A Fast and Effective Multidimensional Scaling Approach for Node Localization in Wireless Sensor Networks

Master Thesis

By

Giorgos Latsoudas

Submitted to the Department of Electronic Engineering & Computer Engineering in partial fulfillment of the requirements for the Master Degree.

Technical University of Crete

Advisor: Professor Sidiropoulos Nikolaos

Co-advisor: Associate Professor Liavas Athanasios

Co-advisor: Associate Professor Potamianos Alexandros

August 2006

CONTENTS

ABSTRACT

Given a set of pairwise distance estimates between nodes, it is often of interest to generate a map of node locations. This is an old nonlinear estimation problem that has recently drawn interest in the signal processing community, due to the emergence of wireless sensor networks. Sensor maps are useful for estimating the spatial distribution of measured phenomena, and for routing purposes. We propose a two-stage algorithm that combines algebraic initialization and gradient descent. In particular, we borrow an algebraic solution known as *Fastmap* from the database literature and adapt it to the sensor network context, using a specific choice of anchor/pivot nodes. The resulting estimates are fed to a gradient descent iteration. The overall algorithm offers very competitive performance at significantly lower complexity than existing solutions of comparable complexity order. For a multiplicative normal measurement noise model that is often adopted in the relevant literature, we also derive the pertinent Cramér-Rao bound (CRB). Simulations indicate that the performance of our algorithm is close to the CRB when the network is (close to) fully connected, in the sense that every node can estimate its distance from all (most) other nodes.

1. INTRODUCTION

The problem of node localization from pairwise distance estimates has recently attracted interest in the signal processing community, owing to the growing interest in wireless sensor networks [1, 2, 4, 6, 7]. Given a matrix of pairwise distances, the localization problem aims to determine the (*relative*) node locations that generate these distances. In other words, one seeks a map of node locations with a given (approximate) distance structure. This is a classic problem originating in psychometrics [8, 9], known as *Multi-Dimensional Scaling* (MDS) [5]. There are many MDS flavors and variants; perhaps the single most important one is *metric MDS*.

The classical approach to solving MDS is based on computing the principal components of a double-centered version of the matrix of squared distances. This works reasonably well (albeit not optimally in the least squares sense, due to the double centering), but its complexity is cubic in the number of nodes, and thus does not scale well with network size. A popular alternative to principal component analysis (PCA) is the use of gradient descent or other numerical optimization tools that aim to optimize a *stress function* of the error between the measured distances and those reproduced by a given configuration of points.

The drawback of gradient descent and related approaches is that they require accurate initialization, due to the multi-modal nature of the stress function.

We propose a two-stage MDS algorithm that employs an algebraic initialization procedure followed by gradient descent. The algebraic initialization step is based on the Fastmap algorithm [3], borrowed from the database literature. Fastmap is a linear-complexity mapping tool, which is, however, sensitive to measurement errors. This is not particularly relevant in the database context; therein actual distances are computed from complete representations. Noise sensitivity is an important issue in wireless ranging applications, due to shadowing, fading, and the use of approximate path loss models.

Due to the fact that distances are invariant to coordinate frame transformations (rotation, reflection, shift), there is a need to employ three so-called *anchor nodes*, whose position is accurately known (e.g., via GPS) in order to fix a desired coordinate frame. Unfortunately, Fastmap is very sensitive to coordinate alignment, because the estimated position of every node (and thus anchor nodes as well) is only based on distances to selected *pivot* nodes - there is no averaging. In order to mitigate this problem, we advocate a judicious choice of anchor/pivot nodes, placed at the outer edges of the network. This placement bypasses the need for alignment and thus alignment errors, thereby providing a higher-quality initialization to the gradient descent. The overall algorithm affords better localization accuracy than PCA-based MDS, at substantially lower complexity cost (quadratic in the number of nodes). Our algorithm is also competitive with re-

spect to recent solutions of the same complexity order, developed specifically for node localization in sensor networks [2]. For relatively dense networks, our algorithm yields comparable estimation performance at a significantly reduced complexity relative to [2], even when the latter is initialized using our adaptation of Fastmap. An exception is the case of very sparse networks, wherein [2] with Fastmap initialization may be preferable when accuracy is more important than complexity.

We remark that there are other algorithms in the recent literature that assume a different measurement model (e.g., 0-1 node connectivity information only, as in [7]), or propose solutions of considerably higher complexity (e.g., as in [1]). We aim for the low-complexity regime, for simplicity and scalability considerations.

The rest of this paper is structured as follows. Section 2 summarizes the standard MDS algorithms: PCA-based, and gradient descent. Section 3 contains a brief review of the Fastmap algorithm. Section 4 presents the proposed two-stage algorithm, and section 5 summarizes Costa's algorithm [2]. Section 6 presents the CRB for a multiplicative normal measurement noise model that is often adopted in the literature on node localization in sensor networks [1, 7]. Section 7 contains extensive simulation results illustrating the performance of the above algorithms and the CRB, under various scenarios and measurement noise models. Conclusions are drawn in section 8.

2. MULTIDIMENSIONAL SCALING

MDS [8, 9, 5] has its origins in psychometrics and psychophysics. MDS postulates that perceptual or objective "dissimilarities" or "distances" between pairs of abstract "objects" can be be generated by points in m-dimensional space. Any set of distances obeying the triangle inequality can be reproduced (or closely approximated) by choosing m to be sufficiently large; but usually $m = 2$ or $m = 3$ is chosen to retain the systematic variation, and also for ease of visualization. Thus, MDS aims to find a geometric configuration of points in 2-D or 3-D space, such that the distances between these points fit as well as possible the given dissimilarity information.

We denote the dissimilarity measure (the estimated distances in our case), between objects i and j as d_{ij}. The set of dissimilarities yields a measured distance matrix $\mathbf{D}$. We also let $\hat{d}_{ij}$ denote the Euclidean distance between (generated by) two points $X_i = (x_{i1}, x_{i2}, ..., x_{im})$ and $X_j = (x_{j1}, x_{j2}, ..., x_{jm})$, i.e.

$$\hat{d}_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2}. \tag{2.1}$$

In classical metric MDS, we estimate the node coordinates $\mathbf{X}$ by computing the m principal components of an element-wise squared and double-centered

version of the matrix $\mathbf{D}$, denoted by $\mathbf{B}$:

$$B = -\frac{1}{2}\mathbf{JPJ}, \tag{2.2}$$

where $\mathbf{P} = \mathbf{D} \odot \mathbf{D}$ is the matrix of squared distances ($\odot$ denotes the element-wise matrix product), and $\mathbf{J}$ is the centering operator,

$$\mathbf{J} = \mathbf{I} - \mathbf{e}\mathbf{e}^T/N, \tag{2.3}$$

with N denoting the number of objects (sensors / nodes), and $\mathbf{e}$ denoting the $N \times 1$ vector of all 1's . For an $N \times N$ matrix $\mathbf{D}$ and for m dimensions, it can be shown that

$$-\frac{1}{2}(d_{ij}^2 - \frac{1}{N}\sum_{j=1}^{N} d_{ij}^2 - \frac{1}{N}\sum_{i=1}^{N} d_{ij}^2 + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N} d_{ij}^2) = \sum_{k=1}^{m} x_{ik}x_{jk}, \tag{2.4}$$

thus the node coordinates can be estimated from the m principal eigenvectors of the matrix $\mathbf{B}$, scaled by the square roots of the corresponding eigenvalues. That is, with $\mathbf{U}_r$ containing the m principal eigenvectors and $\mathbf{V}_r$ diagonal containing the corresponding eigenvalues, $\mathbf{B}_r = \mathbf{U}_r\mathbf{V}_r\mathbf{U}_r^T$ is an optimal least squares approximation of $\mathbf{B}$, and $\mathbf{X}_r = \mathbf{U}_r\mathbf{V}_r^{1/2}$ is an approximation of the node coordinates in m-dimensional space, up to a common coordinate rotation, reflection, and shift. An alignment procedure is necessary to transform the estimated node locations to a desired frame of reference.

It is important to note that, due to the preprocessing steps prior to PCA, this approach is not equivalent to nonlinear least-squares parameter fitting using the original measurements.

Direct minimization of a suitable *stress function* is an alternative to PCA-based MDS [8]. A common[1] stress function is

$$stress^2 = \sum_{i,j} w_{ij}(\hat{d}_{ij} - d_{ij})^2. \tag{2.5}$$

Where $[w_{ij}]$ is the weight matrix, whose elements are equal to 1 if node j is in the measurement range of node i and 0 otherwise. Minimization starts with an initial guess of the node positions (often random), followed by gradient descent iterations. Initialization matters a lot in this context, because the stress function is multi-modal. Furthermore, the number of iterations required for convergence depends on the quality of the initialization.

[1] The negative log-likelihood of the observed data under a suitable measurement noise model would seem to be the natural choice of stress function. This is not fortuitus in our context, however, because the resulting function is not only multi-modal, but also leads to numerical difficulties. For this reason, a least squares criterion is preferred. While still multi-modal, the adopted least squares criterion is much more benign from a numerical optimization viewpoint, and it often yields performance close to the pertinent CRB, as will be seen in the simulations.

3. FASTMAP

The basic element of Fastmap [3] is the projection of the nodes on a properly selected line. This is achieved by selecting two objects O_a, O_b, called *pivots*, and projecting all other objects on the line that passes through them. A pair of pivots is chosen for each of the m dimensions. The coordinates, (i.e., projections on the pivot line) of the objects are found by employing the *cosine law* [3]. Thus, the first coordinate for object O_i is given by:

$$x_i = \frac{d_{ai}^2 + d_{ab}^2 - d_{bi}^2}{2d_{ab}}, \tag{3.1}$$

where d_{ij} is the measured distance between nodes i and j and a, b are the pivot objects. After computing these coordinates for each object O_i, we consider a hyperplane which is orthogonal to the pivot line. We then project the objects on this hyperplane, and repeat the process, this time using

$$\tilde{d}_{ij}^2 = d_{ij}^2 - (x_i - x_j)^2, \quad i, j = 1, ..., N. \tag{3.2}$$

A heuristic method is proposed in [3] for choosing the pivots as far as possible from one another.

In database applications there is no "natural" or preferred coordinate frame of reference, thus the final alignment step is not used, and anchors are not needed.

In the context of sensor networks, however, obtaining absolute position estimates is important. Unfortunately, Fastmap is very sensitive to coordinate alignment, because the estimated position of every node (and thus anchor nodes as well) is only based on distances to the chosen pivot nodes - there is no averaging. In order to mitigate this problem, we advocate a particular choice of anchor nodes that double as pivots, placed at the outer edges of the network. In particular, we assume that the sensor nodes are spread over a square, and place the anchor nodes, which will also serve as pivots, at three vertices (see Fig. 3.1). This placement bypasses the need for alignment and thus alignment errors, thereby providing a high-quality initialization to the gradient descent. Anchors #1 and #2 also serve as pivots for determining the coordinates in the first dimension, while anchors #2 and #3 double as pivots for the second dimension.

We assume that the anchor/pivot nodes which are used by Fastmap can take distance measurements from all the sensor nodes, (even if we don't have full connectivity for the rest of the nodes). This is reasonable if the anchor/pivot nodes are airborne, in higher ground, or on a mast.

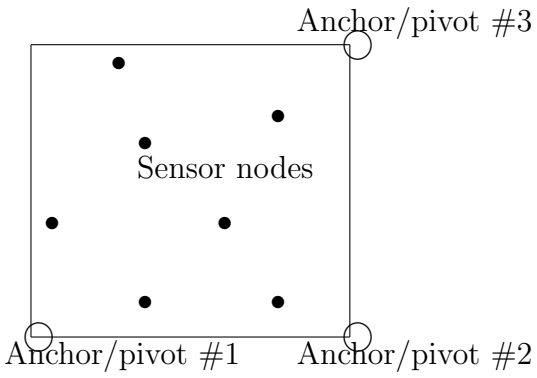

Fig. 3.1: Anchor/pivot node placement

4. PROPOSED TWO-STAGE APPROACH

Fastmap is a fast algebraic mapping method that is rather sensitive to measurement errors, particularly so in the final alignment step. In our context, this sensitivity can be mitigated by the proposed choice of anchor/pivot nodes. The resulting estimates can be used as initialization for gradient descent. Each step of gradient descent costs $\mathcal{O}(N^2)$. Assuming good-enough initialization, only a few gradient descent steps will be needed. This suggests that a substantial complexity reduction relative to PCA and other techniques is possible. Interestingly, estimation accuracy can be improved as well, as we will see.

The basic steps of the two-stage algorithm are shown in Table 4.1. Denoting by (x_i, y_i) the estimated position of node i, the partial derivative of the stress function in (5) is given by

$$\frac{\partial stress}{\partial x_i} = \sum_{j \neq i} w_{ij} \frac{(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - d_{ij})(x_i - x_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}. \tag{4.1}$$

with a similar expression for the partial derivative with respect to y_i. For simplicity, but also to bound complexity, a fixed small number of gradient descent steps (denoted by p) is used in our simulations.

Tab. 4.1: Two-stage Fastmap+SD Algorithm

Input: **D**

1. Run Fastmap using as pivots three anchor nodes, judiciously placed on the three vertices of the square distribution area. Let X be the vector containing the resulting estimated node coordinates.

2. For $i = 1$ to p

   begin

   - evaluate $\nabla stress$ at the point X
   - $X = X - \lambda \nabla stress$

   end

5. COSTA'S ALGORITHM

An iterative distributed estimation algorithm for MDS has been recently proposed in [2], using the principle of *majorization*. The idea behind majorization is simple. Instead of directly minimizing a complicated cost/stress function, majorization uses a simpler (usually quadratic) *majorizing* function that lies over the said cost/stress function and is equal to it at the current parameter estimate. Minimizing the majorizing function thus yields a new parameter estimate whose cost/stress is lower than or equal to that of the previous one. Continuing in this fashion yields a sequence of parameter estimates of decreasing cost/stress values.

The particular cost function proposed in [2] is

$$S = \sum_{i=1}^{N-M} S_i, \tag{5.1}$$

where the local cost functions S_i are given by:

$$S_i = \sum_{j=1, j\neq i}^{N-M} w_{ij}(d_{ij} - \hat{d}_{ij})^2 + \sum_{j=N-M+1}^{N} 2w_{ij}(d_{ij} - \hat{d}_{ij})^2, \tag{5.2}$$

and the last M nodes are the anchors ($M = 3$ in the 2-D case), by convention.

Specializing the principle of majorization to the present context [2] yields the following update

$$\mathbf{x}_i^k = a_i \mathbf{X}^{k-1} \mathbf{b}_i^{k-1}, \tag{5.3}$$

where $\mathbf{X}^k$ is a matrix which contains the position estimates for all the sensor nodes in the kth iteration of the algorithm, a_i is given by

$$a_i^{-1} = \sum_{j=1, j\neq i}^{N-M} w_{ij} + \sum_{j=N-M+1}^{N} 2w_{ij}, \tag{5.4}$$

and the entries of the $N \times 1$ vector $\mathbf{b}_i$ are given by

$$\begin{aligned} \mathbf{b}_i(j) &= w_{ij}(1 - d_{ij}/\hat{d}_{ij}), \quad j \leq N-M, j \neq i, \\ \mathbf{b}_i(j) &= 2w_{ij}(1 - d_{ij}/\hat{d}_{ij}), \quad j > N-M, j \neq i \\ \mathbf{b}_i(i) &= \sum_{i=1}^{N-M} w_{ij} d_{ij}/\hat{d}_{ij} + \sum_{j=N-M+1}^{N} 2w_{ij} d_{ij}/\hat{d}_{ij}, \end{aligned} \tag{5.5}$$

where $\hat{d}_{ij}$ is the reproduced distance computed from the coordinate estimates at iteration k. The algorithm can be executed in a distributed fashion (every node computes its own position coordinates and the corresponding part of the cost function). When the difference between the previous and the current cost values becomes smaller than a threshold ϵ the algorithm terminates. This is guaranteed due to the fact that a single iteration can reduce or maintain, but cannot increase the cost, which is also bounded from below.

6. MEASUREMENT NOISE MODEL AND CRAMÉR-RAO BOUND

Pairwise distance estimates will inevitably contain measurement errors, which are typically amplified with increasing distance between nodes. The choice of measurement noise model depends on many factors, and is application-specific. We shall adopt a certain multiplicative normal noise model from the recent literature on node localization in wireless sensor networks [1, 7], in which the distance measurement error is proportional to the actual distance between the pair of nodes. Thus the measured distance d_{ij} between nodes i, j is assumed to be drawn from

$$d_{ij} \sim \delta_{ij} + \delta_{ij}\mathcal{N}(0, e_r^2), \tag{6.1}$$

where δ_{ij} is the actual distance between nodes i, j, and $\mathcal{N}(0, e_r^2)$ denotes a zero-mean normal random variable of variance e_r^2 (henceforth referred to as *range error variance*). We also assume that the measurements are reciprocal (or symmetrized by averaging prior to further processing); i.e., $d_{ij} = d_{ji}$.

In this section, we derive the Cramér-Rao Bound (CRB) for node localization using the above multiplicative normal noise model. Analogous derivations for different noise models employed in [2], [6] can be found in [6]. An explanation of

the difference between the received signal strength (RSS) noise model described therein and our multiplicative noise model can be found in the appendix.

Define the vector of sensor parameters $\boldsymbol{\gamma} = (\boldsymbol{\gamma_1}\boldsymbol{\gamma_2}...\boldsymbol{\gamma_N})$. Each $\boldsymbol{\gamma_i}$ contains the location coordinates for node i, i.e., $\boldsymbol{\gamma}_i = (x_i,\ y_i])$ in the 2-D case. The unknown parameter vector for the $N-3$ sensors whose locations are unknown[1] is defined as $\boldsymbol{\theta} = (\boldsymbol{\theta}_x\ \boldsymbol{\theta}_y)$, with $\boldsymbol{\theta}_x = (x_1, x_2, ..., x_{N-3})$ and $\boldsymbol{\theta}_y = (y_1, y_2, ..., y_{N-3})$. This is the vector we wish to estimate. Sensors i, j perform pairwise observations d_{ij}. We assume that the observations d_{ij} are statistically independent for $i < j$. The density function of the observations d_{ij} given the locations of nodes i, j is denoted by $f(d_{ij}|\gamma_i,\ \gamma_j)$. Thus the joint log-likelihood is

$$\begin{aligned} l(\mathbf{D}, \boldsymbol{\gamma}) &= \sum_{i=1}^{N} \sum_{j \in H(i),\ j<i} l_{i,j}, \\ l_{i,j} &= log f(d_{ij}|\gamma_i, \gamma_j), \end{aligned} \tag{6.2}$$

where $H(i)$ is the set of nodes which are in the range of node i.

The CRB for coordinate θ_i is $\text{cov}(\theta_i) \geq [{\mathbf{F}_\theta}^{-1}]_{ii}$, where $\mathbf{F}_\theta$ is the Fisher Information Matrix (FIM), given by

$$\mathbf{F}_\theta = \begin{bmatrix} \mathbf{F_{xx}} & \mathbf{F_{xy}} \\ \mathbf{F_{xy}^T} & \mathbf{F_{yy}} \end{bmatrix}. \tag{6.3}$$

The elements for the sub-matrix $\mathbf{F_{xx}}$ are given by

$$\mathbf{F_{xx}}(k,l) = \begin{cases} -\sum_{j \in H(k)} E[\frac{\partial^2}{\partial x_k^2} l_{k,j}], & k = l \\ -I_{H(k)}(l) E[\frac{\partial^2}{\partial x_k \partial x_l} l_{k,l}], & k \neq l \end{cases}, \tag{6.4}$$

[1] In the 2-D case we need 3 anchor nodes.

where $I_{H(k)}(l)$ is the indicator function (1 if l is in the range of k, 0 otherwise). Similar expressions hold for the $\mathbf{F_{xy}}, \mathbf{F_{yy}}$ sub-matrices. For the model in (6.1) these specialize to

$$\mathbf{F_{xx}}(k,l) = \begin{cases} -\sum_{j\in H(k)} \frac{2(x_k-x_j)^2}{\delta_{kj}^4} - \frac{1}{\delta_{kj}^2} - \frac{e_r^2+1}{e_r^2}(-\frac{1}{\delta_{kj}^2} + \\ 4\frac{(x_k-x_j)^2}{\delta_{kj}^4}) - \frac{1}{e_r^2}(\frac{1}{\delta_{kj}^2} - 3\frac{(x_k-x_j)^2}{\delta_{kj}^4}), \quad k=l, \\ \\ -I_{H(k)}(l)(\frac{1}{\delta_{kl}^2} - 2\frac{(x_k-x_l)^2}{\delta_{kl}^4} + \frac{1+e_r^2}{e_r^2}(4\frac{(x_k-x_l)^2}{\delta_{kl}^4} - \\ \frac{1}{\delta_{kl}^2}) - \frac{1}{e_r^2}(3\frac{(x_k-x_l)^2}{\delta_{kl}^2} - \frac{1}{\delta_{kl}^2})), \quad k\neq l, \end{cases} \tag{6.5}$$

$$\mathbf{F_{yy}}(k,l) = \begin{cases} -\sum_{j\in H(k)} \frac{2(y_k-y_j)^2}{\delta_{kj}^4} - \frac{1}{\delta_{kj}^2} - \frac{e_r^2+1}{e_r^2}(-\frac{1}{\delta_{kj}^2} + \\ 4\frac{(y_k-y_j)^2}{\delta_{kj}^4}) - \frac{1}{e_r^2}(\frac{1}{\delta_{kj}^2} - 3\frac{(y_k-y_j)^2}{\delta_{kj}^4}), \quad k=l, \\ \\ -I_{H(k)}(l)(\frac{1}{\delta_{kl}^2} - 2\frac{(y_k-y_l)^2}{\delta_{kl}^4} + \frac{1+e_r^2}{e_r^2}(4\frac{(y_k-y_l)^2}{\delta_{kl}^4} - \\ \frac{1}{\delta_{kl}^2}) - \frac{1}{e_r^2}(3\frac{(y_k-y_l)^2}{\delta_{kl}^2} - \frac{1}{\delta_{kl}^2})), \quad k\neq l, \end{cases} \tag{6.6}$$

$$\mathbf{F_{xy}}(k,l) = \begin{cases} -\sum_{j\in H(k)} 2(x_k-x_j)(y_k-y_j)\frac{1}{\delta_{kj}^4} - \\ 4\frac{1+e_r^2}{e_r^2}\frac{(x_k-x_j)(y_k-y_j)}{\delta_{kj}^4} + \frac{3}{e_r^2}\frac{(x_k-x_j)(y_k-y_j)}{\delta_{kj}^4}, \quad k=l, \\ \\ -I_{H(k)}(l)(-2(x_k-x_l)(y_k-y_l)\frac{1}{\delta_{kl}^4} + 4\frac{1+e_r^2}{e_r^2} \\ \frac{(x_k-x_l)(y_k-y_j)}{\delta_{kj}^4} - \frac{3}{e_r^2}\frac{(x_k-x_j)(y_k-y_j)}{\delta_{kj}^4}), \quad k\neq l. \end{cases} \tag{6.7}$$

7. SIMULATION RESULTS

In this section, we compare the aforementioned algorithms in the context of node localization in sensor networks. Network nodes are considered to be uniformly distributed in a square with area equal to 1, i.e., the x and y coordinates of the sensor nodes are uniformly distributed in $[0, 1]$. We employ the alignment procedure described in [4], where necessary, in order to estimate the absolute coordinates, and adopt root mean squared error (RMSE) as our estimation performance metric:

$$RMSE := \frac{\sum_{i=1}^{N} \sqrt{(x_{ri} - x_{ei})^2 + (y_{ri} - y_{ei})^2}}{N}, \tag{7.1}$$

where x_{ei}, y_{ei} are the estimated coordinates, and x_{ri}, y_{ri} are the actual coordinates of sensor i. The computational complexity orders of the various algorithms under consideration are listed in Tables 7.1 and 7.2, for the case of full and partial connectivity, respectively.

The baseline[1] MDS algorithm performs PCA of the doubly-centered matrix of squared distances, and is henceforth referred to as PCA-based MDS. We also implemented Costa's iterative majorization algorithm. We tried both random initialization and the alternative initialization strategy suggested in [2]. The

[1] PCA-based MDS is not directly applicable in the case of partial connectivity.

Tab. 7.1: Computational complexity orders for full connectivity (N is number of nodes)

Algorithm	Complexity
Fastmap	$\mathcal{O}(N)$
Fastmap+SD	$\mathcal{O}(pN^2)$, $p << N$
PCA	$\mathcal{O}(N^3)$
Costa's	$\mathcal{O}(kN^2)$, $k << N$

Tab. 7.2: Computational complexity orders for partial connectivity (s is the average number of distance measurements collected by a node)

Algorithm	Complexity
Fastmap	$\mathcal{O}(N)$
Fastmap+SD	$\mathcal{O}(psN)$, $p << N$
Costa's	$\mathcal{O}(ksN)$, $k << N$

former yields unsatisfactory results that do not improve with decreasing error variance; the latter often yields complex coordinates when the triangle inequality fails due to measurement errors. It is clear that Costa's algorithm is sensitive with respect to initialization, and could benefit from a better "warm start". For this reason, we also tried using our adaptation of Fastmap to initialize Costa's iteration.

Fig. 7.1 shows the RMSE performance of the various algorithms (PCA,

Fastmap, Fastmap+SD, Fastmap+Costa, and Costa with random initialization) for a network of 80 sensors, as a function of e_r^2. Distance measurements were drawn from the multiplicative noise model in (6.1). The corresponding Cramér-Rao Bound (CRB) is also plotted as a benchmark. For the SD step of the proposed algorithm (Fastmap+SD), a step-size of $\lambda = 0.01$ and $p = 10$ SD iterations were used. The convergence threshold in Costa's algorithm was set to $\epsilon = 0.1$. From Fig. 7.1, we observe that stand-alone Fastmap exhibits poor performance, which quickly degrades with increasing range error variance. When randomly initialized, Costa's algorithm also performs poorly in this setup, and its performance does not improve with decreasing error variance. Fastmap+SD and Fastmap+Costa are the best options from the viewpoint of RMSE performance, and remain relatively close to the CRB, especially for low range error variance. Interestingly, the proposed algorithm is not only less complex, but also more accurate than PCA. This is partially attributed to the fact that PCA uses double centering, which colors the noise, whereas the proposed algorithm directly aims to minimize the stress function.

Fig. 7.2 shows corresponding results for a network of 200 nodes ($\lambda = 0.005$; the remaining setup is the same as in Fig. 7.1). The estimation accuracy of PCA, Fastmap+SD, and Fastmap+Costa, improves relative to Fig. 7.1, as expected. Fastmap does not benefit, due to the lack of (implicit or explicit) averaging, while Costa's algorithm with random initialization performs slightly worse than in Fig. 7.1.

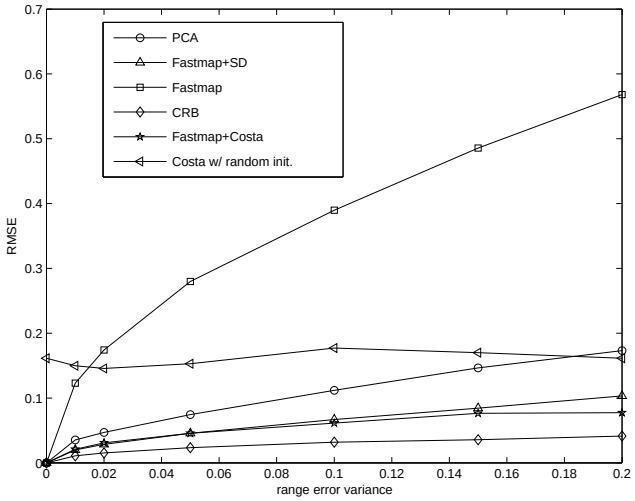

Fig. 7.1: RMSE performance vs. measurement range error variance; N=80, fully connected network, multiplicative normal measurement noise, 100 Monte Carlo runs.

We also tried an additive measurement noise model, i.e., measurements drawn from

$$d_{ij} \sim \delta_{ij} + \mathcal{N}(0, e_r^2), \tag{7.2}$$

in which case the variance of the measurement error is independent of the distance between the two nodes. The results are shown in Fig. 7.3 for the case of 80 nodes, and in Fig. 7.4 for the case of 200 nodes. We observe again that Fastmap+SD and Fastmap+Costa yield approximately the same RMSE performance, significantly outperforming stand-alone Fastmap and PCA.

One might also wonder whether the RMSE comparison of the various algorithms is sensitive with respect to the statistics of the multiplicative noise (normal versus log-normal, see also the appendix). Fig. 7.5 presents simulation results for

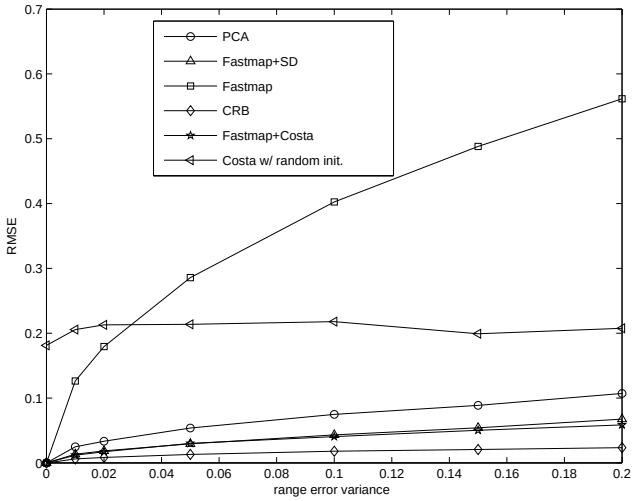

Fig. 7.2: RMSE performance vs. measurement range error variance; N=200, fully connected network, multiplicative normal measurement noise, 100 Monte Carlo runs.

the multiplicative log-normal noise model employed in [2]. We observe that the relative performance ordering of the different algorithms is the same as in Fig. 7.1.

Fig. 7.6 shows the average computational cost in floating point operations (FLOPS) of Fastmap+SD and Fastmap+Costa, as a function of the number of nodes, N. We observe that Fastmap+SD exhibits significantly lower complexity (up to five times lower) than Fastmap+Costa. The values of the step-size λ used for the different values of N are listed in Table 7.3.

In all simulation results presented so far, the network was assumed to be fully connected, i.e., distance measurements were available for each pair of nodes in the network. We now switch to partially connected scenarios. We assume that nodes

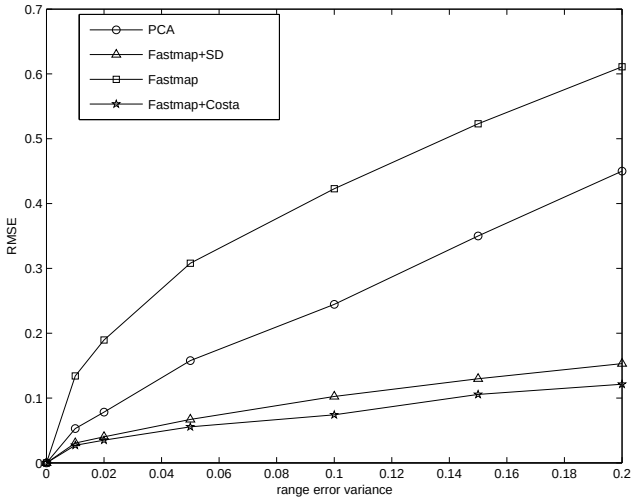

Fig. 7.3: RMSE performance vs. measurement range error variance; N=80, fully connected network, additive normal measurement noise, 100 Monte Carlo runs.

which are further apart than a certain threshold (radio range) cannot hear each other, the corresponding distance measurement is marked as unavailable, and the associated weight in the stress function is set to zero. An exception is that every node is assumed to be within range from each of the three anchor/pivot nodes. We adopt the multiplicative normal noise model in (6.1), and consider two cases: in the first the measurement range is 0.14 and in the second it is 0.3. Fig. 7.7 and Fig. 7.8 show the RMSE performance of Fastmap+SD, Fastmap+Costa, and the CRB (which accounts for the missing data) for the two cases, as a function of range error variance, for $N = 80$ nodes.

Table 7.4 lists the values of λ used in the SD iteration for the three different connectivity scenarios (fully connected, partially connected with measurement range equal to 0.3, or 0.14) and $N = 80$. For Fastmap+SD, we tried two dif-

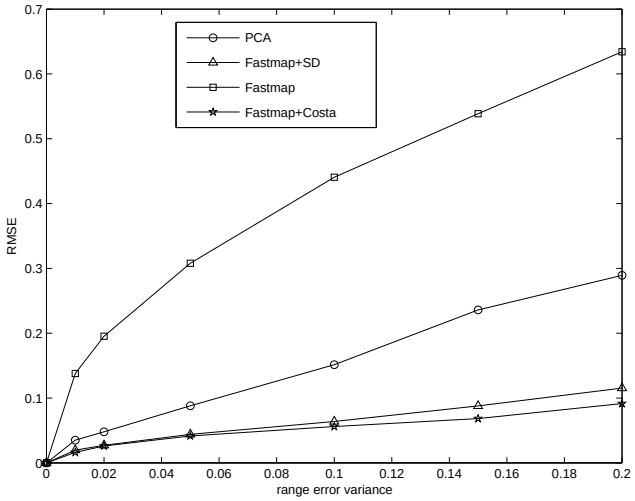

Fig. 7.4: RMSE performance vs. measurement range error variance; N=200, fully connected network, additive normal measurement noise, 100 Monte Carlo runs.

ferent values for the number of SD iterations: $p = 10$ and $p = 30$. From Fig. 7.7 and Fig. 7.8, we observe that the RMSE performance of Fastmap+Costa is better than that of Fastmap+SD with $p = 10$ iterations, and comparable to Fastmap+SD with $p = 30$ iterations. The corresponding FLOP counts in Fig. 7.9 and Fig. 7.10 show that Fastmap+SD with $p = 10$ maintains its computational complexity advantage compared to Fastmap+Costa, although the gap is somewhat smaller than in the case of full-connectivity. Increasing p improves the RMSE performance of Fastmap+SD, but at the cost of computational complexity. For very sparse networks, it appears that Fastmap+Costa is preferable to Fastmap+SD if accuracy is the prime consideration.

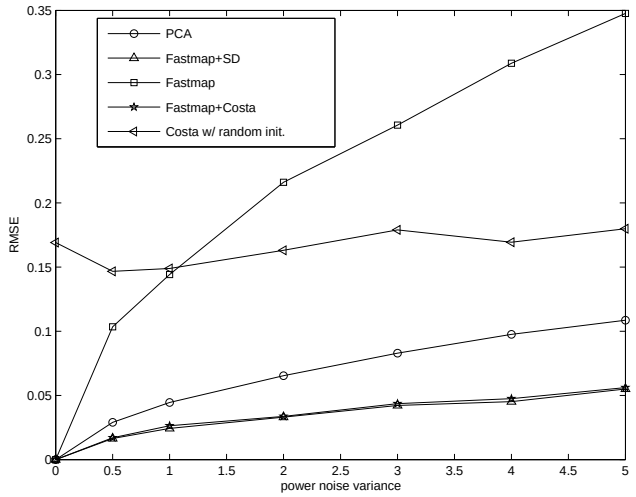

Fig. 7.5: RMSE performance vs. power noise variance (see appendix); N=80, fully connected network, multiplicative log-normal measurement noise, 100 Monte Carlo runs.

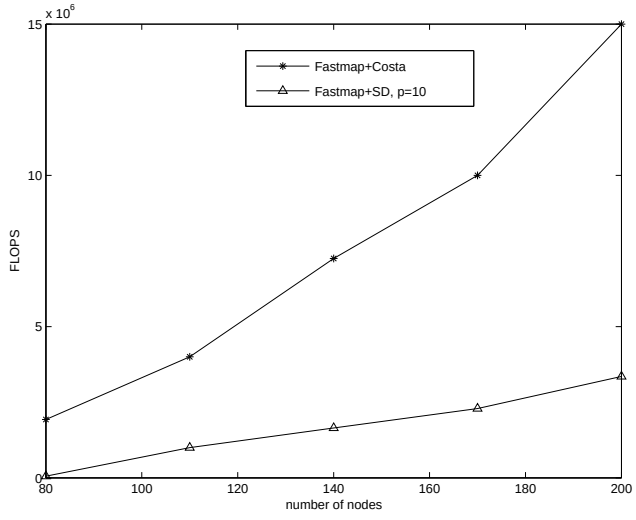

Fig. 7.6: Average computational cost in FLOPS vs. number of nodes; fully connected network, range error variance=0.1, 50 Monte Carlo runs. For Costa's algorithm, threshold=0.1.

Tab. 7.3: Choice of step-size as a function of the number of nodes

N	λ
80	0.01
110	0.0075
140	0.007
170	0.006
200	0.005

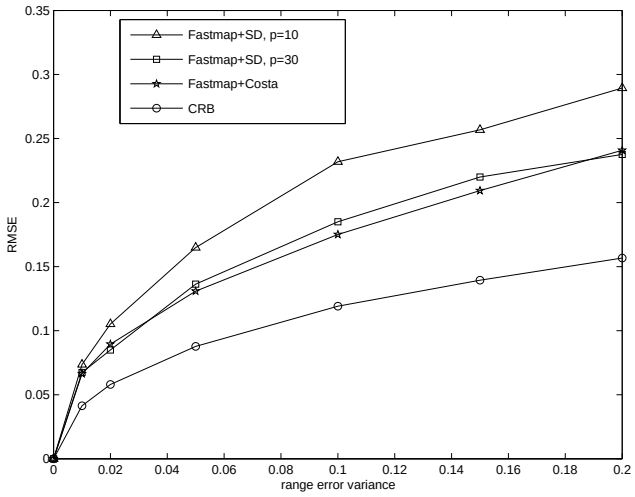

Fig. 7.7: RMSE performance and CRB for limited measurement range = 0.14 (weights corresponding to actual distances greater than this limit are set to zero); N=80, 100 Monte Carlo runs. For Costa's algorithm, threshold=0.1.

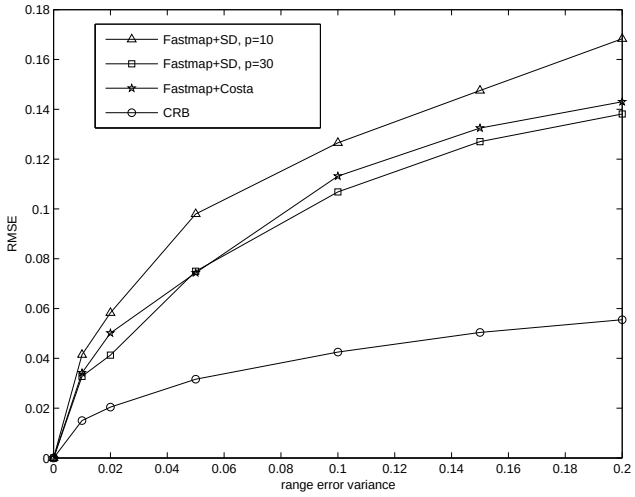

Fig. 7.8: RMSE performance and CRB for limited measurement range =0.3; N=80, 100 Monte Carlo runs. For Costa's algorithm, threshold=0.1.

Tab. 7.4: Choice of step-size as a function of measurement range. 80 sensor nodes

Measurement Range	λ
Infinite	0.01
0.3	0.013
0.14	0.015

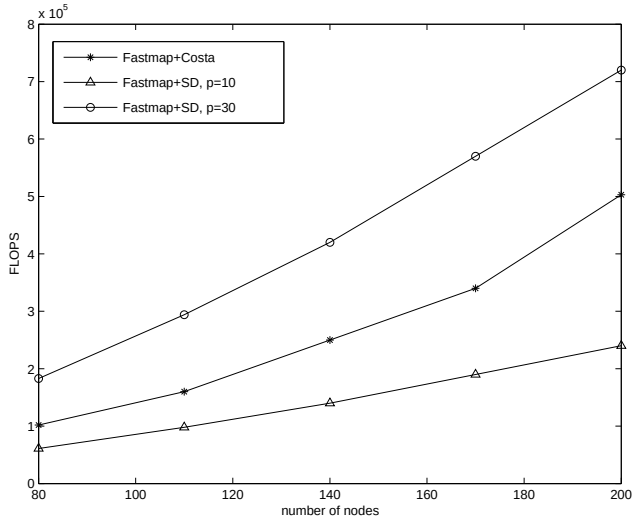

Fig. 7.9: Average computational cost in FLOPS vs. number of nodes for limited measurement range =0.14 (measurements corresponding to actual distances greater than 0.14 are missing); multiplicative normal measurement noise, range error variance=0.1, 50 Monte Carlo runs.

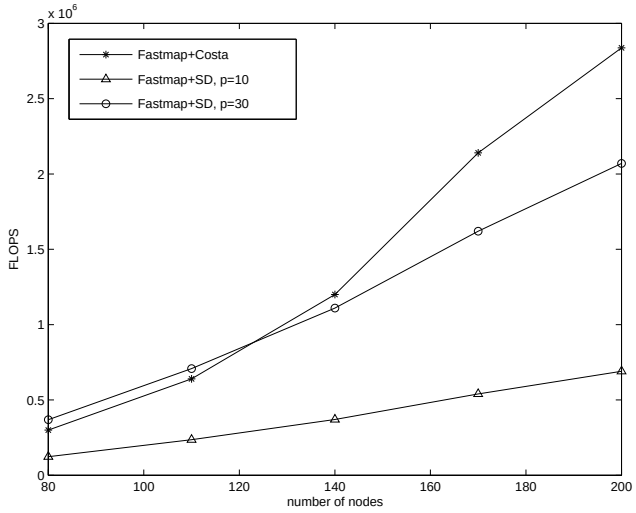

Fig. 7.10: Average computational cost in FLOPS vs. number of nodes for limited measurement range =0.3; multiplicative normal measurement noise, range error variance=0.1 , 50 Monte Carlo runs.

8. CONCLUSIONS

We have proposed a hybrid two-stage (Fastmap+SD) node localization algorithm that offers a better performance - complexity trade-off than existing alternatives of comparable complexity order. The new algorithm employs Fastmap, coupled with judicious selection of anchor nodes that double as pivots, to generate a computationally cheap yet sufficiently accurate initialization for gradient descent. We also proposed using our adaptation of Fastmap as initialization for Costa's algorithm. Extensive simulations indicate that, in the context of our present application: i) Fastmap+SD uniformly outperforms the classical PCA-based MDS, both in terms of complexity and in terms of estimation accuracy; ii) our adaptation of Fastmap is an effective initialization for Costa's algorithm, leading to a substantial reduction in RMSE; and iii) Fastmap+SD and Fastmap+Costa yield comparable RMSE performance, but Fastmap+SD affords considerable reduction in computational complexity, especially for dense networks. For very sparse networks, it appears that Fastmap+Costa is preferable to Fastmap+SD if accuracy is the prime consideration. We have also derived the pertinent CRB for the multiplicative noise model in [1, 7], which was adopted for most of our simulations. Fastmap+SD and Fastmap+Costa operate close to the CRB for dense networks,

but there is a measurable performance gap for sparse networks.

APPENDIX

Normal vs. log-normal multiplicative noise modelling: In [2, 6], the power received at node i from node j, measured in decibel (dB), is modelled as $P_{ij} = \bar{P}_{ij} + v$, where $\bar{P}_{ij}$ is the mean power, and v is a zero-mean Gaussian random variable of standard deviation σ. The mean power is modelled as $\bar{P}_{ij} = P_0 - 10n_p log_{10}\frac{\delta_{ij}}{\delta_0}$, where P_0 is the mean power for a reference distance, δ_0, and n_p is the path loss exponent. It follows that

$$P_0 - P_{ij} = P_0 - \bar{P}_{ij} - v = 10n_p log_{10}\frac{\delta_{ij}}{\delta_0} - v, \tag{8.1}$$

and the associated distance estimate is given by [2]

$$d_{i,j} = \delta_0 10^{(P_0 - P_{ij})/10n_p}. \tag{8.2}$$

Substituting $P_{ij} = \bar{P}_{ij} + v$ and $\bar{P}_{ij} = P_0 - 10n_p log_{10}\frac{\delta_{ij}}{\delta_0}$ yields

$$d_{ij} = \delta_{i,j} 10^{-v/10n_p}. \tag{8.3}$$

Notice that the noise factor is *log-normal*, whereas in the model of [1, 7] (also adopted herein) the noise factor is normally distributed.

BIBLIOGRAPHY

[1] P. Biswas, T.-C. Liang, T.-C. Wang and Y. Ye, "Semidefinite Programming Based Algorithms for Sensor Network Localization," to appear in *ACM Trans. on Sensor Networks*, 2006. See also http://www.stanford.edu/∼yyye/

[2] J. A. Costa, N. Patwari, A. O. Hero, "Distributed Multidimensional Scaling with Adaptive Weighting for Node Localization in Sensor Networks," *ACM Trans. on Sensor Networks*, submitted.

[3] C. Faloutsos, K.-I. Lin, "FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets," in *Proc. ACM SIGMOD*, vol. 24, no. 2, pp. 163-174, 1995.

[4] X. Ji, H. Zha, "Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling," in *Proc. Infocom*, pp. 2652-2661, 2004.

[5] J. B. Kruskal, M. Wish, *Multidimensional Scaling*, Sage Publications, Beverly Hills, CA, 1978.

[6] N. Patwari, A. Hero, M. Perkins, N. Correal, R. O'Dea, "Relative Location

Estimation in Wireless Sensor Networks," *IEEE Trans. on Signal Processing*, vol. 51, no. 8, pp. 2137-2148, Aug. 2003.

[7] Y. Shang, W. Ruml, Y. Zhang, M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Trans. on Parallel and Distr. Systems*, vol. 15, no. 11, pp. 961-974, Nov. 2004.

[8] W.S. Torgerson, "Multidimensional Scaling: I. Theory and method," *Psychometrika*, vol. 17, pp. 401-419, 1952.

[9] W.S. Torgerson, "Multidimensional Scaling of Similarity," *Psychometrika*, vol. 30, pp. 379-393, 1965.