DISTRIBUTED DISTANCE ESTIMATION FOR MANIFOLD LEARNING AND DIMENSIONALITY REDUCTION

Mehmet E. Yildiz[1], *Frank Ciaramello*[1], *Anna Scaglione*[2]

[1]School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, 14850
[2]Electrical and Computer Engineering Department, University of California Davis, Davis, CA, 95616
{mey7,fmc3}@cornell.edu, ascaglione@ucdavis.edu

ABSTRACT

Given a network of N nodes with the i-th sensor's observation $x_i \in R^M$, the matrix containing all Euclidean distances among measurements $||x_i - x_j|| \ \forall i,j \in \{1,\ldots,N\}$ is a useful description of the data. While reconstructing a distance matrix has wide range of applications, we are particularly interested in the manifold reconstruction and its dimensionality reduction for data fusion and query. To make this map available to the all of the nodes in the network, we propose a fully decentralized consensus gossiping algorithm which is based on local neighbor communications, and does not require the existence of a central entity. The main advantage of our solution is that it is insensitive to changes in the network topology and it is fully decentralized. We describe the proposed algorithm in detail, study its complexity in terms of the number of inter-node radio transmissions and showcase its performance numerically.

Index Terms— Distributed computing, manifold estimation, dimensionality reduction.

I. INTRODUCTION

Given a set of observation vectors $x_i \in \mathbb{R}^M, i \in \{1,\ldots,N\}$, the process of determining the true structure of this data is a challenging one. In most of the real world applications, the phenomenon of interest generates data that are physically constrained to reside on a $L < M$ dimensional non-linear manifold rather than on the M dimensional Euclidean space where all possible measurements lie, and therefore the basic definitions of Euclidean distances or local neighborhoods do not relate appropriately to the different data. Fortunately, in many applications such as speech recognition, visual tracking and image classification, this manifold is continuously differentiable, *i.e*, smooth [1], [2]. In this case, if the data are sampled finely enough over the manifold, then each data point and its neighbors lie on or close to a K dimensional hyperplane which is tangent to the manifold at a given data point. Thus, the manifold distances on those local hyperplanes can be approximated as Euclidean, and these distances can be patched together to characterize the manifold of interest [3]. Similarly, *dimensionality reduction* which aims to represent the set of observation $x_i \in \mathbb{R}^M$ in a lower dimensional space such that internode similarities are preserved, also utilizes the idea of linearity on the local neighborhoods [4], [5], [6].

While the above algorithms ([1]- [6]) perform sufficiently well on real world data, they lack of a feature: decentralization. When the observations are collected by several sensors which are geographically separated, fusing all the observations to a central entity may not be possible or practical. For this reason, Costa *et.al.* proposed a method for the *decentralized* manifold estimation in the context of sensor localization [7]. The proposed decentralized algorithm requires that geographically close sensors' observations are also close on the manifold (so that neighborhood distances can be estimated as Euclidean) from which the observations are sampled. While such an assumption holds true in the context of sensor localization, it is not valid generally when other sensor modality are studied. For instance, in distributed camera networks, physically close cameras may not record correlated data due to angle differences and obstructions.

In this paper, we propose a fully decentralized method for the reconstruction of the distance matrix of the sensor observations such that the distance matrix will be available at all sensors. Our method is based on the idea studied in [8] for the distributed construction of an estimate of the correlation matrix. Once the full (observation) distance matrix is constructed at all of the sensors, each sensor can decide its local (manifold) neighborhood. The neighborhood and distance information can then be used for non-linear dimensionality reduction methods, such as ISOMAP [4], to estimate the geodesic distances over the manifold.

In Section II, we discuss our decentralized method for the reconstruction of the distance matrix at all sensors. In Section III, we utilize our reconstruction method in the context of manifold reconstruction and dimensionality reduction by integrating it to the ISOMAP method. We conclude our study in Section IV.

II. DISTANCE MATRIX ESTIMATION

As mentioned in Section I, we are interested in the case where a network of N sensors, each gathering a single M-dimensional observation, *i.e.*, $x_i \in \mathbb{R}^M, i \in \{1,\ldots,N\}$, aims to estimate the inter-observation distances for the whole network. The sensors communicate via a power limited wireless interface, that allows them to converse only within a limited range R. We assume that the network is dense enough to be strongly connected, *i.e.*, there is a path of wireless links (not necessarily single hop) between each node pair in the network. We make the simplification of assuming that the communications among the neighbors are two-way and noise free and that only nearby interference can cause errors [9]. Denoting the geographical distance between node i and node j as d_{ij}, the neighbor set of node i is $\mathcal{N}_i = \{j \in \{1,\ldots,N\} | d_{ij} \leq R\}$. We note that the neighborhood definition is symmetric, *i.e.*, if $j \in \mathcal{N}_i$ then $i \in \mathcal{N}_j$. For any pairs of nodes $i,j \in \{1,\ldots,N\}$, the distance between the observations of node i and node j is equal to $||x_i - x_j||$, where $||.||$ denotes the $\mathcal{L}^2$ of its argument. If we define an $M \times N$ matrix $X = [x_1, x_2, \ldots, x_N]$ and the $N \times N$ positive definite and real matrix $T = X^T X$, the internode distance between node i and node j is equal to:

$$\begin{aligned} ||x_i - x_j|| &= \sqrt{x_i^T x_i + x_j^T x_j - 2x_i^T x_j} \\ &= \sqrt{T_{ii} + T_{jj} - 2T_{ij}} \end{aligned} \quad (1)$$

Therefore, reconstructing T is a sufficient condition for recovering the inter-node observation distances. If the dimensionality M is very large, seeking an alternative to the exchange of the raw data can be beneficial. We note that eigenvalues of T are real and T is a positive definite matrix. In the rest of the paper, we assume that $0 < S \leq rank(X^H X)$ largest eigenvalues of T are distinct, *i.e.*, $\lambda_1 > \lambda_2 > \ldots > \lambda_S$.

II-A. Iterative Power Method

For the reconstruction of T, we will utilize the distributed algorithm proposed in [8] for reconstructing the sample covariance matrix of the sensor observations of a given network. The algorithm is based on the *power method* which is an iterative method for determining the largest eigenvalue and the corresponding (dominant) eigenvector of a given matrix. In particular, the power method starts with an initial estimate

of the dominant eigenvector of the matrix, and at each iteration the estimate is updated. Mathematically, given an initial estimate of the dominant eigenvector $v_1(0) \in \mathbb{C}^N$ of the matrix X^TX, the algorithm follows the iteration:

$$v_1(k+1) = \frac{X^TXv_1(k)}{||X^TXv_1(k)||} \tag{2}$$

where $k \geq 0$ is the iteration index. It can be shown that under some regularity conditions, *i.e.*, the largest eigenvalue is unique and the initial estimate of the eigenvector has non-zero component in the direction of the dominant eigenvector, it can be shown that $v_1(k)$, $k \geq 0$, is bounded and thus there exists a subsequence of $v_1(k)$ which converges to a multiple of the dominant eigenvector of the matrix X^TX [10]. Moreover, since X^TX is symmetric, its eigenvalues are real and therefore the whole sequence $v_1(k)$ converges to a multiple of the dominant eigenvector of X^TX. If we denote this limit as $v_1^\star$, *i.e.*, $\lim_{k\to\infty} v_1(k) = v_1^\star$, then the corresponding eigenvalue is simply $\lambda_1 = (v_1^\star)^TX^TXv_1^\star/||v_1^\star||^2$. Once $v_1^\star$ and λ_1 are calculated, the second largest eigenvalue λ_2 and the corresponding eigenvector $v_2^\star$ can be found by utilizing the power method on $X^TX - \lambda_1 v_1^\star (v_1^\star)^T/||v_1^\star||$ provided that the regularity conditions are satisfied. Similarly, λ_k and $v_k^\star$ can be calculated by running the power method on $X^TX - \sum_{l=1}^{k-1} \lambda_l v_l^\star (v_l^\star)^T/||v_l^\star||$.

II-B. Distributed Power Method via Consensus

At this point, it is clear that the power method performs the eigenvalue matrix decomposition, and in the following we discuss how such an algorithm can be completely decentralized, providing an identical distance matrix to all node as the algorithm iterates, and using only near neighbors communications. If we focus on the k-th iteration of the power method in (2), the i-th element of the $v_1(k)$ is equal to:

$$[v_1(k)]_i = \alpha_{k-1} \sum_{l=1}^{M} [x_i]_l \left(\sum_{j=1}^{N} [x_j]_l [v_1(k-1)]_j \right), \tag{3}$$

where $\alpha_{k-1} \triangleq ||X^TXv_1(k-1)||^{-1}$. We note that $[x_i]_l$,$1 \leq l \leq M$, is the i-th sensor observation and thus already known to sensor i. If the norm of the estimate at time k and the inner summation in (3) can be calculated in a decentralized fashion for all $1 \leq l \leq M$, then $[v_1(k)]_i$ can be reconstructed at sensor i.

Average Consensus Algorithms: At this point of the algorithm, we rely on the so called synchronous average consensus methods, namely, linear iterative average consensus. Consider a set of (N) nodes where each node stores a real scalar value $z_i(0) \in \mathbb{R}$ where i denotes the node index. Average consensus is a distributed method allowing all nodes to compute the average of the initial states ($\bar{z}(0) = 1/N \sum_{i=1}^{N} z_i(0)$) in an iterative fashion via only near neighbors' communications. We consider in our analysis the *synchronous* linear consensus algorithms where every sensor, simultaneously, updates its own state value by a weighted sum of differences between its neighbors' values and its own value:

$$z_i(k+1) = z_i(k) + \sum_{j \in \mathcal{N}_i} W_{ij} \left(z_j(k) - z_i(k) \right) \tag{4}$$

where W is the weight matrix with non-negative entries and $\mathcal{N}_i$ is the neighbor set of node i, *i.e.*, node $j \in \mathcal{N}_i$ if $d_{ij} \leq R$. In this case, the network-wide update is given by

$$z(k+1) = Wz(k) = W^{k+1}z(0), \tag{5}$$

where $z(k) = [z_1(k) \dots z_N(k)]^T$. Under the conditions that

$$W \geq 0, \mathbf{1}^TW = \mathbf{1}^T,\ W\mathbf{1} = \mathbf{1},\ \rho(W - \mathbf{1}\mathbf{1}^T/N) < 1 \tag{6}$$

where $\rho(\cdot)$ denotes the spectral radius of its argument and $\mathbf{1}$ is the all ones vector, it has been shown that [11]:

$$\lim_{k\to\infty} z(k) = \bar{z}(0)\mathbf{1}. \tag{7}$$

In other words, if the update weights W satisfy the above conditions, as the number of iterations grows, each node's state value converges to the initial average. Interestingly, if the first three conditions are satisfied, the fourth condition is equivalent to the network connectivity condition, *i.e.*, it is satisfied if and only if the network is strongly connected. Since we are only interested in strongly connected network, the entries of the W matrix can be chosen (offline) such that the rest of the conditions are satisfied. Thus, such an algorithm is guaranteed to converge to the initial global average via only local neighbor communication.

We are going to utilize the average consensus algorithm to calculate (3) in a distributed way. Let us focus on the scenario where, at $k = 0$, each sensor randomly determines the value of the corresponding entry of $v_1(k)$, *i.e.*, sensor j assigns a value to $[v_1(0)]_j \in \mathbb{R}$. Then, each sensor j can calculate $[x_j]_l[v_1(0)]_j$ locally for a given l since it has access to both the local observation and the j-th entry of $v_1(0)$. If we multiply this quantity by N and denote this as the initial observation of node j as in the average consensus problem, *i.e.*, $z_j(0) = N[x_j]_l[v_1(0)]_j$, then by running the average consensus algorithm

$$\bar{z}(0) = 1/N \sum_{j=1}^{N} z_j(0) = \sum_{j=1}^{N} [x_j]_l [v_1(0)]_j, \tag{8}$$

can be determined at each node. The decentralization of the power method we propose is based on computing the inner summation in (3) as the limit (8) of a consensus averaging session among the sensors. In particular, M average consensus algorithms can be run in parallel to calculate the above summation for all $1 \leq l \leq M$, diffusing the information that each node locally needs to compute (3). In fact, once these M summations are determined, each node i can calculate $[v_1(1)]_i$ by simply calculating the inner products of its own observation and reconstructed summations. Moreover, the norm of the current estimate,*i.e.*, $||v(1)|| = \sqrt{\sum_{j=1}^{N} [v(1)]_j^2}$ can be calculated with one more average consensus since each node has access to the corresponding entry of the estimate. We note that in this case $z_j(0) = N[v(1)]_j^2$. Thus, we have completed a single iteration of the power method such that the j-th entry of the eigenvector is available at the node j. This algorithm is to be repeated for each iteration of the power algorithm.

For sufficiently large k, $v_1(k)/||v_1(k)||$ will closely estimate the dominant eigenvector of X^TX. If we denote the stopping time of the algorithm as $K \geq 0$, $u_1 = v_1(K)/||v_1(K)||$, the largest eigenvalue of X^TX can be estimated as:

$$\lambda_1 = u_1X^TXu_1. \tag{9}$$

We note that the vector X^TXu_1 is the same as (3) except the fact that there is no α factor in front. Thus, as we have discussed above, it can be calculated via average consensus such that $[X^TXu_1]_j$ is available at sensor j. Since we can rewrite (9) as

$$\lambda_1 = \sum_{j=1}^{N} [u_1]_j [X^TXu_1]_j, \tag{10}$$

and both of the summands are available at sensor j, this summation can also be calculated in a distributed way by initializing the consensus algorithm by $z_j = N[u_1]_j[X^TXu_1]_j$. Therefore, λ_1 is constructed in a distributed way.

Once u_1 and λ_1 are calculated, one can run the power method on $X^HX - \lambda_1 u_1 u_1^T$ to determine λ_2 and u_2 as we have discussed in Section II-A. Parallel to our discussion above, we can also show that these quantities can be estimated in a distributed way. If we denote $v_2(k)$ as the estimate of the second dominant eigenvector of X^TX at the k-th step of the power method:

$$v_2(k) = \alpha_{k-1} \left(X^HX - \lambda_1 u_1 u_1^T \right) v_2(k-1), \tag{11}$$

where $v_2(k-1)$ is the previous estimate and α_{k-1} is the normalization constant as mentioned before. As we have discussed above, the first term of the equation, *i.e.*, $X^HXv_2(k-1)$ can be calculated in a distributed way such that i-th entry of the output vector is available

at node i. If we denote the second term as c_2, then $[c_2]_i$ is equal to:

$$[c_2]_i = \lambda_1 [u_1]_i \sum_{j=1}^{N} [u_1]_j [v_2(k-1)]_j. \quad (12)$$

We note that $[u_1]_j$ is already available at sensor j since it has been already estimated and $[v_2(k-1)]_j$ is also available from the previous step. Then, initializing the consensus method by $z_j = N[u_1]_j[v_2(k-1)]_j$, the summation in (12) can be calculated. Therefore, each node i can determine the corresponding value of the vector c. At this point, we completed one iteration of the power method for calculating the second largest eigenvalue and the corresponding eigenvector of $X^T X$. In a similar manner, one can estimate the S largest (significant) eigenvalues and the corresponding eigenvectors in a distributed way. We note that $S \leq rank(X^H X) = \min(N, M)$. In practice, due to the nature of the observed data, one can expect $S << \min(N, M)$.

Once S most significant eigenvalues and eigenvectors are constructed, we mainly need to distribute the entries of these eigenvectors since i-th entries of these eigenvectors are only available at node i. Of note is that we only need to distribute S parameters per sensor which is expected to be much less than the dimension of the original observation space, M. The distribution of the final S parameters can be done by multicasting or flooding. Notice that, when a message containing these S parameters is received, the nodes have to determine where the message is originated from and to which eigenvector this entry belongs to, to have an identical map. This can be achieved by embedding such information into the packet headers.

Constructing S most significant eigenvectors, each node can estimate the inner product matrix as

$$\hat{T} = \sum_{i=1}^{S} \lambda_i u_i u_i^T. \quad (13)$$

Once T is constructed, internode distances are calculated by (1).

II-C. Algorithm Complexity

In this section, we explore the complexity of the algorithm in terms of the total number of radio transmissions for the reconstruction of the distance matrix. For analysis purposes, we assume that the nodes are distributed randomly on a 2-dimensional unit torus and the neighborhood radius is chosen as $\Theta(\sqrt{\log N/N})$ where $\Theta(.)$ is the asymptotic tight bound if its argument. We note that such modeling has been shown to closely reflect the behavior of the wireless sensor networks [9] and this particular choice of the radius guarantees that the network is strongly connected with high probability [12]. It has been shown that for random geometric graphs, consensus algorithms requires $O(N^{2.5}/\sqrt{\log N})$ radio transmissions to achieve the true average to accuracy of $1/N^a$ where $O(.)$ is the asymptotic upper bound of its argument [13]. On the other hand, for each iteration of the power method in (3), one needs $M+1$ parallel consensus algorithms. Moreover, one needs to run the power method for each eigenvector, *i.e.*, total of S times. Thus, the complexity of the first part is $O\left(N^{2.5}/\sqrt{\log N}(M+1)SK\right)$ where K is the stopping time for the power method. In the second part, S reconstructed eigenvectors has to be multicasted over the network. Since the $R = \Theta(\sqrt{\log N/N})$, the number of hubs to get from one corner to the diagonally opposite corner is $\Theta(\sqrt{N/\log N})$. Therefore, the number of radio transmissions required for the second part is $O\left(SN\sqrt{N/\log N}\right)$. Combining the first and the second part, the overall complexity of the algorithm is:

$$O\left(S(M+1)K\ N^{2.5}/\sqrt{\log N} + S\ N^{1.5}/\sqrt{\log N}\right) \quad (14)$$

We note that both with respect to the network size N and the initial observation dimension M, the dominant term in the complexity of the algorithm is due to calculating eigenvalues and eigenvectors in a distributed way. Since the number of significant eigenvectors are expected to be much less than the dimension of the observation space, distributing this information is computationally light. While it is true that the proposed algorithm may have a higher complexity than simply multicasting initial observations, we only need to distribute $S << M$ parameters per sensor and the first part of the algorithm (power method) is robust to both dynamic topologies and link failures.

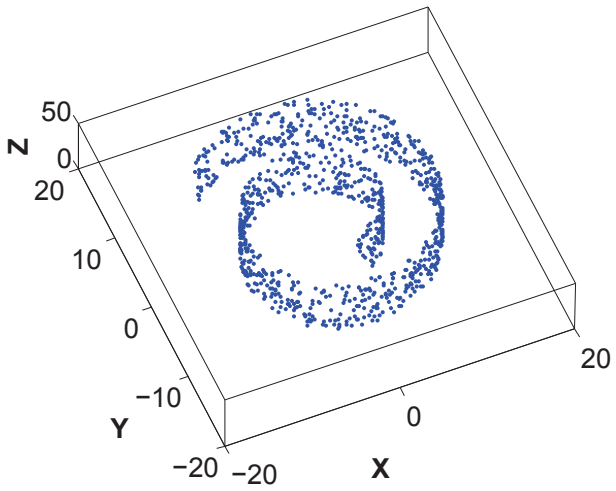


Fig. 1. Original observation in 3-dimensional Euclidean space.

III. SIMULATION

In this section, we utilize the proposed algorithm to estimate the distances among the observations of a random network and then to reduce the dimensionality of these observations. Our network consists of $N = 1000$ nodes which are randomly distributed on an 1×1 square, and the connectivity radius is chosen as $\sqrt{\log(N)/N}$. The observations of the sensors are in 3 dimensional Euclidean space and also on a Swiss roll. Of note is that a Swiss roll can be fully described in 2 dimensions. The sensor observations are given as in Fig. 1. We also note that the sensors and the observations are randomly matched, therefore two close by points on the manifold do not necessarily belong to two physically neighboring nodes.

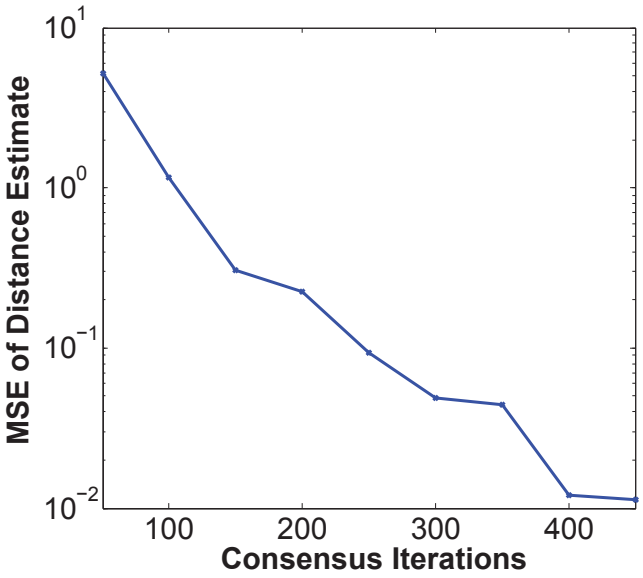
In the decentralized distance estimation procedure, there is a tradeoff between the accuracy of the estimate and the complexity of the algorithm. The stopping time for the power method K and the number of consensus iterations L should be chosen such that the algorithm can be executed in a practical amount of time and the distance estimate is accurate enough for the ISOMAP algorithm to be able to identify the underlying manifold. Fig. 2(a) shows the behavior of the mean squared distance between true distance matrix D and the estimated distance matrix $\hat{D}$ with respect to the number of consensus iterations for a fixed value of L and S in a log scale. Mathematically speaking, the MSE is defined as:

$$\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(D_{ij} - \hat{D}_{ij}\right)^2. \quad (15)$$

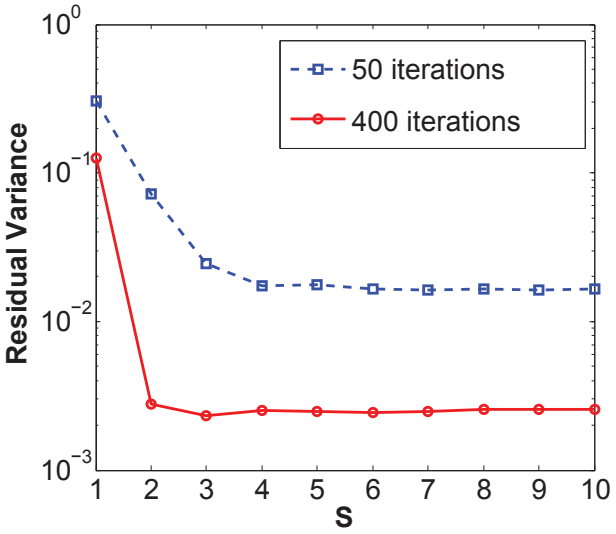
The distance estimate converges at approximately 400 consensus iterations, as illustrated in Fig. 2(a). Moreover in this particular simulation only 5 iterations of the power method were required to achieve an MSE of 10^{-3} between the true distances and the estimated distances.

Fig. 2(b) shows the residual error between the estimated data points and the original data with respect to the number of number of significant eigenvectors that are reconstructed. If the data were analyzed by a central entity, one would expect that the residual error would saturate at $S = 2$ since this is the true dimension of the underlying manifold, *i.e.*, Swiss roll [4]. While with our proposed method one can still determine $S = 2$ correctly when the number of iterations is relatively large, as this number decreases, the extra error carried out in the system results in the wrong estimation of S as in Fig. 2(b).

In Fig. 3, we have reconstructed the sensor observations on a 2 dimensional space by using centralized ISOMAP and the proposed



(a) number of consensus iterations vs (log) MSE between the true distance matrix and the estimated distance matrix.



(b) Plot of residual variance after ISOMAP dimensionality reduction.

Fig. 2.

method. The parameters were chosen as $S = 2$, $K = 50$ and the number of consensus iterations to be equal to 500. The centralized ISOMAP is assumed to be true reconstruction. Once can clearly see that decentralized algorithm has little performance loss in terms of its ability to accurately identify the low-dimensional manifold.

IV. CONCLUSION

In this paper, we have proposed a decentralized method for reconstructing distances among the sensor observations on a given network. Unlike the existing literature, our method neither requires an existence of a central entity nor relies on the assumption that geographically close sensors' observations are also close on the underlying manifold. Our iterative power method and average consensus based decentralized algorithm reconstructs the distance matrix at each sensor so that each node in the network can reconstruct and perform dimensionality reduction individually. Furthermore, we have simulated the performance characteristics of our method and showed that even under highly practical settings, its performance is comparable to the centralized setting.

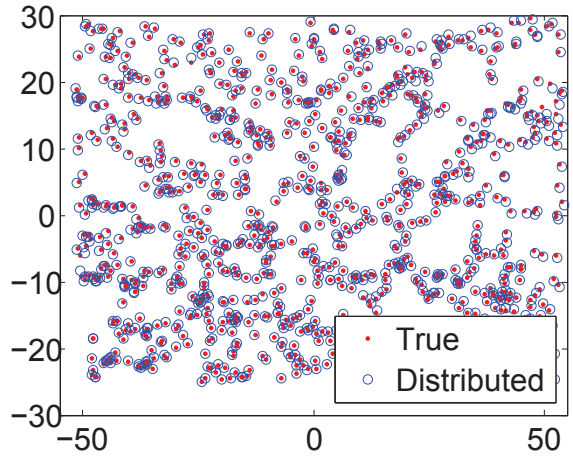


Fig. 3. ISOMAP output with true distance matrix and estimated distance matrix.

V. REFERENCES

[1] C. Bregler and S. Omohundro, "Nonlinear manifold learning for visual speech recognition." Proc. of IEEE Conference on Computer Vision, 1995.

[2] Q. Wang, G. Xu, and H. Ai, "Learning object intrinsic structure for robust visual tracking." Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.

[3] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[4] J. Tenembaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2322, 2000.

[5] O. C. Jenkins and M. J. Mataric, "A spatio-temporal extension to isomap nonlinear dimension reduction." Proc. of International Conference on Machine Learning, 2004.

[6] M. H. Law, , and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 377–391, 2006.

[7] J. A. Costa, N. Patwari, and A. O. H. Iii, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, pp. 39–64, 2006.

[8] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance." To Appear at Proc. of Asilomar Conference on Signals, Systems and Computers, 2008.

[9] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, March 2000.

[10] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, 2006.

[11] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[12] S. Boyd, A. Gosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, June 2007.

[13] T. C. Aysal, M. Yildiz, and A. Scaglione, "Broadcast consensus," *To Appear in IEEE Transactions on Signal Processing*, March 2008.