Efficient Kernel Density Estimation Over Distributed Data*

Chris Giannella Haimonti Dutta Sourav Mukherjee Hillol Kargupta†

Abstract

We consider the problem of Kernel Density Estimation (KDE) over distributed data. Our work is motivated by the fact that, in some cases, large data repositories are, in their native form, distributed over many sites. In environments where communication is limited, centralization of the data can be problematic. We examine a framework where the sites build local models which are combined at a central site to produce an approximation to the KDE over the entire dataset. We carry out experiments comparing two realizations of this framework (uniform sampling and linear binning). We compare these two techniques in terms of their accuracy at: directly approximating the centralized KDE (measured by 2-norm function distance), density-based classification, and density-based clustering.

1 Introduction

Traditional data mining and analysis techniques assume that the data set of interest in located entirely on a single server. However, this assumption is not always justified. In some cases, large data repositories are, in their native form, distributed over many servers, perhaps over a large geographic distance. To utilize traditional data analysis technology, the data must be first centralized. But, this introduces large communication costs. In environments where this commodity is limited, centralization can be problematic. Moreover, in the case where local data is sensitive – not to be divulged in its raw form – centralization introduces unacceptable security issues. The fields of distributed [3] and privacy-preserving [13] data mining have arisen to address these needs.

In this paper, we address the problem of non-parametric density estimation over distributed data. Our work is motivated by a recent paper by Merugu and Ghosh [7] that addresses the problem of parametric density estimation over distributed data. The goal of their approach was to reduce communication complexity, reduce synchronization requirements, and provide privacy. In particular, each site builds a local model describing its data, then sends it to a central coordinator. The coordinator combines the local models to produce a model by which the global KDE can be estimated. Our approach differs from theirs in that we consider *non*-parametric density estimation, specifically, Kernel Density Estimation (KDE). The advantage of doing so is the added flexibility of not assuming an underlying distribution family. Similar to Merugu and Ghosh, we develop a "local model, global combination" approach to address the same goals: reduce communication, synchronization, and provide privacy. Due to space limitations, we only discuss the reduction of communication and synchronization leaving privacy for a forthcoming extended paper.

Paper organization: Section 2 discusses related work from the Distributed Data Mining (DDM) literature, in particular, other papers adopting the local model, global combination approach. Section 3 discusses a standard KDE formulation from Statistics and two local model, global combination schemes for distributed computation: uniform sampling and linear binning. Section 4 describes an evaluation of the accuracy of these schemes at direct density function accuracy (2-norm), density-based classification accuracy, and density based clustering accuracy (accuracies are measured with respect to centralized KDE computation). Finally, Section 5 concludes the paper.

2 Related Work

KDE has been extensively studied in the Statistics literature [10, 11, 14] and applied in data mining [2] (on centralized data).

2.1 Local Model, Global Combination Approaches As discussed earlier, Merugu and Ghosh [7] address the problem of distributed parametric density estimation. Later, the same authors address density estimation on *vertically* distributed data [8] – each site has some of the attributes of the same set of tuples. This differs from our work which assumes the data is *horizontally* distributed – each site has a different set of tuples over the same set of attributes.

Klusch *et al.* [6] have written perhaps the most

*Primary affiliation of all authors: Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County USA, {cgiannel,hdutta1,sourav1,hillol}@cs.umbc.edu

†Also affiliated with Agnik, LLC., Columbia, Maryland

closely related work. They consider the problem of density based clustering on distributed data. Sites build local density grids which are centralized and combined. Then a standard density based clustering algorithm is applied using density estimates based on Shannon sampling from signal processing. Our work is quite similar to Klusch's (we use a grid based approach), however, he provided no experiments to assess the effectiveness of the approach. Our work addresses this shortcoming through an experimental comparison of the accuracy of a grid based approach with a uniform sampling approach.

Zhang and Cheung [16] address the problem of distributed estimation of a manifold fitting the data. Silvestri and Orlando [12] address the problem of estimating the frequent itemsets over a distributed transaction database. Both of these address a different kind of analysis (manifold fitting, frequent itemset discovery) than we do (KDE).

2.2 Other DDM Related Work Recently work has been carried out on data analysis and mining in a large scale distributed network (peer-to-peer network), a sampling follows. Kempe *et al.* [4] discuss the use of gossip to address the problem of simple aggregate computation (*e.g.* Sum). Wolff and Schuster [15] address the problem of association rule mining based on local majority voting. Datta *et al.* [1] address the problem of K-means clustering. None of these address non-parametric density estimation.

3 Kernel Density Estimation

3.1 Centralized KDE The formulation of KDE presented here is based on standard techniques described in the literature ([10],[11],[14]). Let $X \subseteq \mathbb{R}^n$ be a domain of interest and $f\colon X \to \mathbb{R}_{\geq 0}$ be an unknown probability density function. Let $D \subseteq X$, be a set of independent samples drawn according to f. For any $\mathbf{x} \in X$, we estimate $f(\mathbf{x})$ by:

$$\hat{f}_D(\mathbf{x}) = \frac{1}{|D|} \sum_{\mathbf{s} \in D} \prod_{i=1}^{n} \frac{1}{h_i} K(\frac{x_i - s_i}{h_i})$$

where $|D|$ is the number of tuples in D, K is the *kernel function*,[1] and x_i, s_i, h_i are the i^{th} components of $\mathbf{x}$, $\mathbf{s}$, $\mathbf{h}$. The choice of the bandwidth vector $\mathbf{h}$ is important and can be made in several standard ways. We use the "normal plug-in" method:

$$h_i = (\frac{4}{|D|(n+2)})^{\frac{1}{n+4}} \hat{\sigma}_i$$

[1] We use the standard Gaussian kernel.

were $\hat{\sigma}_i$ is the sample standard deviation computed from the i^{th} column of D.

3.2 KDE on Distributed Data We assume a distributed environment comprised of p sites, each containing a subset D_l of D. We define the *local KDE function* for site l as:

$$\hat{f}_{D_l}(\mathbf{x}) = \sum_{\mathbf{s} \in D_l} \prod_{i=1}^{n} \frac{1}{h_i} K(\frac{x_i - s_i}{h_i})$$

and write the global KDE as a sum of local KDEs:

$$\hat{f}_D(\mathbf{x}) = \frac{1}{|D|} \sum_{l=1}^{p} \hat{f}_{D_l}(\mathbf{x})$$

KDE can be performed on distributed data as:

1. The parameters $|D|$ and $\hat{\sigma}_i$ are computed using converge-cast [9] resulting in the central site getting $\mathbf{h}$ and $|D|$. These are broadcast to all the sites. This initialization step requires two rounds of complete synchronization.

2. When the central site is presented with a set of new input vectors $\{\mathbf{x}\} \subseteq X$, another round of complete synchronization is required to compute $\{\hat{f}_D(\mathbf{x})\}$. The central site broadcasts $\{\mathbf{x}\}$ to all sites, which, in turn, send their local KDEs $\{\hat{f}_{D_l}(\mathbf{x})\}$ back. The central site sums these and divides by $|D|$ to get the global KDEs, $\{\hat{f}_D(\mathbf{x})\}$.

3.3 Improvements The scheme presented above has a shortcoming: complete synchronization is needed for every set of vectors $\{\mathbf{x}\}$ on which the KDE is simultaneously calculated. This can be problematic if the overarching data analysis task requires KDE to be computed on a set of points $\mathbf{x}^1, \ldots, \mathbf{x}^q$ in a sequential manner. This is necessary if $\mathbf{x}^i$ is determined based on the KDE of the previous points, for example, in density-based clustering. As a result, $q + 2$ rounds of complete synchronization are required for the data analysis task.

One solution is for each site l to construct a model, M_l, approximating its local KDE, $\hat{f}_{D_l}$, then send M_l to the central site. It combines the local models to produce a global model, M, which is used to approximate the global KDE, $\hat{f}_D$. Since the central site only uses M, then no contact with the other sites is required to estimate the KDE of $\mathbf{x}^1, \ldots, \mathbf{x}^q$. Hence, only two complete rounds of synchronization are required. However, the density estimation produced is an approximation of the centralized KDE, $\hat{f}_D$. We examine this issue empirically through experiments involving two realizations of the framework.

Uniform Sampling: Site l takes a uniform sample (without replacement), M_l, from D_l. This is the local model of site l. The central site combines these local models into a global model in the obvious way, $M = \cup_{l=1}^{p} M_l$. Finally $\hat{f}_D(\mathbf{x})$ is approximated as $\hat{f}_M(\mathbf{x})$ (using the bandwidth vector $\mathbf{h}$ found from the whole dataset D as described earlier).

Linear Binning: Linear Binning is based on the concept of a hyper-rectangular grid enclosing the domain, X. The grid is defined by the lower-most and upper-most corner points of the hyper-rectangle, $\vec{L}$, $\vec{U}$ and ρ the number equal partitions of each side. To agree upon $\vec{L}$ and $\vec{U}$, each site can choose a bounding hyper-rectangle for its data, then send the corner-points to a central site. The central site chooses a hyper-rectangle enclosing all corner-points and broadcasts its corner-points to all sites. All of these messages can be included in step 1. in Section 3.2. The choice of ρ depends upon the amount of space allocated the grids and is described later (the central site chooses ρ and broadcasts).

Each grid point $\mathbf{g}$ has an associated weight $w_g = \sum_{l=1}^{p} w_{l,g}$. The local contributions to this weight, $w_{l,g}$ are given by $w_{l,g} = \sum_{\mathbf{s} \in D_l} s(w_{l,g})$, where $s(w_{l,g})$ is the contribution of the local data point $\mathbf{s}$ to $w_{l,g}$. Given a data point $\mathbf{s}$ in D, and a grid point $\mathbf{g}$, we compute $s(w_{l,g})$ as follows. First, the grid cell containing $\mathbf{s}$ is found. If $\mathbf{g}$ is not a vertex of this cell, then $s(w_{l,g}) = 0$. Otherwise, let $\mathbf{g}^*$ be the vertex of this cell diagonally opposite to $\mathbf{g}$. Define:

$$s(w_{l,g}) = \frac{Volume(\mathbf{s}, \mathbf{g}^*)}{Volume(\mathbf{g}, \mathbf{g}^*)}$$

where: $Volume(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{n} |x_i - y_i|$.

The local model M_l is the collection of all local weights $w_{l,g}$. The global model, M, is constructed by summing the local weights: $w_g = \sum_{l=1}^{p} w_{l,g}$. Given a new vector $\mathbf{x} \in X$, M is used to approximate $\hat{f}_D(\mathbf{x})$ as:

$$\frac{1}{|D|} \sum_{\mathbf{g}} w_g \prod_{i=1}^{n} \frac{1}{h_i} K(\frac{x_i - g_i}{h_i})$$

A third realization of this framework was proposed by Klusch *et al.* [6] based on local density grids and Shannon sampling. However, Klusch did not conduct any experiments to assess accuracy. We leave consideration of his realization in our experiments to future work.

4 Evaluation

The primary goal of this paper is to empirically compare the accuracies of the two realizations of the local model, global combination framework for approximating a KDE. To do so, we carry out a series of experiments in a simulated environment on a single machine. Since our experiments are only intended to measure accuracy with respect to communication cost, a genuine distributed environment is unnecessary.

We synthetically generate dataset D. This data is then partitioned into p parts, one for each site.[2] Next, a fixed communication allowance $c \in [0, 1]$ is specified. This determines the total communication allowed the distributed algorithm *i.e.* the amount of space each local model can occupy. The total communication required to centralize D is $4mn$ bytes[3] where $|D| = m$, hence, the total communication allowed is $4cmn$ bytes. Assuming this budget is allocated equally, each site can use at most $4mnc/p$ bytes for its local model. For uniform sampling, each site will sample mc/p tuples for its local model. For linear binning, the size of the local model is determined by the number of grid points. Assuming each grid dimension partitioned into ρ equal parts, we have $\rho = \lceil [\frac{cmn}{p}]^{1/n} \rceil - 1$.

Let M^u and M^b denote the approximation to the global KDE produced from the uniform sampling and linear binning global models, respectively. The experiments vary c and measure the accuracy of M^u and M^b with respect to the following measures. The process of generating M^u and measuring accuracy is repeated for some number of trials and average results are reported with error bars indicating the 95% confidence interval.

4.1 L2-Norm Given function $h : X \to \mathbb{R}$, the two norm of h, denoted $||h||_2$ is $\int_{\mathbf{x} \in X} h(\mathbf{x})^2 d\mathbf{x}$. This can be computed as follows. Let Γ be a uniform grid of X (all cells have the same volume V) – we are assuming X is a hyper-rectangle. Note, Γ is not the same as the grid discussed earlier with linear binning. In fact, Γ ought to be much finer. We approximate $||h||_2$ by $V \sum_{g \in \Gamma} h(g)^2$.

We measure the L2-norm error, $||\hat{f}_D - M^u||$ and $||\hat{f}_D - M^b||$, as communication increases. We also plot $||\hat{f}_D||$ as a baseline. Three experiments are performed, with each experiment using a data set having higher dimensionality and larger number of tuples. However, the ratio of the number of tuples to the number of dimensions is kept constant. In each experiment, a synthetic data set, consisting of a mixture of 5 Gaussian distributions, is used. The number of sites and uniform sampling trials is 10 and 50.

Our objective, in these experiments, is two-fold. (1) To examine the behavior uniform sampling and linear binning with respect to increasing communication. (2) To determine which of the two is more communication-efficient. More specifically, to compare

[2] In our tests, we used five or ten sites as indicated.

[3] assuming each tuple component requires 4 bytes

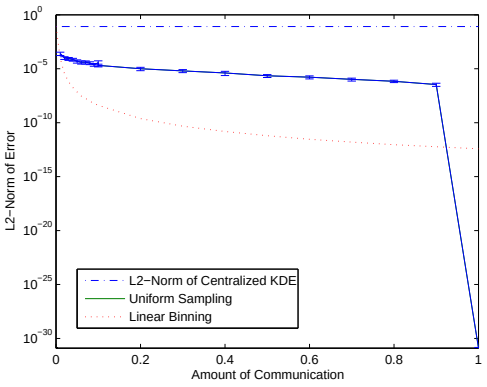

Figure 1: L2-norm comparison on a 5000 tuple, 1D dataset.

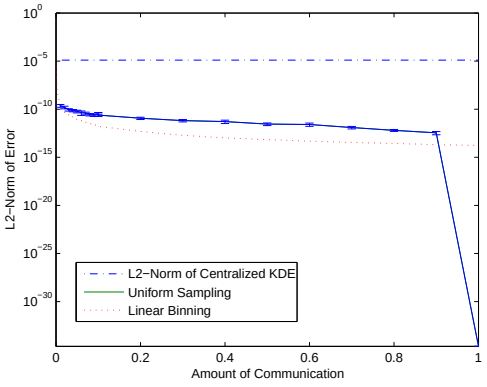

Figure 2: L2-norm comparison on a 10000 tuple, 2D dataset.

the performance of these techniques as the data dimensionality increases. We expect uniform sampling to perform better than linear binning for higher dimensional data. This is because, for a fixed communication allowance, the number of partitions on each dimension of the grid, ρ, grows exponentially with dimensionality, n. Hence, the linear binning grid grows exponentially more sparse with dimensionality.

We test the above conjecture in three experiments whose results are shown in Figures 1, 2, and 3. The x-axes of these figures, "Amount of Communication", refer to c, the communication allowance. The figures contain a horizontal line showing the L2-norm of $\hat{f}_D$ to be used as a baseline against which the L2-norm errors are compared. The degree to which the errors are smaller than the horizontal gives an indication as to their overall quality.

The plots show that the error of linear binning decreases smoothly but rather quickly at first, then slows.

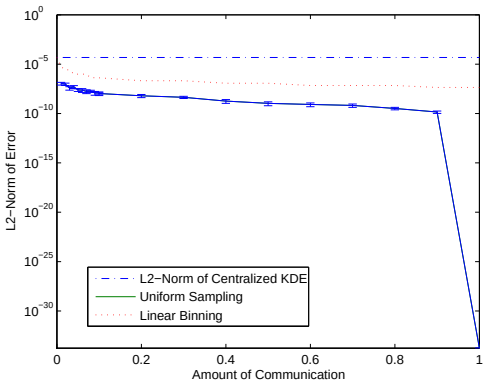

Figure 3: L2-norm comparison on a 20000 tuple, 4D dataset.

Uniform sampling, on the other hand, demonstrates an initial slow decrease followed by a sharp drop to zero at high communication.[4] We also see that linear binning exhibits lower error for one dimensional data, but higher for four dimensional – confirming our conjecture. Finally, we see that the overall error obtained by both methods is low for small communication. For example, at 20% communication, both methods have L2-norm error at least 5 orders of magnitude less than $||\hat{f}_D||$ for 1 and 2 dimensional data, at least 3 orders for 4 dimensional data.

4.2 Supervised Learning We compare the accuracy of M^u and M^b in terms of their classification accuracy when used for *non-parametric discrimination*. Assume the tuples of D are labeled in one of two ways. Let $D(0)$ denote the collection of tuples labeled one way and $D(1)$ denote the other (according to a labeling function $Lab : X \to \{0, 1\}$). Given a new tuple $\mathbf{x} \in X$, the goal is to classify $\mathbf{x}$. To do this, we estimate the *posterior odds* as follows

$$\frac{Pr(Lab(\mathbf{x}) = 0|D)}{Pr(Lab(\mathbf{x}) = 1|D)} \approx \frac{\hat{f}_{D(0)}(\mathbf{x})}{\hat{f}_{D(1)}(\mathbf{x})} \frac{|D(0)|}{|D(1)|}.$$

Thus, $\mathbf{x}$ is classified as 0 if the above expression is greater than one, otherwise, it is classified as 1.

We use a 2-dimensional synthetic data set with 20000 tuples. The tuples are drawn from a mixture of 4 Gaussian distributions. The means of the Gaussians are located at the vertices of a square in 2-dimensional space, say $ABCD$. Distributions having means at A and C belong to one class, while those having means

[4] As expected, for $c = 1$, uniform sampling achieves zero error. In this case, the entire dataset is centralized.

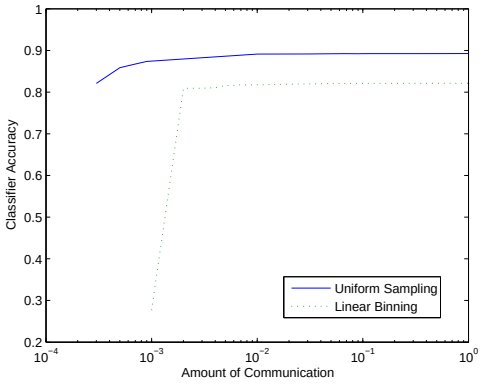

Figure 4: Classification accuracy comparison on a 20000 tuple, 2D dataset.

at B and D belong to the other class. Classification accuracy of M^u and M^b are each measured using 10-fold, cross validation. The number of sites and uniform sampling trials is 10 and 20. The results are shown in Figure 4.[5]

We find that as the amount of communication increases, the accuracy of the classifiers increase very rapidly. However, the maximum accuracy of the linear binning classifier falls below the maximum accuracy of the uniform sampling classifier (the accuracy of uniform sampling at $c = 1$ is exactly the same as that of the centralized approach). Most of the points in the dataset are close to the means of the Gaussians that generated them. Hence, these points can be classified easily, even with very little communication. This explains why accuracy is high even at small amounts of communication. Greater amounts of communication are required to properly classify points that lie near the "boundaries" of two (or more) Gaussians. The plot of the L2-norm of these classifiers in the 2-dimensional case, as shown in Figure 2 might provide insights into what happens as communication is increased. We find that as communication increases, the accuracy of uniform sampling ultimately becomes much greater than the accuracy of linear binning. When the amount of communication equals that required for full centralization, the L2-norm for uniform sampling becomes almost zero, while that for linear binning does not. This might be correlated to the difference in maximum classifier accuracies; we would like to investigate this in more detail in future work.

[5]Error bars are not included because they are very small – all have length less than 0.64% of their associated point.

4.3 Clustering We measure the accuracy of M^u and M^b by computing the clusterings they produce when used in a standard density-based clustering algorithm and the clustering produced when the algorithm is applied to a centralized dataset. Accuracy is measured in terms of the difference between distributed and centralized clusterings. The clustering algorithm we use is due to Kittler [5]. Assuming a function $\hat{f}$ is used to estimate the pdf f, the algorithm consists of the following steps. (1) Map the data set, D onto a sequence and associate each point with its density estimation from $\hat{f}$. The mapping is designed with the goal that points associated with the same mode of the mapping will also be associated with the same mode of f. The interested reader is referred to [5] for further details of this procedure. (2) Separating points between modes of the mapping are determined (a one dimensional problem). This process, breaks up the dataset into clusters.

Using the KDE over dataset D once centralized, let $C_1, \ldots, C_t$ denote the clustering obtained by applying Kittler's algorithm. For site l, let $C_1^l, \ldots, C_{t_l}^l$ denote the clustering obtained when Kittler's algorithm is applied to D_l using M^u or M^b. The accuracy of this clustering is quantified in terms of the difference between $C_1^l, \ldots, C_{t_l}^l$ and the intersection of the centralized clustering with D_l: $\hat{C}_1 = D_l \cap C_1, \ldots \hat{C}_t = D_l \cap C_t$. The accuracy of the clustering at site l is

$$Acc(l) = \frac{\sum_{r \neq s \in D_l} err_l(r,s)}{|D_l|(|D_l|-1)}.$$

where $err_l(r,s)$ is zero if $r, s \in C_\alpha^l$ and $r, s \in \hat{C}_\beta$ for some $1 \leq \alpha \leq t_l$ and $1 \leq \beta \leq t$. Otherwise, it equals one. Let $avg(u)$ denote $Acc(\ell)$ averaged over all sites for M^u and $avg(b)$ denote the same for M^b.[6] The number of sites and uniform sampling trials is 5 and 20.

To test the distributed clustering algorithms, we generated a two dimensional dataset consisting of 10000 tuples sampled from a mixture of two Gaussians. Figure 5 illustrates the results. The curve labeled "Linear Binning" is $avg(b)$ and the curve labeled "Uniform Sampling" is $avg(u)$. We see that uniform sampling outperforms linear binning for almost all communication sizes.

5 Conclusions

We considered the problem of Kernel Density Estimation (KDE) on data distributed over a collection of sites. Our work was motivated by the fact that, in some cases, large data repositories are, in their native

[6]When the clusterings are perfectly matched, the accuracy is approximately 0.51 (not zero) – the fraction of tuple pairs appearing in the same cluster among $C_1, \ldots, C_t$.

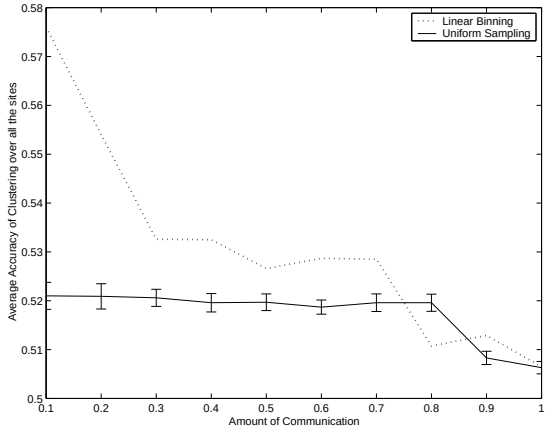

Figure 5: Clustering accuracy comparison on a 10000 tuple, 2D dataset.

form, distributed over many sites, perhaps over a large geographic distance. In environments where communication is limited, centralization of the data can be problematic. We examined a "local model, global combination" approach to estimate the global KDE in a communication-efficient fashion. Moreover, only two rounds of complete synchronization are required to estimate the density of any set of points in the domain. This low synchronization property is useful in data analysis techniques which determine the next point at which the density is to be estimated based on estimations of previous points (*i.e.* density-based clustering).

We examined two realizations of this framework, uniform sampling and linear binning, We conducted experiments measuring the accuracy of each with respect to: L2-norm, classification based on density estimation, and clustering based on density estimation. In all cases we compared the KDE produced by the whole dataset once centralized and the approximation obtained by linear binning or uniform sampling. We observed that uniform sampling outperformed linear binning for classification, clustering, and L2-norm at four dimensions.

In some situations the data held by each site is sensitive and not to be revealed in the plain to others. Before closing the paper, we point out that linear binning can be adapted using secure multi-party computation techniques to offer privacy. The resulting algorithm has some nice properties (i) it has communication complexity independent of $|D|$; (ii) it does not require complete synchronization in approximating $\hat{f}_D(\mathbf{x})$, for each $\mathbf{x} \in X$. In a forthcoming extended version of this paper, we will explicate the details of this algorithm.

Acknowledgments

We thank the U.S. National Science Foundation for support through award IIS-0329143 and CAREER award IIS-0093353. We also thank Kun Liu and Souptik Datta for their valuable contributions.

References

[1] Datta S., Giannella C., and Kargupta H. K-Means Clustering Over a Large, Dynamic Network. In *Proc. SIAM Data Mining*, 2006.

[2] Domeniconi C. and Gunopulos D. An Efficient Density-Based Approach for Data Mining Tasks. *Knowledge and Information Systems*, 6(6):750–770, 2004.

[3] Kargupta H. and Sivakumar K. Existential Pleasures of Distributed Data Mining. In *Data Mining: Next Generation Challenges and Future Directions, MIT/AAAI Press*, pages 3–26, 2004.

[4] Kempe D., Dobra A., and Gehrke J. Computing Aggregate Information using Gossip. In *Proc. IEEE Symp. on Found. of Comp. Sci.*, pages 482–491, 2003.

[5] Kittler J. A Locally Sensitive Method for Cluster Analysis. *Pattern Recognition*, 8:23–33, 1975.

[6] Klusch M., Lodi S., and Moro G. Distributed Clustering Based on Sampling Local Density Estimates. In *Proc. Int. Joint Conf. on Artif. Intel.*, pages 485–490, 2003.

[7] Merugu S. and Ghosh J. Privacy-Preserving Distributed Clustering Using Generative Models. In *Proc. IEEE Conf. on Data Mining*, pages 211–218, 2003.

[8] Merugu S. and Ghosh J. A Distributed Learning Framework for Heterogeneous Data Sources. In *Proc. ACM SIGKDD*, pages 208–217, 2005.

[9] Peleg D. *Distributed Computing: a Locality-Sensitive Approach.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.

[10] Scott D. *Multivariate Density Estimation: Theory, Practice, and Visualization.* John Wiley & Sons, Inc., New York, NY, 1992.

[11] Silverman B. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall, New York, NY, 1998.

[12] Silvestri C. and Orlando S. Distributed Approximate Mining of Frequent Patterns. In *Proc. ACM Symp. on Applied Computing*, pages 529–536, 2005.

[13] Verykois V., Bertino E., Fovino I., Provenza L., Saygin Y., Theodoridis Y. State-of-the-art in Privacy Preserving Data Mining. *SIGMOD Record*, 33(1):50–57, 2004.

[14] Wand M. and Jones M. *Kernel Smoothing.* Chapman & Hall, New York, NY, 1995.

[15] Wolff R. and Schuster A. Association Rule Mining in Peer-to-Peer Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 34(6):2426–2438, 2004.

[16] Zhang X. and Cheung W. Visualizing Global Manifold Based on Distributed Local Data Abstraction. In *Proc. IEEE Conf. on Data Mining*, pages 821–824, 2005.