Distributed Density Estimation Using Non-parametric Statistics

Yusuo Hu
Microsoft Research Asia
No.49, Zhichun Road, Beijing 100080, China
Yusuo.Hu@microsoft.com

Hua Chen [1]
Tsinghua University,
Beijing 100084, China
chh00@mails.tsinghua.edu.cn

Jian-guang Lou
Microsoft Research Asia
No.49, Zhichun Road, Beijing 100080, China
jlou@microsoft.com

Jiang Li
Microsoft Research Asia
No.49, Zhichun Road, Beijing 100080, China
jiangli@microsoft.com

Abstract

Learning the underlying model from distributed data is often useful for many distributed systems. In this paper, we study the problem of learning a non-parametric model from distributed observations. We propose a gossip-based distributed kernel density estimation algorithm and analyze the convergence and consistency of the estimation process. Furthermore, we extend our algorithm to distributed systems under communication and storage constraints by introducing a fast and efficient data reduction algorithm. Experiments show that our algorithm can estimate underlying density distribution accurately and robustly with only small communication and storage overhead.

Keywords *Kernel Density Estimation, Non-parametric Statistics, Distributed Estimation, Data Reduction, Gossip*

1 Introduction

With the great advance of networking technology, many distributed systems such as peer-to-peer (P2P) networks, computing grids, sensor networks have been deployed in a wide variety of environments. As the scales of these systems grow, there is an increasing requirement for efficient methods to deal with large amounts of data that are distributed over a set of nodes. Particularly, in many applications, it is often required to learn a global distribution from scattered measurements. For example, in a sensor network, we may need to know the distribution of some variable in a target area. In a P2P system, we may want to learn the global distribution of resources for indexing or load balancing. In a distributed agent system, each agent may need to know some global information of the environment through collaborative learning and make decisions based on the current knowledge.

In this paper, we focus on the distributed density estimation problem, which can be described as follows: Given a network consisting of N nodes, where each node holds some local measurements of a hidden random variable $\mathbf{X}$, the task is to estimate the *global* unknown probability density function (pdf) $f(\mathbf{x})$ from all the observed measurements on each node.

Recently, some distributed density estimation algorithms based on parametric models have been proposed [12,14,18]. In these approaches, the unknown distribution is modeled as a mixture of Gaussians, and the parameters are estimated by some distributed implementations of the Expectation Maximization (EM) algorithm. However, we argue that the parametric model is not always suitable for distributed learning. It often needs some strong prior information of the global distribution such as the number of components and the form of the distribution. Furthermore, the EM algorithm is highly sensitive to the initial parameters. With a bad initialization, it may require many steps to converge, or get trapped into some local maxima. Therefore, for most distributed systems, a general and robust approach for distributed density estimation is still needed.

Non-parametric statistical methods have been proven robust and efficient for many practical applications. One of the most used nonparametric techniques is the Kernel Density Estimation (KDE) [23], which can estimate arbitrary distribution from empirical data without much prior knowledge. However, since KDE is a data-driven approach, for distributed systems, we need an efficient mechanism to broadcast data samples. Furthermore, to incrementally learn the nonparametric model from the distributed mea-

[1]The work presented in this paper was carried out at Microsoft Research Asia.

27th International Conference on Distributed Computing Systems (ICDCS'07)
0-7695-2837-3/07 $20.00 © 2007 IEEE

surements, the nodes need to frequently exchange their current local estimates. Thus, a compact and efficient representation of the local estimate on the node is also needed.

In this paper, we propose a gossip-based distributed kernel density estimation algorithm to estimate the unknown distribution from data. We also extend the algorithm to handle the case where the communication and storage resources are constrained. Through theoretical analysis and extensive experiments, we show the proposed algorithm is flexible, and applicable to arbitrary distributions. Compared with the distributed EM algorithm, such as Newcast EM [14], we find that our distributed estimation method based on non-parametric statistics is much more robust and accurate.

Our main contributions are as follows:

- We propose a distributed non-parametric density estimation algorithm based on gossip-based protocol. To the best of our knowledge, it is the first effort to generalize the KDE to the distributed case.

- We prove that the distributed estimate is asymptotically consistent with the global KDE when there is no resource constraint. We also provide rigorous analysis on the convergence speed of the distributed estimation protocol.

- We propose a practical distributed density estimation for situations of limited storage and communication bandwidth. Experiments show that our distributed protocol can estimate the global distribution quickly and accurately.

This paper is organized as follows: First, in Section 2, we briefly review some related work on distributed density estimation methods. We then discuss the problem of the distributed density estimation and present our solution in Section 3. The experimental results of our algorithm are presented in Section 4. The conclusion is given in Section 5.

2 Related Work

Recently, distributed estimation has raised interest in many practical systems. Nowak [18] uses the mixture of Gaussians to model the measurements on the nodes, and proposes a distributed EM algorithm to estimate the means and variances of the Gaussians. Later in [14], Kowalczyk et. al. propose a gossip-based implementation of the distributed EM algorithm protocol. Jiang et. al. [12] apply similar parametric model and distributed EM algorithm in sensor network and use multi-path routing to improve the estimation resilience to link and node failure. However, the EM-based algorithms depend on proper initialization of the number and parameters of the Gaussian components. To alleviate this issue, in [24], a distributed greedy learning algorithm is proposed to incrementally estimate the components of the Gaussian Mixture. However, it requires much more time to learn the model in this way and there is still possibility of being trapped into local minimum because of the greedy nature of the EM algorithm.

Some other research work focuses on extracting the true signals from noisy observations in sensor networks. Ribeiro et al. [20, 21] studied the problem of bandwidth-constrained distributed mean-location parameter estimation in additive Gaussian or non-Gaussian noises. Delouille et. al. [4] used the graphical model to describe the measurements of the sensor networks and proposed an iterative distributed algorithm for linear minimum mean-squared-error (LMMSE) estimation of the mean-location parameters. These approaches put more effort on exploiting the correlations between sensor measurements and eliminating the noise in observations, while we aim to discover the completely unknown pattern from distributed measurements without much assumption or dependency.

Nonparametric models [22, 23], which do not rely on the assumption of the underlying data distribution, are quite suitable for unsupervised learning. However, there is little research on exploiting the usage of nonparametric statistics for distributed systems, except that a few papers have addressed the decentralized classification and clustering problems using kernel methods [13, 17]. To the best of our knowledge, our work is the first effort to generalize the kernel density estimation to the distributed case. We believe that the distributed estimation technique is a useful building block for many distributed systems, and non-parametric methods will play a more important role in distributed systems.

3 Distributed Non-parametric Density Estimation

In this section, we first review some definitions about kernel density estimation, and then present our distributed estimation algorithm. We prove that the estimation process unbiasly converges to the global KDE if each node has enough storage space and communication bandwidth. Our analysis shows that the estimation error decreases exponentially as the number of gossip cycle increases. Based on the basic distributed algorithm, we then extend it to a more flexible version with constrained resource usage by introducing an efficient data reduction mechanism.

3.1 Kernel Density Estimation

Kernel Density Estimation (KDE) is a widely used non-parametric method for density estimation. Given a set of

i.i.d data samples $\{\mathbf{x}_i, i = 1, \ldots, N, \mathbf{x}_i \in R^d, \}$ from some unknown distribution f, the kernel density estimation of f is defined as [23]

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} w_i K_{\mathbf{H}_i}(\mathbf{x} - \mathbf{x}_i), \tag{1}$$

where

$$K_{\mathbf{H}_i}(\mathbf{x} - \mathbf{x}_i) = |\mathbf{H}_i|^{-\frac{1}{2}} K(\mathbf{H}_i^{-\frac{1}{2}}(\mathbf{x} - \mathbf{x}_i)) \tag{2}$$

is the i-th *kernel* located at $\mathbf{x}_i$. Here, $\mathbf{H}_i$ is named *bandwidth matrix*, which is symmetric and positive. w_i, $i = 1, \ldots, N$ are sample weights satisfying

$$\sum_{i=1}^{N} w_i = 1. \tag{3}$$

The kernel function $K(\cdot)$ is a d-variate non-negative, symmetric real-value function. In this paper, we use the following Gaussian kernel due to its simplicity.

$$K(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \exp(-\frac{1}{2}\mathbf{x}^T\mathbf{x}). \tag{4}$$

It has been proved that, given enough data samples, the Kernel Density Estimate (1) can approximate any arbitrary distribution with any given precision [22], i.e. KDE is asymptotically consistent with the underlying distribution. In practice, it usually requires only a few samples for KDE to generate the estimate which can approximate the underlying distribution very well.

3.2 Gossip-based Distributed KDE Algorithm

Consider a distributed network consisting of n distributed nodes. Initially, each node i has a local measurement $\mathbf{x}_i$, which can be regarded as i.i.d. samples of a random variable $\mathbf{X}$ with unknown distribution F. The problem of distributed density estimation is how to let each node get an estimate of the global distribution F, or alternatively the probability density function f, quickly and efficiently. In the following discussion, we assume that there is an underlying communication mechanism for any two nodes in the system to establish a communication channel (physical or virtual) and exchange messages.

The basic idea of our distributed kernel density estimation method, is to incrementally collect information and approximate the global KDE through a gossip mechanism. To achieve this goal, we maintain a local kernel set on each node i as follows,

$$S_i = \{\langle w_{i,l}, \mathbf{x}_{i,l}, \mathbf{H}_{i,l}\rangle, l = 1, \ldots, N_i\}, \tag{5}$$

where N_i is the current number of kernels on node i, and $\langle w_{i,l}, \mathbf{x}_{i,l}, \mathbf{H}_{i,l}\rangle$ is the l-th kernel. $\mathbf{x}_{i,l}$ is referred as the *location* of the kernel. $w_{i,l}$ and $\mathbf{H}_{i,l}$ are the weight and bandwidth matrix of the kernel. Initially, each node i only has one kernel $\langle w_{i,1}, \mathbf{x}_{i,1}, \mathbf{H}_{i,1}\rangle$, the weight $w_{i,1} = 1$, and $\mathbf{x}_{i,1} = \mathbf{x}_i$ is its local sample, and $\mathbf{H}_{i,1} = \mathbf{H}_i$ is the corresponding bandwidth matrix.

According to (1), the local estimation of the pdf on node i can be calculated as

$$\hat{f}_i(\mathbf{x}) = \sum_{l=1}^{N_i} w_{i,l} K_{\mathbf{H}_{i,l}}(\mathbf{x} - \mathbf{x}_{i,l}). \tag{6}$$

The gossip-based distributed estimation algorithm is illustrated in *Algorithm 1*. In the estimation process, every node periodically selects a random node from all other nodes on the network, and exchanges kernels with it. After exchanging, both the initiating node and target node merge their new received kernels into their local kernel sets and update their own local estimate.

During the above process, the size of local kernel set on a node grows quickly. Intuitively, each node will collect many kernels to estimate the global density distribution in a very short time, and the local estimate on the nodes will be closer and closer to the global KDE as the algorithm proceeds.

To analyze the consistency and convergence speed of our estimation algorithm, we define the *relative estimation error* on node i as $e_i = \frac{||\hat{f}_i - \hat{f}||}{||\hat{f}||}$, where $||\cdot||$ is the L^p-norm of the real function, and $\hat{f}_i(\mathbf{x})$ is the local density estimation based on the kernel sets S_i at the end of a gossip cycle, and

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K_{\mathbf{H}_i}(\mathbf{x} - \mathbf{x}_i)$$

is the global kernel density estimate.

Theorem 1. *For the algorithm 1, given any $\epsilon, \delta > 0$, after $\log(\frac{N^2}{\delta\epsilon^2})/\log(2\sqrt{e})$ gossip cycles, the relative estimation error on node i satisfies*

$$P(e_i < \epsilon) \geqslant 1 - \delta$$

Proof. Since all kernels are originated from the original data samples, we can re-write the local density estimation at node i

$$\hat{f}_i(\mathbf{x}) = \sum_{l=1}^{N_i} w_{i,l} K_{\mathbf{H}_{i,l}}(\mathbf{x} - \mathbf{x}_{i,l})$$

as

$$\hat{f}_i(\mathbf{x}) = \sum_{k=1}^{N} \tilde{w}_{i,k} K_{\mathbf{H}_k}(\mathbf{x} - \mathbf{x}_k),$$

Algorithm 1: Distributed KDE without Resource Constraints

1 The local kernel set S_i at node i is initialized as $S_i = \{\langle 1, \mathbf{x}_i, \mathbf{H}_i\rangle\}$

2 Gossip: On each node i, two following procedures are run in parallel:

1. **Active Procedure** : Node i periodically execute the following procedure for τ gossip cycles.

   (a) Randomly select a node j as its gossip target.

   (b) Send its local kernel set $S_i = \{\langle w_{i,l_1}, \mathbf{x}_{i,l_1}, \mathbf{H}_{i,l_1}\rangle, l_1 = 1, \ldots, N_i\}$ to node j.

   (c) After receiving the response from j, which consists of node j's local kernel set $S_j = \{\langle w_{j,l_2}, \mathbf{x}_{j,l_2}, \mathbf{H}_{j,l_2}\rangle, l_2 = 1, \ldots, N_j\}$, node i updates its own local kernel set as follows

$$S_i^{new} = \{\langle w'_{i,l_1}, \mathbf{x}_{i,l_1}, \mathbf{H}_{i,l_1}\rangle, l_1 = 1, \ldots, N_i\} \cup \{\langle w'_{j,l_2}, \mathbf{x}_{j,l_2}, \mathbf{H}_{j,l_2}\rangle,\ l_2 = 1, \ldots, N_j\},$$

   where

$$w'_{i,l_1} = \frac{1}{2} w_{i,l_1}, w'_{j,l_2} = \frac{1}{2} w_{j,l_2},$$

   During the union, the kernels with the same location and bandwidth matrix are merged, and the corresponding weights are summed together.

2. **Passive Procedure** : When the node i receives a gossip message from another node j, the following procedure is triggered.

   (a) Node i responds node j with its local kernel set.

   (b) Node i update its local kernels according to the received kernel set from j using the same method in the active procedure.

3 After the active procedure terminates, each node i uses its final kernel set to calculate $\hat{f}_i(\mathbf{x})$.

where $\tilde{w}_{i,k}$ is the weight of the kernel which corresponds to the original data sample $\mathbf{x}_k$ and $\tilde{w}_{i,k} = 0$ if the sample $\mathbf{x}_k$ is not presented in S_i.

Suppose node j is node i's gossiping target in cycle τ. Similarly, the local estimate at node j

$$\hat{f}_j(\mathbf{x}) = \sum_{l=1}^{N_j} w_{j,l} K_{\mathbf{H}_{j,l}}(\mathbf{x} - \mathbf{x}_{j,l})$$

can be written as

$$\hat{f}_j(\mathbf{x}) = \sum_{k=1}^{N} \tilde{w}_{j,k} K_{\mathbf{H}_k}(\mathbf{x} - \mathbf{x}_k),$$

where $\tilde{w}_{j,k}$ is the weight of the kernel corresponding to the original data sample $\mathbf{x}_k$ and let $\tilde{w}_{j,k} = 0$ if the sample $\mathbf{x}_k$ is not presented in S_j.

After gossip, node i and j have updated kernel sets $S'_i = S'_j$, and the new local estimates can also be represented by the weighted sum of original kernels. According to *Algorithm 1*, the new sample weights $\tilde{w}'_{i,k}$ and $\tilde{w}'_{j,k}$ are

$$\tilde{w}'_{i,k} = \tilde{w}'_{j,k} = \frac{\tilde{w}_{i,k} + \tilde{w}_{j,k}}{2} \tag{7}$$

Thus, the gossip process is actually the averaging process of the weights of the original data samples. For any data sample $\mathbf{x}_k$, denote $\mu_{\tau,k}, \sigma^2_{\tau,k}$ as the mean and variance of the weight $\tilde{w}_{i,k}$ in the τ-th gossip cycle. It is easy to see that, the mean value of all the weights is unchanged, i.e. $\mu_{\tau,k} = \mu = \frac{1}{N}$, while the variance of the kernel weights $\sigma^2_{\tau,k} = \sigma^2_\tau$ reduces quickly through the gossip process.

According to the property of the averaging process [11], when the selection of the gossip target is uniformly random, the reduction rate of the variance satisfies

$$\sigma^2_\tau = \frac{1}{2\sqrt{e}} \sigma^2_{\tau-1}. \tag{8}$$

Note that $\sigma^2_0 = \frac{1}{N}$, we have

$$\sigma^2_\tau = \frac{1}{N} (\frac{1}{2\sqrt{e}})^\tau. \tag{9}$$

In the τ-th gossip cycle, the relative estimation error on node i is bounded by

$$\begin{aligned} e_i^\tau &= \frac{\|\hat{f}_i^\tau - \hat{f}\|^2}{\|\hat{f}\|^2} \\ &= \frac{N \| \sum_{k=1}^N (\tilde{w}^\tau_{i,k} - \frac{1}{N}) K_{\mathbf{H}_k}(\mathbf{x} - \mathbf{x}_k) \|}{\| \sum_{k=1}^N K_{\mathbf{H}_k}(\mathbf{x} - \mathbf{x}_k) \|} \\ &\leqslant N \max_k |\tilde{w}^\tau_{i,k} - \frac{1}{N}| \\ &\leqslant N \left(\sum_{k=1}^N (\tilde{w}^\tau_{i,k} - \frac{1}{N})^2 \right)^{1/2} \end{aligned} \tag{10}$$

Since $E\left(\sum_{k=1}^N (\tilde{w}^\tau_{i,k} - \frac{1}{N})^2\right) = N\sigma^2_\tau$, using Markov inequality, we have

$$P(e_i^\tau < \epsilon) \geqslant 1 - \delta \tag{11}$$

when $\tau \geqslant \log(\frac{N^2}{\delta\epsilon^2}) / \log(2\sqrt{e})$.

□

The above theorem shows that the relative estimation error of the *algorithm 1* decreases exponentially and the local estimate at each node i converges to the global estimate in $O(\log N)$ steps. Since the number of kernels transmitted in one gossip cycle will not exceed the number of initial measurements N, there will be at most $O(N \log N)$ kernels to be transmitted in total.

3.3 Resource-Constrained Distributed KDE

The algorithm 1 presented above does not take account in the node's resource limitation in distributed networks. We assumed that each node has enough storage space to store at most N local kernels and large communication capability to transmit at most N kernels during a gossip cycle. However, this assumption is not always valid in some distributed systems, especially when the size of system N scales up. Here we extend the basic distributed estimation algorithm to a more flexible version to meet the resource constraint requirements.

Usually there are three constraints often considered in distributed systems:

1. Communication Constraints: The communication bandwidth between the nodes is often limited by the channel capacity. For our estimation algorithm, when the communication bandwidth is limited, it will become quite inefficient to send the full local kernel set from one node to another in a single gossip cycle.

2. Storage Constraints: The storage at a node is also limited. Sometimes it is impractical to store too large kernel set on the node. When the number of kernels that can be stored on the node is less than the total number of samples, it would be impossible for the distributed estimation to approach the global density function without any information loss. Instead, the goal will become to estimate the global KDE as accurately as possible with limited number of kernels.

3. Power Constraints: In some systems, e.g. the sensor network, the computation on a node is also limited to save the energy. Calculating KDE on a large sample set is computation intensive and consumes considerable power. Therefore, it would be inefficient to calculate KDE if the result kernel set is too large.

Taking the above constraints into consideration, we explicitly set some parameters of our gossip algorithm to limit the resource consumption of the estimation process. For simplicity, rather than specifying the accurate size of packet to be transmitted or stored, we set the maximal number of kernels sent from a node to another node in each gossip cycle to be L. We also set the maximal number of kernels can be stored at each node to be M, and the final estimate would be represented by at most M kernels.

To maintain the same estimation efficiency when the storage and communication are limited, the key problem would be how to represent the current density estimate with only a small set of representative samples, which is in fact a data reduction problem.

There are a number of data reduction methods proposed [6–8, 16]. However, most previous methods aim to extract some representative samples and are computational expensive, thus are not suitable for our distributed algorithm. In our case, we need to compress the dataset until its size is below some communication or storage threshold. And we want to preserve the density estimation as much as possible. Furthermore, the data reduction method should be simple and fast.

Here we propose a simple while efficient data reduction method, which can be regarded as a bottom-up construction of a KD-tree structure [1]. Suppose there is a density estimate represented by K_1 Gaussian kernels $\{\langle w_l, \mathbf{x}_l, \mathbf{H}_l\rangle, l = 1, \ldots, K_1\}$, our goal is to output K_2 kernels $\{\langle w'_l, \mathbf{x}'_l, \mathbf{H}'_l\rangle, l = 1, \ldots, K_2\}$ to approximate the original density estimate $\hat{f}(\mathbf{x})$, where $K_2 < K_1$. To this goal, we iteratively choose two kernels with minimum distance and merge them into a new kernel. After $K_1 - K_2$ iterations, we get the reduced kernel set whose size is exactly K_2. The detailed data reduction method is described in Algorithm 2.

Based on the above data reduction algorithm, we can modify the distributed KDE algorithm to meet the resource-constraints. During the estimation process, we employ the data reduction method whenever the size of kernel set exceeds the communication or storage constraints.

There is some information loss during the data reduction procedure, which will influence the accuracy of the whole estimation process. Obviously, the information loss of data reduction is determined by the parameters L and M. In practice, we may need to adjust these parameters to achieve the desirable trade-off between the overhead (bandwidth and storage) and the performance (accuracy and speed) of the algorithm. Because KDE is highly robust and only requires a few data samples to generate an acceptable estimate, our data reduction method can efficiently represent the local kernel set with a relative small constraint parameter. Experiments show that, with a small communication and storage overhead, our algorithm can result in estimate whose accuracy is comparable with the global KDE and the estimation results converge almost as quickly as *Algorithm 1* does. We will further analysis the performance of our algorithm in the experimental part.

Algorithm 2: Data Reduction Algorithm for Distributed KDE

Input: The kernel set
$S = \{\langle w_i, \mathbf{x}_i, \mathbf{H}_i\rangle, i = 1, \ldots, K_1\}$,
$K_2 (K_2 < K_1)$.

Output: The compressed kernel set
$\{\langle w'_j, \mathbf{x}'_j, \mathbf{H}'_j\rangle, j = 1, \ldots, K_2\}$
approximating the original KDE.

1 $l \leftarrow K_1$;
2 **while** $l > K_2$ **do**
3 Select two kernels j_1 and j_2 which have minimal distance between their centers $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$;
4 Merge the two kernels into a parent kernel j according to the following rule:

$$
\begin{aligned}
w'_j &= w_{j_1} + w_{j_2} \\
\mathbf{x}'_j &= \frac{w_{j_1}\mathbf{x}_{j_1} + w_{j_2}\mathbf{x}_{j_2}}{w_{j_1} + w_{j_2}} \\
\mathbf{H}'_j &= \frac{w_{j_1}(\mathbf{H}_{j_1} + \mathbf{x}_{j_1}\mathbf{x}_{j_1}^{\mathrm{T}}) + w_{j_2}(\mathbf{H}_{j_2} + \mathbf{x}_{j_2}\mathbf{x}_{j_2}^{\mathrm{T}})}{w_{j_1} + w_{j_2}} - \mathbf{x}'_j\mathbf{x}'^{\mathrm{T}}_j
\end{aligned}
\tag{12}
$$

delete kernels j_1 and j_2 from S, and add kernel j into S;
5 $l \leftarrow l - 1$.
6 **end**

3.4 Practical Considerations

In the above discussion, we have assumed that each node holds only one measurement. In some practical networks, it is also possible that each node holds a number of measurements. In this case, the initial local set on each node will contain more than one kernel. And our algorithm can be extended to this case without any special difficulty.

There are some other implementation considerations when applying the above estimation algorithm in practical systems. First, in most large-scale distributed systems, there is no global clock available. It is difficult for all the nodes to run the estimation algorithm synchronously. Furthermore, in some dynamic systems, since the nodes are dynamically joining and leaving the network, we need to periodically restart the estimation process to provide the up-to-date distribution estimates. To implement restarting and asynchronous estimation, we run the algorithm in consecutive rounds of length $\Delta = N_c\delta$ (where δ is the length of a gossip cycle, and N_c is the number of cycles in a round), and start a new instance of the distributed KDE in each round. In the implementation, we attach a unique and increasing round identifer to each gossip message so that the messages of different rounds can be distinguished from each other. Once the node receives a round identifier which is larger than its current one, it will immediately join the new estimation process. At any time, the result of the last round is taken as the current estimate of the density distribution. New joining nodes can get the current estimate immediately from other nodes and will join the next round of estimation. Similar restarting mechanism have been adopted in some aggregation algorithms [11].

Another factor that may affect the estimation result is the choice of bandwidth matrix. In our experiments, we set the global bandwidth matrix $\mathbf{H}_i = h^2\mathbf{I}$ in advance for the sake of simplicity and clarity. We observed that, for most cases, simply setting h to a small value can result in good estimation result. However, it is also possible to to estimate the bandwidth matrix from the local kernel set in the gossip process. And there is already a number of data-driven methods for estimating global or local bandwidth matrix from sample sets [3, 5, 19]. In practice, we can use our estimation algorithm with any bandwidth selection method, if necessary.

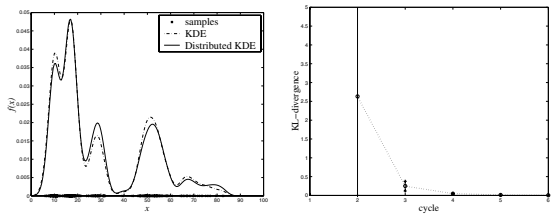
For some distributed systems, such as in sensor networks or P2P networks, nodes can only contact some connected neighbors. It is difficult or impossible to randomly select a node from all other nodes to contact. However, some previous work on gossip protocol shows that the topology of the overlay network only affects the convergence speed of the averaging process [2, 11]. Therefore, the nodes can still get the same estimation result by periodically exchanging their local estimate only with its neighbors. For P2P network, another possible solution to this problem would be deploying a *peer sampling service* [9] over the network. The peer sampling service provides each node an updating node cache with good randomness through periodically exchanging and updating the nodes in the cache. A detailed evaluation and discussion on the implementation of the peer sampling service can be found in [9].

4 Simulation and Discussion

To evaluate the performance of our distributed KDE algorithm, we test our algorithm on a number of datasets with different size, dimension and distribution. We explore how the constraint parameters affect the convergence and accuracy of the distributed algorithm. We also compare the performance of our algorithm with the Newcast-EM algorithm [10] on some non-Gaussian data distributions. In addition, we examine the robustness of our algorithm when there are frequent node failures during the estimation process.

4.1 Experimental Methodology

We simulate a distributed network which consists of a number of connected nodes. Initially, each node holds only one data sample which is drawn from an underlying distribution. We implement our distributed KDE algorithms on



(a) The 1D dataset and estimation results. (b) Convergence

Figure 1. Distributed KDE on 1D dataset

these nodes. In each experiment, we run the algorithm for a number of gossip cycles and investigate the estimation result.

To quantitatively analysis the estimation accuracy and compare our algorithm with others, we use the Kullback-Leibler divergence (KL-divergence) [15] as the performance metric of the estimation algorithm. In our experiments, we monitor the change of local density estimate on each node and calculate the K-L divergence from the baseline distribution, which may be the global KDE or the real underlying distribution, to the local estimate.

4.2 Simulation Results

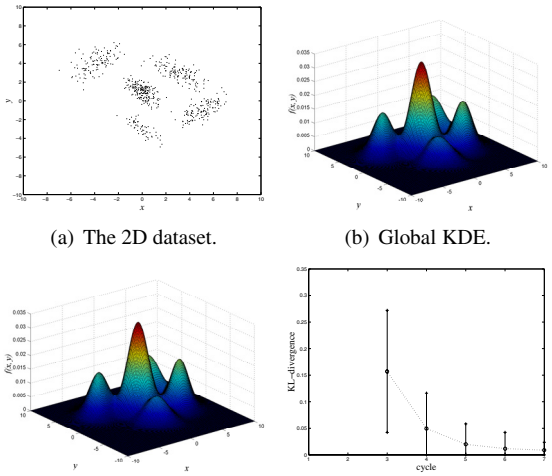
4.2.1 Estimation Accuracy and Convergence

Since the convergence of the distributed KDE without constraints is guaranteed by *Theorem 1*, we mainly focus on the performance of the resource-constrained distributed estimation algorithm. Fig. 1 shows the estimation result on a 1D dataset, which consists of 200 samples drawn from 5 mixed Gaussian distribution. Such a heavily overlapped distribution (five weighted Gaussians with different variances and locate closely) is difficult to estimate even in global settings.

Fig. 1(a) shows the local estimation result on a node after 6 gossip cycles. The global KDE result and the original dataset are also plotted in the figure. We can see that the estimated curve of our proposed algorithm is close to the global result.

To further verify the overall performance of our algorithm, we calculate the K-L divergence from the global KDE result to the distributedly estimated pdf on every node in each gossip cycle. Then, we compute the mean and variance of the K-L divergence values from all nodes within each gossip cycle. The change of K-L divergence is shown in Fig. 1(b). We can see that both the mean and the variance decreases exponentially, and after four cycles, the estimation errors on most nodes drop to a small value.

We also tested our Distributed KDE algorithm on some high-dimensional datasets. Fig. 2 shows the estimation result on a 2D dataset which consists of 600 samples drawn from a mixed distribution of 5 Gaussian components.



(a) The 2D dataset. (b) Global KDE.

(c) Distributed KDE. (d) Convergence

Figure 2. Distributed KDE on 2D dataset

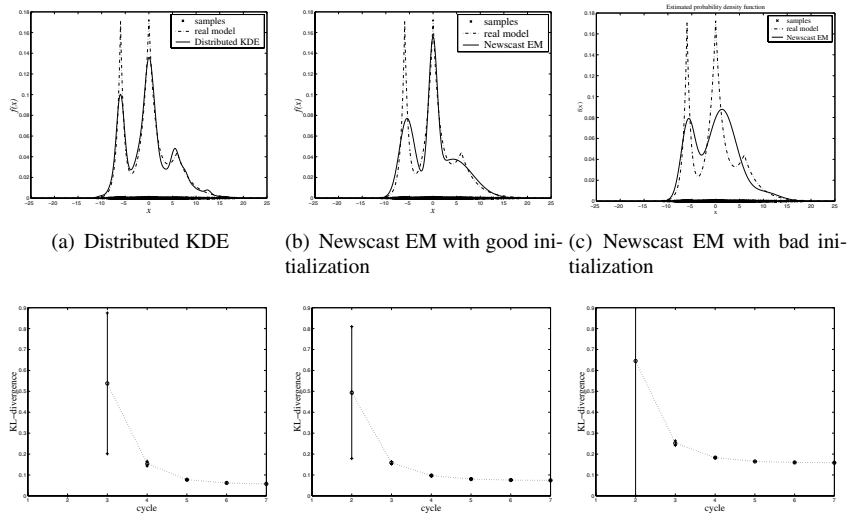
Fig. 2(a) shows the location of the original data samples. The result of global KDE is shown in Fig. 2(b), and the distributed estimation result on a random chosen node is shown in Fig. 2(c). We can see that the distributed kernel density estimation is close to the global KDE. Fig. 2(d) shows the change of the statistics of the KL-divergence with gossip cycles. We observe that the convergence trend is similar with 1D case.

4.2.2 Non-Gaussian Distribution

We compare our proposed distributed KDE algorithm with Newscast EM algorithm [14]. Fig. 3 shows the estimation results of our distributed KDE algorithm and Newcast EM on the same 1D non-Gaussian dataset, which consists of 200 samples drawn from three bi-exponential distributions.

We run the Newscast EM algorithm with different initial values. In the first experiment, in order to obtain the best estimation result, we set the initial location of the Gaussian to the center of the three distributions. The result is shown in Fig. 3(b) and Fig. 3(e). In the second experiment, we slightly change the initial location of the Gaussian, and the corresponding result is shown in Fig. 3(c) and Fig. 3(f).

From Fig. 3(a) and Fig. 3(d), we can see that, though without any prior information of the distribution, our distributed DKDE algorithm still works well and results in highest estimation accuracy. From Fig. 3(b) and Fig. 3(e), we can see that, even tuned with the best tuned parameters, the EM estimation result still deviates from the underlying distribution. The estimation error of the distributed EM algorithm is due to its wrong model assumption. From Fig. 3(c) and Fig. 3(f), we observe that the sub-optimal initialization of the EM algorithm leads to a much worse estimation of the probability density function, which shows that



(a) Distributed KDE (b) Newscast EM with good initialization (c) Newscast EM with bad initialization

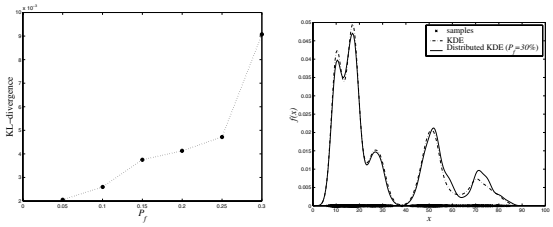
(d) Convergence of Distributed KDE (e) Convergence of Newscast EM with good initialization (f) Convergence of Newscast EM with bad initialization

Figure 3. Comparison with Newscast EM algorithm

the distributed EM algorithm is highly sensitive to initial parameters. On the other hand, we also find that the convergence speed of Newscast EM with good initialization is slightly faster than our proposed algorithm. This is because the EM algorithm has utilized extra prior information (e.g. the number of Gaussian models) and thus can quickly reduce the estimation error at the beginning.

4.2.3 Resilience to Node Failure

To test the robustness of our distributed algorithm, we simulate the situation where nodes fail frequently in the estimation process. At the beginning of each gossip cycle, we select $N \times P_f$ nodes randomly selected and discard them from the network. We conduct a series of experiments where the failure rate P_f ranges from 5% to 30%, and compare the estimation results with the global KDE result which is calculated from the original complete dataset. Fig. 4 shows the estimation result under node failure. We can see that the estimation result is still acceptable even when there are a high percentage nodes failing in each cycle, which shows the distributed KDE is robust against node failure.



(a) KL-divergences (b) Estimation result ($P_f = 30\%$).

Figure 4. Impact of node failure

5 Conclusion

In this paper, we proposed a distributed kernel density estimation algorithm which can learn the non-parametric model from distributed measurements with arbitrary distribution efficiently. It requires little prior information about the underlying distribution and converges to the global KDE with any given precision in $O(\log N)$ steps. To extend the the algorithm to deal with the situations of limited resources, we propose an efficient data reduction method for exchanging local density estimate efficiently and also designed a resource-constrained distributed kernel density estimation algorithm. Through extensive simulation experiments, we found that our resource-constrained distributed estimation algorithm can achieve high estimation accuracy with only small communication and storage overhead. Compared with the distributed EM algorithm, our estimation method is more robust and accurate.

6 Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable suggestions and comments.

References

[1] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, Sep. 1975.

[2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Inform. Theory*, 52(6):2508–2530, Jun. 2006.

[3] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):281–288, 2003.

[4] V. Delouille, R. Neelamani, and R. Baraniuk. Robust distributed estimation in sensor networks using the embedded polygons algorithm. In *ACM IPSN '04*, pages 405–413, Berkeley, California, USA, Apr. 2004.

[5] T. Duong and M. L. Hazelton. Cross-validation bandwidth matrices for multivariate kernel density estimation. *Scandinavian Journal of Statistics*, 32(3):485–506(22), 2005.

[6] M. Girolami and C. He. Probability density estimation from optimally condensed data samples. *IEEE Trans. on PAMI*, 25(10):1253–1264, Oct. 2003.

[7] L. Holmstrom and A. Hamalainen. The self-organizing reduced kernel density estimator. In *Proc. IEEE International Conference on Neural Networks*, pages 417–421, 1993.

[8] A. T. Ihler, J. W. Fisher III, and A. S. Willsky. Using sample-based representations under communications constraints. *Massachusetts Institute of Technology, LIDS Technical Report No. 2601*, Dec. 2004.

[9] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. Steen. The peer sampling service: Experimental evaluation of unstructured gossip-based implementations. *Lecture Notes in Computer Science*, 3231(3):79–98, 2004.

[10] M. Jelasity, W. Kowalczyk, and M. Steen. Newscast computing. Technical Report IR-CS-006, Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands, Nov. 2003.

[11] M. Jelasity, A. Montresor, and O. Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Trans. on Computer Systems*, 23(3):219–252, Aug. 2005.

[12] H. Jiang and S. Jin. Scalable and robust aggregation techniques for extracting statistical information in sensor networks. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS06)*, 2006.

[13] M. Klusch, S. Lodi, and G. Moro. Distributed clustering based on sampling local density estimates. In *Proc. IJCAI-03*. AAAI Press., 2003.

[14] W. Kowalczyk and N. A. Vlassis. Newscast EM. In *Advances in Neural Information Processing Systems 17*, pages 713–720, Cambridge, MA, 2005. MIT Press.

[15] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, (22):79–86, 1951.

[16] P. Mitra, C. A. Murthy, and S. K. Pal. Density-based multiscale data condensation. *IEEE Trans. on PAMI*, 24(6):734–747, 2002.

[17] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Decentralized detection and classification using kernel methods. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 80. ACM Press, 2004.

[18] R. Nowak. Distributed EM algorithms for density estimation and clustering in sensor networks. *IEEE Trans. on Signal Processing*, 51(8):2245–2253, Aug. 2003.

[19] V. C. Raykar and R. Duraiswami. Very fast optimal bandwidth selection for univariate kernel density estimation. Technical Report CS-TR-4774, Department of computer science, University of Maryland, Collegepark, 2005.

[20] A. Ribeiro and G. B. Giannakis. Bandwidth-constrained distributed estimation for wireless sensor networks-part i: Gaussian case. *IEEE Transactions on Signal Processing*, 54(3):1131–1143, 2006.

[21] A. Ribeiro and G. B. Giannakis. Bandwidth-constrained distributed estimation for wireless sensor networks-part ii: unknown probability density function. *IEEE Transactions on Signal Processing*, 54(7):2784–2796, 2006.

[22] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. New York: Wiley-Interscience, 1987.

[23] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley-Interscience, 1992.

[24] N. Vlassis, Y. Sfakianakis, and W. Kowalczyk. Gossip-based greedy gaussian mixture learning. In *Proc. 10th Panhellenic Conf. on Informatics.*, Volos, Greece, Nov. 2005.