d_{\text{EMD}}(p^{(1)}, p^{(2)}).$$

According to Eq. (19), minimizing $\mathcal{L}_{hmd}$ is equivalent to

$$\min_{\mathbf{\Xi}^{(2)},\mathbf{w}} \mathcal{L}'_{hmd}(\mathbf{\Xi}^{(2)}, \mathbf{w}) = (1-\lambda)\, KL(f^{(2)}(x), p^{(2)}(x;\mathbf{\Xi}^{(2)})) + \lambda \sum\nolimits_{l,z} w_{lz}\, KL(p(x;\boldsymbol{\theta}_l^{(1)}), p(x;\boldsymbol{\theta}_z^{(2)}))$$

$$s.t. \quad w_{lz} \geq 0, \sum_{l}^{C_1} w_{lz} = \alpha_z^{(2)}, \sum_{z}^{C_2} w_{lz} = \alpha_l^{(1)}, \sum_{z}^{C_2} \alpha_z^{(2)} = 1$$

Similar to Eq. (6), $\mathcal{L}'_{hmd}$ has a variational upper bound:

$$\mathcal{L}'_{hmd}(\mathbf{w}, \mathbf{\Xi}^{(2)}) \leq \mathcal{G}(q_{\mathbf{x}}(\cdot), \mathbf{\Xi}^{(2)}, \mathbf{w}) \quad \mathcal{H}(q_{\mathbf{x}}(\cdot)) + \mathcal{E}(q_{\mathbf{x}}(\cdot), \mathbf{\Xi}^{(2)}) + \mathcal{D}(\mathbf{w}, \mathbf{\Xi}^{(2)})$$

where $\mathcal{H}(q_{\mathbf{x}}(\cdot)) = (1-\lambda)\sum_z \mathbf{E}_{f^{(2)}}[q_{\mathbf{x}}(z)\log q_{\mathbf{x}}(z)]$,

$$\mathcal{E}(q_{\mathbf{x}}(\cdot), \mathbf{\Xi}^{(2)}) = -(1-\lambda)\sum\nolimits_z \mathbf{E}_{f^{(2)}}[q_{\mathbf{x}}(z)\log\alpha_z^{(2)}] - (1-\lambda)\sum\nolimits_z \mathbf{E}_{f^{(2)}}[q_{\mathbf{x}}(z)(\langle\boldsymbol{\theta}_z^{(2)}, T(\mathbf{x})\rangle - \Psi(\boldsymbol{\theta}_z^{(2)}))]$$

and $\mathcal{D}(\mathbf{w}, \mathbf{\Xi}^{(2)}) = \sum_{l,z} w_{lz}\, KL(p(\mathbf{x};\boldsymbol{\theta}_l^{(1)}), p(\mathbf{x};\boldsymbol{\theta}_z^{(2)}))$.

Alternative optimization is used to minimize $\mathcal{G}$:

w-step: With $q_{\mathbf{x}}(\cdot)$ and $\mathbf{\Xi}^{(2)}$ fixed, minimize $\mathcal{G}$ w.r.t. $\mathbf{w}$:

$$\mathbf{w}^{[t+1]} = \arg\min_{\mathbf{w}} \mathcal{D}(\mathbf{w}, \mathbf{\Xi}^{(2),[t]}) \quad (20)$$

$$s.t.\ w_{lz} \geq 0, \sum\nolimits_{l}^{C_1} w_{lz} = \alpha_z^{(2),[t]}, \sum\nolimits_{z}^{C_2} w_{lz} = \alpha_l^{(1),[t]},$$

which is just the computation of EMD and can be efficiently solved by linear programming.

q**-step**: With $\mathbf{w}$ and $\mathbf{\Xi}^{(2)}$ fixed, minimize $\mathcal{G}$ w.r.t. $q_{\mathbf{x}}(\cdot)$:

$$q_{\mathbf{x}}^{[t+1]}(\cdot) = \arg\min_{q_{\mathbf{x}}(\cdot)} \mathcal{E}(q_{\mathbf{x}}(\cdot), \mathbf{\Xi}^{(2),[t]}) + \mathcal{H}(q_{\mathbf{x}}(\cdot))$$

The result is identical to Eq. (15,16). The property for soft-clustering still holds here: in each q-step, the upper bound is touched, i.e., $\mathcal{L}'_{hmd}(\mathbf{w}, \mathbf{\Xi}^{(2)}) = \mathcal{G}(q_{\mathbf{x}}^{[t+1]}(\cdot), \mathbf{\Xi}^{(2)}, \mathbf{w})$.

Ξ-step: With $\mathbf{w}$ and $q_{\mathbf{x}}(z)$ fixed, minimize $\mathcal{G}$ w.r.t. $\mathbf{\Xi}^{(2)}$:

$$\mathbf{\Xi}^{(2),[t+1]} = \arg\min_{\mathbf{\Xi}^{(2)}} \mathcal{E}(q_{\mathbf{x}}^{[t+1]}(\cdot), \mathbf{\Xi}^{(2)}) + \mathcal{D}(\mathbf{w}^{[t+1]}, \mathbf{\Xi}^2)$$

$$s.t. \quad \sum\nolimits_z \alpha_z^{(2)} = 1$$

Using Lagrangian method, we can obtain the closed form of the optimal solution for this step: $\forall z \in \mathcal{C}$

$$\alpha_z^{(2),[t+1]} = \mathbf{E}_{f^{(2)}}[q_{\mathbf{x}}^{[t+1]}(z)] \quad (21)$$

$$\boldsymbol{\mu}_z^{(2),[t+1]} = \frac{(1-\lambda)\mathbf{E}_{f^{(2)}}[q_{\mathbf{x}}^{[t+1]}(z)T(\mathbf{x})] + \lambda\sum_l w_{lz}^{[t+1]}\boldsymbol{\mu}_l^{(1)}}{(1-\lambda)\mathbf{E}_{f^{(2)}}[q_{\mathbf{x}}^{[t+1]}(z)] + \lambda\sum_l w_{lz}^{[t+1]}}$$

The result means that, in each $\mathbf{\Xi}$-step, the estimators of expectation parameters $\boldsymbol{\mu}_z^{(2)}$ on current data distribution $f^{(2)}$ are directly adjusted by the estimators $\boldsymbol{\mu}_l^{(1)}$ of last epoch.

In fact, the evolutionary k-means in [Chakrabarti *et al.*, 2006] is a special case of HMD with approximately computing of d_{EMD} in w-step. The EFM used in k-means is the mixture of spherical Gaussians with identical constant variance σ^2 and prior $\frac{1}{C}$: $p(\mathbf{x};\mathbf{\Xi}) = \frac{1}{C}\sum_z \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_z,\sigma^2 I)$. Then the objective function in Eq. (20) is $\mathcal{D} = \frac{1}{2\sigma^2}\min_{\mathbf{w}}\sum_{l,z} w_{lz}\|\boldsymbol{\mu}_l^{(1)} - \boldsymbol{\mu}_z^{(2)}\|^2$, with the same constraints on $\mathbf{w}$ as those in Eq.(19, 20). [Chakrabarti *et al.*, 2006] approximate $\mathcal{D}$ by $\sum_z \|\boldsymbol{\mu}_z^{(2)} - \boldsymbol{\mu}_{g(z)}^{(1)}\|$, where $g(z) = \arg\min_l \|\boldsymbol{\mu}_z^{(2)} - \boldsymbol{\mu}_l^{(1)}\|$, which means, they assigned each current component to the nearest center at last epoch, then summed the distances between them as the divergence between two mixtures. Based on HMD, we can extend the approach of [Chakrabarti *et al.*, 2006] to all the EFMs, resulting in the approximate HMD algorithm, which is different from HMD in $\mathbf{w}$-step (Eq. (20)):

$$w_{lz}^{[t+1]} = \alpha_z^{(2),[t]} \cdot \mathbf{I}_{[l=\arg\min_l KL(p(\mathbf{x};\boldsymbol{\theta}_l^{(1)}), p(\mathbf{x};\boldsymbol{\theta}_z^{(2)}))]} \quad (22)$$

However, as pointed out in [Chi *et al.*, 2007], this approach could be unstable, i.e., sensitive to small perturbation on the centers.

In both frameworks, the assumption is that epoch data are arbitrary I.I.D. samples from the corresponding epoch distribution, accordingly, the data sizes of different epochs need not be the same. Additionally, in both frameworks, cluster numbers C_i of different epochs are not assumed to be the same, consequently, both frameworks are able to deal with the variation of cluster number. Moreover, using different specific exponential families, both frameworks can produce a large family of evolutionary clustering algorithms.

3.3 Dynamic evaluation of λ

Parameter λ reflects the preference to historical data/model, which should be determined by the dependency between the adjacent data distributions. If current data distribution deviates much from the historical, then the impact of historical data/model should be suppressed. A mechanism for dynamic evaluation of λ is required. However, this problem has not been studied in previous works [Chakrabarti *et al.*, 2006; Chi *et al.*, 2007; Tang *et al.*, 2008].

[Gretton *et al.*, 2007] proposed a non-parametric test statistic to check the dependency between two distributions based on two sets of i.i.d samples. The empirical estimation of the test statistic is $\tau(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = \frac{1}{n_1^2}\sum_{i,j}^{n_1} k(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)}) - \frac{2}{n_1 n_2}\sum_{i,j}^{n_1,n_2} k(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(2)}) + \frac{1}{n_2^2}\sum_{i,j}^{n_2} k(\mathbf{x}_i^{(2)}, \mathbf{x}_j^{(2)})$, where $k(\cdot,\cdot)$ is an universal kernel. In this paper, the RBF kernel $k(x,y) = \exp\{-\frac{\|x-y\|^2}{\sigma^2}\}$ is used. In fact, τ measures the discrepancy between the two distributions.

Using the test statistic, we can evaluate $\lambda^{(i)}$ as:

$$\lambda^{(i)} = \lambda_0 \exp\{-\beta \cdot \tau(\mathbf{X}^{(i)}, \mathbf{X}^{(i-1)})\} \quad (23)$$

where $\lambda_0 \in [0,1]$ reflects a basic preference to historical data/model, and β reflects the sensitive to variation of the test statistic.

3.4 Comparisons between HDD and HMD

Now we give a comparison analysis to the two frameworks. HDD is efficient in computing and allows users to change the component family $\mathcal{F}_\Psi$ in the EFM model, e.g., from GMM to MMM, while HMD does not allow that. What's more, HMD needs more time to compute EMD, especially when cluster numbers are large. However, in general, if the basic assumption holds that $f^{(1)}$ and $f^{(2)}$ are close, HMD will perform

better than HDD in preserving the smoothness of clustering results, which will be explained below.

When statically clustering via an EFM, estimation of $p^{(2)}$ on $f^{(2)}$ produces a solution p_0 stuck in a local minimum of loss $dist(f^{(2)}, p^{(2)})$, as illustrated in Fig. 1 (a.2, b.2).

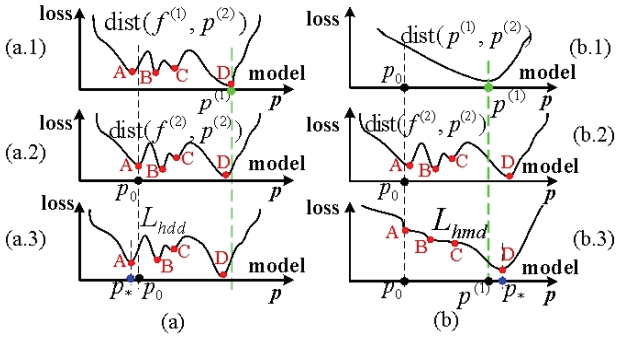


Figure 1: Comparison between HDD and HMD

In HDD, as $f^{(1)}$ and $f^{(2)}$ are assumed close to each other, static loss $dist(f^{(2)}, p^{(2)})$ and temporal loss $dist(f^{(1)}, p^{(2)})$ are also close to each other (Fig.1 (a.1, a.2)). Then the overall loss $\mathcal{L}_{hdd}$, the weighted sum of the two losses, is also close to them (Fig.1 (a.3)). Then the candidate solution p_0 in static setting (Fig.1 (a.2)) will still be stuck in a nearby local minimum (p_*) in evolutionary setting (Fig.1 (a.3)).

In HMD (Fig.1 (b)), the candidate solution p_0 in static setting can be heavily penalized by the large loss resulting from a large deviation from $p^{(1)}$ (the large loss $dist(p^{(1)}, p_0)$) in evolutionary setting. The penalty can drag out the solution from p_0 and push it toward another minimum more close to $p^{(1)}$.

4 Experiments

We demonstrate the validation of the two frameworks by experiments on three typical clustering algorithms based on EFMs, i.e., GMM, k-means, and multinomial mixture model (MMM). Evolutionary GMM is tested on a synthetic data designed by ourselves. Evolutionary k-means and evolutionary MMM are tested on a real text data set.

4.1 Data sets

The GMM data set are samples from an evolving 2D GMM model with noise. In 20 epochs, all the parameters of GMM model are slowly evolving. Data size is also varying. Five epochs and the overall data are illustrated in Fig. 2. This data set will be used to provide experiential evidences to the comparison analysis in Sec. 3.4. We will also demonstrate the necessity of dynamic evaluation of λ on this data set.

The real data set is "NSF Research Awards Abstracts" [2], which consists of the abstracts describing NSF awards for basic research, covering 14 years from 1990 to 2003. We extract the field "NSF program" indicating the research area as the class label. A subset containing the top 10 classes covering 13 years (1990-2002) is selected as our experimental data. This subset consists of 19,728 documents with 15,412 identical words. There are 10 classes in the first 9 years, and 9 classes in the following 4 years. For evolutionary k-means, the *tf-idf* feature is used, while for evolutionary MMM, the word count feature is used.

[2]http://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html

4.2 Algorithms

Besides HDD and HMD, another two algorithms are considered: first, the static baseline, i.e. clustering via an EFM independently at each epoch, denoted as IND; second, the approximate HMD as Eq. (22), denoted as APP-HMD. In evolutionary k-means, the APP-HMD reduces to the algorithm of [Chakrabarti *et al.*, 2006].

We cannot compare with PCQ and PCM of [Chi *et al.*, 2007], as they cannot deal with the case when epoch data are arbitrary I.I.D. samples from epoch distributions, as pointed out in Sec. 1.

Besides the four algorithms, to illustrate the necessity of dynamic evaluating λ, we also run HDD and HMD with static λ on GMM data, which will be denoted as HDD-S and HMD-S, respectively.

4.3 Criterions

The clustering quality at each epoch will be measured by *Normalized Mutual Information* (NMI), which is a widely used criterion for clustering. High value on NMI reflects good consistency with the true class label. The temporal smoothness of clustering results will be measured by the two types of temporal loss: $KL(f^{(1)}||p^{(2)})$ and $d_{\text{EMD}}(p^{(1)}, p^{(2)})$. They will be called *data measured temporal loss* (DTL) and *model measured temporal loss* (MTL), respectively. Low DTL and MTL reflect good smoothness of clustering results along time.

4.4 Methodology

For all algorithms, the data epochs are traversed through by N (=50) times. In one traverse, at each epoch, an identical initialization generated randomly is imposed to all algorithms. Then the criterions, e.g., NMI, DTL, MTL are calculated at each epoch. The mean and standard deviation of the criterions at each epoch are calculated across the N runs.

On GMM data, we demonstrate the necessity of dynamic evaluation of λ by another experiment: we replace the 16th epoch with the 4th epoch, which makes the epoch distribution change abruptly at the 16th and 17th epochs. We run HDD-S and HMD-S and compare the performance of them with that of HDD and HMD.

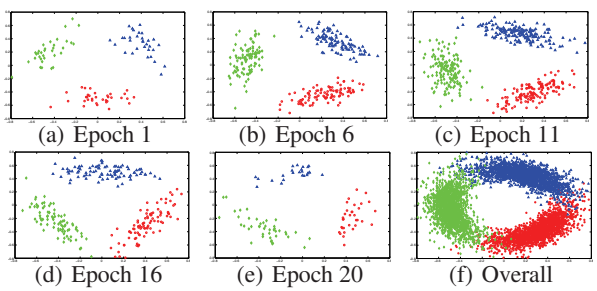
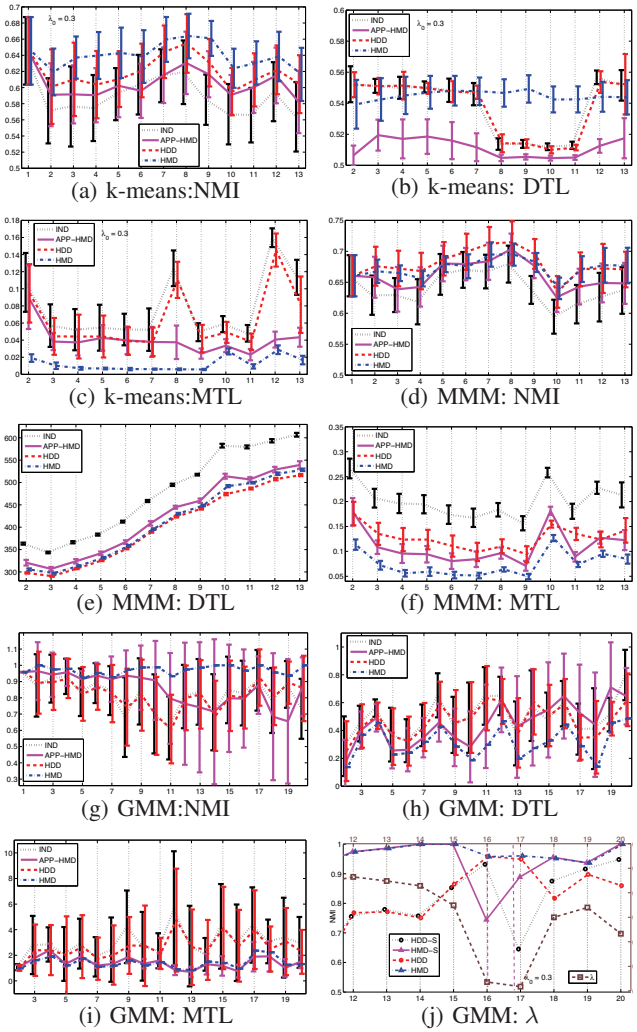


Figure 2: Synthetic GMM data set



(a) k-means:NMI (b) k-means: DTL (c) k-means:MTL (d) MMM: NMI (e) MMM: DTL (f) MMM: MTL (g) GMM:NMI (h) GMM: DTL (i) GMM: MTL (j) GMM: λ

Figure 3: Results of experiments

4.5 Results

The results are illustrated in Fig. (3). In general, compared to IND, HDD and HMD enhance the clustering quality at each epoch, i.e., higher mean value and much lower deviation value on NMI (Fig.3(a), 3(d), 3(g)), meanwhile, HDD and HMD better preserve the smoothness of clustering results along time, i.e., much lower mean and deviation value on DTL (Fig. 3(e), 3(b), 3(h)) and MTL (Fig. 3(f), 3(c), 3(i)). APP-HMD does not perform well, the results on NMI are even worse than those of IND on GMM data.

Results on GMM data (Fig.3(g),3(h),3(i)) provide evidences to our analysis in Sec.3.4. Due to local optima, IND gives results with large deviation (large value on deviation of NMI, mean of DTL and mean of MTL). HDD is easy to be stuck by the same local optima, resulting in the almost same result as that of IND. HMD gives the best performance.

Fig. 3(j) demonstrates the necessity of dynamic evaluation of λ: at 16th and 17th epochs, due to the abrupt change of data distribution, with static $\lambda(=\lambda_0)$, the historical data/model harms current clustering. However, dynamic evaluation of λ as Eq. (23) gives rather low value of λ at the two epochs, suppressing the impact of historical data/model. For clarity, only mean values on NMI are plot in Fig. 3(j).

5 Conclusion

We deal with the problem of evolutionary clustering where distribution of data evolves along time. We propose two general density estimation based online frameworks. They give uniform evolutionary solutions to all the conventional clustering algorithms based on EFMs.

Acknowledgments

This research was supported by National Science Foundation of China (Grant No. 60835002 and 60721003).

References

[Ahmed and Xing, 2008] A. Ahmed and E. Xing. Dynamic non-parametric mixture models and the recurrent Chinese restaurant process: with applications to evolutionary clustering. *SDM*, 2008.

[Banerjee *et al.*, 2005] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with Bregman divergences. *JMLR*, 6:1705–1749, 2005.

[Beal, 2003] M.J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. Thesis for degree of doctor of philosophy, University of London, 2003.

[Chakrabarti *et al.*, 2006] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. *KDD*, 2006.

[Charikar *et al.*, 1997] M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. *ACM Symposium on Theory of Computing*, 1997.

[Chi *et al.*, 2007] Y. Chi, X.-D. Song, D.-Y. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. *KDD*, 2007.

[Gretton *et al.*, 2007] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel approach to comparing distributions. *AAAI*, 2007.

[Hartigan and Wong, 1979] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[Ng *et al.*, 2002] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 2002.

[Rubner *et al.*, 1998] Y. Rubner, C. Tomasi, and LJ Guibas. A metric for distributions with applications to image databases. *ICCV*, 1998.

[Tang *et al.*, 2008] L. Tang, H. Liu, J.-P. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. *KDD*, 2008.

[Vapnik, 2000] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2000.

[Wang *et al.*, 2007] Y. Wang, S.-X. Liu, L.-Z. Zhou, and H. Su. Mining naturally smooth evolution of clusters from dynamic data. *SDM*, 2007.