Visualizing Global Manifold Based on Distributed Local Data Abstractions

Xiaofeng Zhang
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
xfzhang@comp.hkbu.edu.hk

William K. Cheung
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
william@comp.hkbu.edu.hk

Abstract

Mining distributed data for global knowledge is getting more attention recently. The problem is especially challenging when data sharing is prohibited due to local constraints like limited bandwidth and data privacy. In this paper, we investigate how to derive the embedded manifold (as a 2-D map) for a horizontally partitioned data set, where data cannot be shared among the partitions directly. We propose a model-based approach which computes hierarchical local data abstractions, aggregates the abstractions, and finally learns a global generative model – generative topographic mapping (GTM) based on the aggregated data abstraction. We applied the proposed method to two benchmarking data sets and demonstrated that the accuracy of the derived manifold can effectively be controlled by adjusting the data granularity level of the adopted local abstraction.

1. Introduction

Recent progress in automatic data collection, data storage and networking technologies has resulted in high accessability of distributed and massive data for application domains like e-Science and e-Commerce. This becomes important for data mining techniques, if needed, to be applied in a distributed environment. Distributed data mining is challenging as data sharing is in many cases prohibited due to local constraints like limited bandwidth and data privacy. The former constraint is faced as the distributed data can be of high volume, e.g., in e-Science. The latter one happens when the local data owners indicate high privacy concern while they still prefer some degree of personalized e-services, like the situations in e-Commerce.

To avoid sharing data directly, one possible approach is to adopt some flexible statistical model for abstracting the local data so that the local data granularity (or the local privacy level from the data privacy perspective) can be controlled. A model with high complexity usually can retain more details when compared with one of low complexity. The use of the model-based approach for distributed data mining can be found in [4, 3, 5].

In particular, Zhang *et al.* demonstrated in [5] how a global cluster model can be learned based on local data abstractions. In this paper, this distributed model-based approach was extended to visualizing the embedded manifold of a set of distributed data. Generative topographic mapping (GTM) which is an effective nonlinear mapping tool for visualizing high dimensional data was chosen to be the global model, and Gaussian mixture model (GMM) [2] was chosen for local data abstraction due to its representation flexibility. To learn the global GTM, we proposed a modified EM-like algorithm for learning directly from the aggregated local data abstraction. The experimental results obtained based on two benchmarking data sets demonstrated that the proposed distributed learning approach can achieve comparably good visualization results and at the same time satisfy the limited bandwidth and data privacy requirements of the local sources in a controlled manner.

2 Problem Formulation

2.1 Local Data Abstraction

Local data abstraction is here defined as the process of representing a given set of data by its statistics forming an abstraction. Via the abstractions, the statistical information of the data can be shared, instead of the data themselves. The abstraction process is formulated as a parametric density estimation problem and a hierarchical GMM with different numbers of components at different levels of the hierarchy is adopted to support sharing local data details at different data granularity levels.

Assume that there are totally L distributed data sources. Let $t_i \in \Re^d$ denote the i^{th} observed data item of dimension d, θ_l denote the set of parameters of the local model (GMM) as the abstraction of the l^{th} source, θ_{lj} denote the

j^{th} component's parameters of the l^{th} local model (including the component's mean μ_{lj} and covariance matrix Σ_{lj}), α_{lj} denote the mixing proportion of the j^{th} component in the l^{th} local model. The probability density function of the l^{th} local model $p_{local}(t_i|\theta_l)$ with K_l components is given as,

$$p_{local}(t_i|\theta_l) = \sum_{j=1}^{K_l} \alpha_{lj} p_j(t_i|\theta_{lj})$$
$$\sum_{j=1}^{K_l} \alpha_{lj} = 1$$
$$p_j(t_i|\theta_{lj}) = (2\pi)^{-\frac{d}{2}} |\Sigma_{lj}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(t_i - \mu_{lj})^T \Sigma_{lj}^{-1}(t_i - \mu_{lj})\}.$$

The local GMM parameters, *i.e.*, $\{\theta_1, \theta_2, ..., \theta_L\}$, are first derived as the abstractions of the distributed local data by applying the agglomerative hierarchical algorithm (AGH). Given the dendrogram, illustrated in Figure 1, a hierarchy of GMMs with different number of components for representing the local data at different granularity levels can easily be computed. In particular, at a particular data granularity level, a local Gaussian component can be derived by computing the mean and covariance matrix of the data within a group at that level. Then the local GMM parameters can be sent to a global server for learning a global data model. In principle, the global model can be any type of generative model.

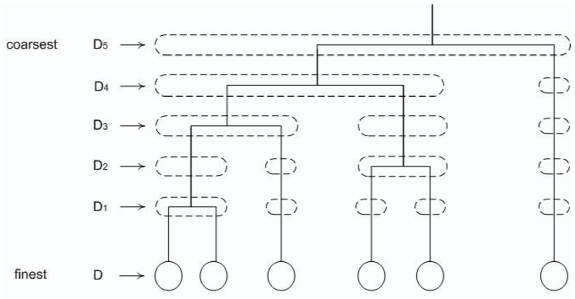


Figure 1. A hierarchy of data abstractions. A higher level of abstraction is acquired by merging and computing the statistics of the two nearest data subgroups at the next lower level.

2.2 Learning A Global GTM

Generative topographic mapping (GTM) [1] is a probabilistic non-linear latent variable model which can be used to explore the embedded manifold of a set of high-dimensional data. GTM assumes that the data are generated due to a lattice of latent variables in a low-dimensional (usually 2D) latent space. Via a non-linear mapping, a point in the *latent* space is mapped to a data item in the *data* space. Visualizing the latent space with the original high-dimensional data projected back to it can result in an "unfolded" version of the embedded manifold which is useful for understanding the structure and organization of the data.

2.2.1 GTM Formulation

Let N denote the total number of data items, $z_k \in \Re^H$ denote the k^{th} lattice point (altogether M) defined in the latent space. $y(z; W) := W\Psi(z)$ maps in an non-linear fashion a point z in the latent space onto a corresponding point y in the data space, with the mapping governed by a generalized linear regression model Ψ weighted by W. A multivariate Gaussian distribution in the data space is assumed in GTM for t_i given z_k, given as

$$p(t_i|z_k, W, \beta) = (2\pi)^{-\frac{d}{2}} \beta^{\frac{d}{2}} \exp\{-\frac{\beta}{2}\|(t_i - y(z_k; W)\|^2\} \quad (1)$$

where β is the reciprocal of the data variance.

The EM algorithm is typically used for estimating the parameters W and β. The E-step for the original GTM is given as

$$R_{ik}(W_{old}, \beta_{old}) = P(z_k|t_i, W, \beta) = \frac{p(t_i|z_k, W_{old}, \beta_{old})}{\sum_{j=1}^{M} p(t_i|z_j, W_{old}, \beta_{old})}$$

and the M-step is given as

$$\sum_{i=1}^{N}\sum_{k=1}^{M} R_{ik}(W_{old}, \beta_{old})\{W_{new}\Psi(z_k) - t_i\}\Psi(z_k)^T = 0 \quad (2)$$

$$\frac{1}{\beta_{new}} = \frac{1}{Nd}\sum_{i=1}^{N}\sum_{k=1}^{M} R_{ik}(W_{old}, \beta_{old})\|W_{new}\Psi(z_k) - t_i\|^2. \quad (3)$$

2.2.2 Learning from Local Data Abstraction

In order to learn the global GTM model parameters directly from the local GMM parameters, we first approximate the original estimated indicators R_{ik} by a uniform distribution over the data items corresponding to a particular GMM component which is now to be shared instead of the data.

Assume that R_{lk} is now an indicator for the l^{th} local component[1] with its underlying data to be generated by the k^{th} global component. That is, the likelihood of the subset of the data generated by the k^{th} component of the global model is assumed to be approximated by an overall estimate of the corresponding l^{th} local component being generated by the same component of the global model. R_{ik} can then be approximated as

$$R_{ik} \approx \frac{\sum_{i \in l^{th}\, source} R_{ik}}{N_l} \quad (4)$$

where N_l denotes the number of data from the l^{th} source. R_{lk} is hence equivalent to

$$R_{lk} = \sum_{i \in l^{th}\, source} R_{ik}. \quad (5)$$

To estimate R_{lk}, the formulation adopted is given as

[1]Note that in the subsequent derivation, we abuse the index "l" to refer to one of the local components of the aggregated local model.

$$R_{lk} = \frac{\exp\{-D(p_{local}(t|\theta_l)||p_{gtm}(t|z_k, W, \beta))\}}{\sum_{j=1}^{M} \exp\{-D(p_{local}(t|\theta_l)||p_{gtm}(t|z_j, W, \beta))\}} \quad (6)$$

where the Kullback Leibler (KL) divergence between a local component and a global component, denoted as $D(p_{local}||p_{gtm})$, can be derived as[2]

$$\ln \frac{\beta^{-\frac{d}{2}}}{|\Sigma_l|^{\frac{1}{2}}} + \frac{\beta}{2} tr(\Sigma_l) + \frac{1}{2}(\beta(||y(z_k; W) - \mu_l||^2 - d). \quad (7)$$

For the extreme case that one local GMM is used to represent one data item, the first two terms of Eq.(7) will become constant with respect to the local data. Thus, only the third term will be in effect and Eq.(7) degenerates back to the original GTM's E-step (Eq.(1)). Accordingly, the new M-step can be derived as

$$\sum_{l=1}^{L}\sum_{k=1}^{M} R_{lk}(W_{old}, \beta_{old})\{W_{new}\Psi(z_k) - \mu_l\}\Psi(z_k)^T = 0 \quad (8)$$

$$\begin{aligned} \frac{1}{\beta_{new}} &= \frac{1}{Nd}\sum_{k=1}^{M}(\sum_{l=1}^{L} R_{lk}(W_{old}, \beta_{old})(\Sigma_l + \mu_l\mu_l^T)) \\ &- \frac{1}{Nd}\sum_{k=1}^{M}((W_{new}\Psi(z_k))^2 \sum_{l=1}^{L} R_{lk}). \end{aligned} \quad (9)$$

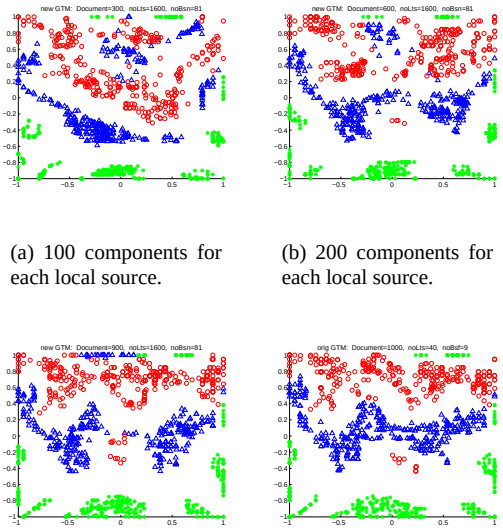
2.2.3 GTM Initialization Based on Local Abstractions

Given the aggregated local model, the initialization of the global GTM can be obtained as equivalent to that of the original GTM. Original GTM uses principle component analysis (PCA) for initializing β and W. For the proposed method, the original data are lacking for computing the global data covariance matrix, and thus the PCA. Fortunately, one can easily show that the global covariance matrix can analytically be derived based on the covariance matrices of the local data, given as

$$\begin{aligned} \mu_{global} &= \frac{\sum_{l=1}^{L} N_l\mu_l}{N} \\ \Sigma_{global} &= \frac{\sum_{l=1}^{L} N_l(\Sigma_l + \mu_l\mu_l^T)}{N} - \mu_{global}\mu_{global}^T. \end{aligned}$$

3 Experiments on Visualizing Distributed Data

To evaluate the effectiveness of the proposed approach of visualizing distributed data using GTM, experiments were performed based on two synthetic datasets — oil flow data and S-curve data for benchmarking. In each experiment, the data set was first horizontally partitioned in a random manner into three equal parts as local data sources. Then, global GTMs were to be learned under different settings for comparison. Both the original GTM learned directly from the original dataset and the new GTM learned from the aggregated local model were tested. For all the experiments, 1600 latent lattice points were chosen as the global GTM parameters.



(a) 100 components for each local source.

(b) 200 components for each local source.

(c) 300 components for each local source.

(d) The original GTM.

Figure 2. The visualization of the oil flow data using GTMs. The posterior means of the projected data of the three different configurations, namely *homogeneous*, *annular* and *stratified*, are labelled as red circles, blue triangles and green asterisks, respectively. Their posterior modes are all shown as crosses.

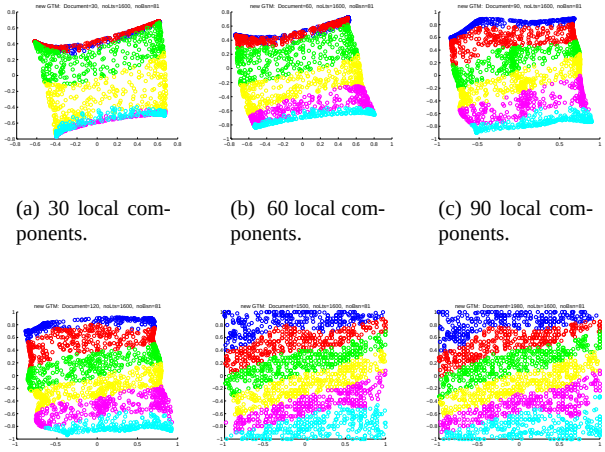
The oil flow dataset was originally used in [1] for mimicking the measurements of oil flows mixed with gas and water along multi-phase pipelines. The 12-dimensional data set consists of 1000 instances evenly distributed among three different geometrical configurations. In the related experiments, 100, 200 and 300 local components were tested for the abstraction of each local data source. We expect that if each local data item is to be represented by one local Gaussian component (the extreme case), the performance of the proposed approach will be equivalent to that of the original GTM. If less local components are assumed, the visualization results will start to degrade.

The visualization results obtained for the oil flow data where shown in Figure 2. It was observed that the visualization map using 300 local components for each source was

[2] Due to the space limitation, detailed derivation of Eq. (7) is omitted.

comparable to that of the original GTM, as shown in Figure 2(d). The visualization result degraded gracefully when the number of local components was dropped to 200 and then to 100. This is consistent to what being anticipated.

S-Curve data another commonly used benchmarking dataset for testing nonlinear manifold learning algorithms. It is in a 3-D data space of which the embedded manifold's shape of 2-D and is like the shape of alphabet 'S'. For the ease of visualization, the 2000 data items in the dataset were labelled. We first divided the dataset into six continuous parts along the data embedded 2-D manifold and were then labelled as blue, red, green, yellow, magenta and cyan circles respectively. 30, 60, 90, 120 and 1500 local components were chosen for the abstractions of each local source. The corresponding visualization results were shown in Figure 3.



(a) 30 local components. (b) 60 local components. (c) 90 local components.

(d) 120 local components. (e) 1500 local components. (f) The original GTM.

Figure 3. The visualization of the S-Curve data which was partitioned into three equally weighted distributed local sources for this experiment.

Figure 3(f) reveals the unfolded manifold obtained using the original GTM learned directly from the data. Those obtained using the proposed GTM based on different numbers of local components are shown in Figure 3(a-e). The unfolded manifold in Figure 3(a) was obtained with only 30 components per local source and found to be the worst when compared with the others using more local components. In particular, in the top region of the map, it can been seen that the blue circles tangled up with the red ones which means that it failed to unfold the top part of the original S-curve data well. A similar situation was observed for the bottom part. In Figure 3(b-d), the aforementioned two unfolded areas, *i.e.* the top and bottom parts, started to be unfolded and were finally completely unfolded as the number of local components per source was increased from 30 to 60, 90 and 120. When the number of local components was close to the number of data items, as shown in Figure 3(e), the visualization results was found to be almost equivalent to that of the original one shown in Figure 3(f).

4 Conclusions

In this paper, we proposed the use of the model-based approach for visualizing distributed data with the constraint that the distributed local data cannot be shared directly. Gaussian mixture models (GMM) was adopted for local data abstraction and generative topological mapping (GTM) was chosen as the global model for high-dimensional data visualization. A novel EM-like algorithm was proposed for learning the global GTM solely based on the aggregated local GMM. The effectiveness of the proposed method was rigorously evaluated using a number of datasets with promising results. Gracefully degrading global visualization results were obtained as the granularity level of the local data became coarser. We believe that the positive results obtained and the formulation we introduced in this paper hint the potential of the proposed method to be a principled way of mining highly distributed high-dimensional data in a distributed environment with limited bandwidth or high data privacy concern.

5 Acknowledgement

This work is jointly supported by RGC Central Allocation Research Grant (HKBU 2/03/C) and Hong Kong Baptist University FRG Grant (FRG/05-06/I-16).

References

[1] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–235, 1998.

[2] G. J.McLachlan and K. E. Basford. *Mixture Models - Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[3] M. Klusch, S. Lodi, and G. L. Moro. Distributed Clustering Based on Sampling Local Density Estimates. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 485–490, Mexico, August 2003.

[4] S. Merugu and J. Ghosh. Privacy-preserving Distributed Clustering using Generative Models. In *The Third IEEE International Conference on Data Mining (ICDM'03)*, Melbourne, FL, November 2003.

[5] X. Zhang and W. K. Cheung. Learning Global Models Based on Distributed Data Abstractions. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2005)*, Edinburgh, August 2005.