
Learning Sensor Location from Signal Strength and Connectivity*

Neal Patwari¹, Alfred O. Hero III¹, and Jose A. Costa²

¹ Dept. of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, USA
npatwari@umich.edu and hero@umich.edu

² Center for the Mathematics of Information
California Institute of Technology, Pasadena, USA
jcosta@caltech.edu

Received signal strength (RSS) or connectivity, *i.e.*, whether or not two devices can communicate, are two relatively inexpensive (in terms of device and energy costs) measurements at the receiver that indicate the distance from the transmitter. Such measurements can either be quickly dismissed as too unreliable for localization, or idealized by ignoring the non-circular nature of a transmitter's coverage area. This chapter finds a middle ground between these two extremes by using measurement-based statistical models to represent the inaccuracies of RSS and connectivity.

While a particular RSS or connectivity measurement may be hard to predict, a statistical model for RSS and connectivity can in fact be well-characterized. Many numerical examples are used to provide the reader with intuition about the variability of real-world RSS and connectivity measurements.

This chapter then gives a description of three sensor localization algorithms which are based on 'manifold learning', a class of non-linear dimension reduction methods. These algorithms include Isomap [1], the distributed weighted multi-dimensional scaling (dwMDS) algorithm [2], and the Laplacian Eigenmap adaptive neighbor (LEAN) algorithm [3]. The performance of these estimators is compared via simulation using the RSS and connectivity measurement models.

The results show that while RSS and connectivity measurements are highly variable, these manifold learning-based algorithms demonstrate their robustness by achieving location estimates with low bias and often variance close to the lower bound. Due to their desirability as low cost, low complexity mea-

* This work was partially funded by the DARPA Defense Sciences Office under Office of Naval Research contract #N00014-04-C-0437. Distribution Statement A. Approved for public release; distribution is unlimited.

surements, system designers should consider RSS and connectivity for sensor network applications.

1 Introduction

As indicated in the scope of this edited volume, localization is a key technology in wireless sensor networks. However, there is a tradeoff between accurate sensor location and simple, low cost and energy efficient devices. In many applications, device cost will be a more severe constraint than accuracy.

Application Example: Logistics. As an example, consider deploying a sensor network in a warehouse. Pallets, boxes and parts to be warehoused are tagged with wireless sensors when first brought into the facility. The sensors will allow both monitoring of storage conditions (such as humidity and temperature) and determination of the object location at all times. Compared to radio-frequency identification (RFID) tags, which are only located when they pass within a few feet of a reader, networked wireless sensors can be queried and located as long as they are within range of the closest other wireless sensor in the network.

The objective in warehouse logistics is allowing a human to more quickly locate a particular object in the warehouse; thus being able to locate the device to within a few meters will likely be acceptable. The high density of objects to be tracked will require that each sensor must have extremely low costs, on the order of cents, in order to make it cost effective. Connectivity and RSS become very desirable methods.

Application Example: Traffic Monitoring. RSS and connectivity would be valuable in vehicle traffic monitoring. If you're stuck on the highway in a traffic jam, and can't get a traffic report on the radio, it would be very valuable to know how far the backup extends, in order to plan an alternate route if necessary. If a small percentage of cars had a short-range wireless sensor (with accelerometer and a compass), a message could hop forward, find the point at which the traffic eases, and then measure backwards, via RSS or connectivity, the distance to your car. Its possible that messages could propagate backwards to alert the drivers of cars before they hit the jam. Low device cost and long battery life would be critical to achieve a high enough density of sensors to make this application useful.

1.1 Relevant Research

There is now a large literature of algorithms which use RSS or connectivity (a.k.a. proximity) for sensor localization. These algorithms are either distributed or centralized. A number of distributed localization algorithms are compared in [4]. RSS-based localization has been implemented on motes and

Besides the papers within this volume, the references within [5, 6] also provide links to the extensive literature.

This chapter focuses on the subset of algorithms which propose manifold learning-based algorithms for sensor localization, which are introduced in Section 4. In general, manifold learning approaches can lead to distributable algorithms that extract the information perceived to be most accurate. The communication can often be limited to just the nearest neighbors, and there are manifold learning algorithms guaranteed to find the global optimal solution.

1.2 Outline of Chapter

Prior to discussing the manifold learning algorithms themselves, this chapter explores in detail statistical measurement models and their derivation from empirical radio channel studies. Section 2.1 presents RSS measurements and Section 2.2 presents connectivity measurements. Then, to provide more intuitive understanding of the probabilities involved, Section 3 presents two numerical examples and graphically presents results which can be derived directly from the statistical models of Section 2. Finally, Section 4 presents a general overview and comparison of three manifold learning-based algorithms. Section 5 makes those comparisons quantitative by using the statistical models from Section 2 in simulations. Section 6 discusses the results and concludes.

2 Measurement Models

The key to developing reliable sensor localization systems which use pair-wise measurements is to accurately represent the statistics of the measurements. The system designer must design sensor networks that will be deployed in many places and many environments, none of which are known to the designer. Over the ensemble of these deployments, the environment-dependent errors in the measurements are unpredictable and must be modeled as random. In this section, we discuss what past measurements have indicated about these statistical models.

2.1 Received Signal Strength

It is assumed that a system designer will do two things to attempt to reduce the variability in RSS measurements. In particular, a pair of sensors will:

1. Make multiple RSS measurements over time, and
2. Use a wideband measurement of RSS.

The first assumption is that in applications in which sensors are mostly stationary, it would be acceptable to trade off some time delay for increased

accuracy. When there are moving objects such as people or vehicles in the environment, the time averaging possible with the multiple measurements allows us to reduce their effects.

The latter assumption helps reduce the RSS variability to frequency selective fading. In multipath channels, multipath signals add constructively or destructively at the receiver as a function of the frequency. A wideband measurement, such as using frequency hopping spread-spectrum (FH-SS) or orthogonal-frequency division multiplexing (OFDM), can reduce the variability by averaging RSS over a large frequency band.

The RSS measurement that has been averaged is largely subject to shadowing effects. Shadowing is the signal attenuation caused by stationary objects in the radio channel, such as walls, furniture, and buildings, and can't be averaged out by time or frequency averaging.

Statistical Model

Typically, the ensemble mean received power in a real-world, obstructed channel decays proportional to d^{-n_p} , where n_p is the 'path-loss exponent', typically between 2 and 4 [7, 8]. The ensemble mean power at distance d is typically modeled as

$$\bar{P}(d) = P_0 - 10n_p \log_{10} \frac{d}{\Delta_0} \quad (1)$$

where P_0 is the received power (dBm) at a short reference distance Δ_0 .

The difference between a measured received power and its ensemble average, due to the randomness of shadowing, is modeled as log-normal (*i.e.*, Gaussian if expressed in dB). The log-normal model is based on many years of radio channel measurement results [8, 7, 9] and analytical evidence [10]. This model has further been tested via experimental measurements in wireless sensor networks in both indoor and outdoor environments, at both 900 MHz and at 2.4 GHz [11, 12]. These measurements have verified the ensemble mean power model from (1) and that the variation around the ensemble mean is log-normal. Results from one of these measurement campaigns, reported in [12], are shown in Fig. 1. This campaign measured the RSS between each pair of sensors in a 44-node wireless sensor network.

As a result, the received power (dBm) at sensor i transmitted by j , $P_{i,j}$, is distributed as

$$f(P_{i,j} = p | \{\mathbf{z}_i\}_{i=1}^N) = \mathcal{N}(p; \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|), \sigma_{dB}^2), \quad (2)$$

where $\mathcal{N}(x; \mu, \sigma)$ is our notation for the value at x of a Gaussian p.d.f. with mean μ and variance σ , and the actual transmitter-receiver separation distance $\|\mathbf{z}_i - \mathbf{z}_j\|$ is given by

$$\|\mathbf{z}_i - \mathbf{z}_j\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (3)$$

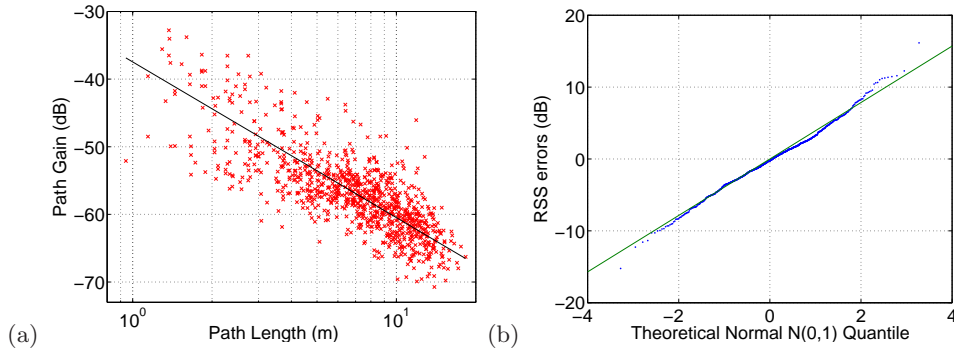


Fig. 1. RSS measurements in a wireless sensor network in [12] show that (a) mean RSS $\bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)$ decays linearly with log distance as in (1) with $n_p = 2.3$, and $\sigma_{dB} = 3.92$; and (b) the quantile-quantile (QQ) plot of RSS errors, $P_{i,j} - \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)$ (dB), vs. the Gaussian distribution.

for a two-dimensional location coordinate $\mathbf{z}_i = [x_i, y_i]^T$. Note that the standard deviation of received power (when received power is expressed in dBm), σ_{dB} , has units of (dB) and is relatively constant with distance. Typically, σ_{dB} is as low as 4 and as high as 12 [7].

Estimating Range from RSS

The ‘range’, *i.e.*, the estimated distance between devices i and j , can be estimated from $P_{i,j}$. First, the maximum likelihood estimate of range is presented. The log-likelihood of $P_{i,j}$ given $d_{i,j} = \|\mathbf{z}_i - \mathbf{z}_j\|$ is,

$$\log f(P_{i,j} | \{\mathbf{z}_i\}_{i=1}^N) = c_1 - \frac{[P_{i,j} - \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)]^2}{2\sigma_{dB}^2} \quad (4)$$

where log indicates the natural logarithm, and c_1 is a constant independent of $\{\mathbf{z}_i\}_{i=1}^N$. Because of the quadratic form, it is clear that the maximum of the log-likelihood occurs when $P_{i,j} = \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)$, where the \bar{P} is given in (1). As a direct result, the distance $\delta_{i,j}^{MLE}$ which best estimates $\|\mathbf{z}_i - \mathbf{z}_j\|$ in the maximum-likelihood sense is,

$$\delta_{i,j}^{MLE} = \Delta_0 10^{\frac{\Pi_0 - P_{i,j}}{10n_p}} \quad (5)$$

Consider what happens if we write $P_{i,j} = \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|) + \eta_{i,j}$, where $\eta_{i,j}$ is ‘noise’ in the measurement which is zero-mean, Gaussian, with variance σ_{dB}^2 . In this case,

$$\begin{aligned} \delta_{i,j}^{MLE} &= \Delta_0 10^{\frac{\Pi_0 - \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|) - \eta_{i,j}}{10n_p}} \\ \delta_{i,j}^{MLE} &= \|\mathbf{z}_i - \mathbf{z}_j\| 10^{-\frac{\eta_{i,j}}{10n_p}}. \end{aligned} \quad (6)$$

The expected value of the MLE distance estimate is,

$$E[\delta_{i,j}^{MLE}] = C\|\mathbf{z}_i - \mathbf{z}_j\|, \quad (7)$$

where

$$C = \exp\frac{1}{2\gamma}, \quad \text{where } \gamma = \left(\frac{10n_p}{\sigma_{dB} \log 10}\right)^2. \quad (8)$$

The parameter C is a multiplicative bias factor, a function of the ratio, σ_{dB}/n_p . For $\sigma_{dB}/n_p = 1.70$, as measured in [12], $C = 1.08$, and for $\sigma_{dB}/n_p = 2.45$, as measured in [11], $C = 1.18$. So, depending on the channel parameters, this bias can be a significant factor.

Motivated by (7), a bias-corrected estimator (a pseudo-MLE) of distance can be defined just by dividing the MLE by C ,

$$\delta_{i,j}^{BC} = \frac{\Delta_0}{C} 10^{\frac{P_0 - P_{i,j}}{10n_p}}. \quad (9)$$

The most important result of the log-normal model is that RSS-based range estimates (from either estimator above) have standard deviation proportional to their actual range. Consider the variance of the MLE distance estimator, which can be calculated to be,

$$\text{var}\{\delta_{i,j}^{MLE}\} = (C^4 - C)\|\mathbf{z}_i - \mathbf{z}_j\|^2$$

This is why RSS errors are referred to as multiplicative. In comparison, errors in distance estimates based on time-of-arrival (TOA) are additive. Clearly, RSS is most valuable between nearby sensors.

2.2 Connectivity Measurements

It is common for localization research to consider connectivity (a.k.a. proximity) measurements as a simple, inexpensive, low-bandwidth, and backward-compatible location measurement. Whether or not devices have accurate RSS measurement circuitry on their receivers, two devices can determine whether or not they can communicate. Two sensors are *not* considered to be connected solely based on the distance between them – two sensors are connected if the receiving sensor can successfully demodulate packets transmitted by the other sensor. The receiver fails to successfully demodulate packets when the received signal strength (RSS) is too low. Since RSS is a random variable due to the unpredictability of the fading channel, and connectivity is a function of RSS, connectivity is also a random variable.

Binary Quantization Model

Specifically, the connectivity measurement of sensors i and j , $Q_{i,j}$, is modeled as a binary quantization of RSS,

$$Q_{i,j} = \begin{cases} 1, & P_{i,j} \geq P_1 \\ 0, & P_{i,j} < P_1 \end{cases} \quad (10)$$

where $P_{i,j}$ is the received power (dBm) at sensor i transmitted by sensor j , and P_1 is the receiver threshold (dBm) under which packets cannot be demodulated. Note that we can both talk about a receiver power threshold P_1 and also the threshold distance d_1 at which the mean received power is P_1 . From (1), this threshold distance is

$$d_1 = \Delta_0 10^{\frac{P_0 - P_1}{10n_p}}. \quad (11)$$

Noise-free Connectivity Model

To generate simulation results for connectivity-based localization, the ‘noise-free’ connectivity model is sometimes used. In this model, radio coverage is assumed to be a perfect circle around the transmitter. Thus, pairs of devices will have exact knowledge of whether or not they are separated by more or less than the coverage radius. Since this is not complete knowledge of distance itself, such a model would still result in localization uncertainty or ‘error’. However, these errors are likely to be just a small contribution to the localization errors in a real system. While such a model might be appropriate for the formulation or visualization of localization algorithms, it is not a means for accurate simulation of estimator variance.

More Detailed Connectivity Models

In reality, being in-range of another device (transmitting a packet which the other device correctly demodulates) is not a step function of received power. Two additional sources of variation in connectivity measurements are:

1. The randomness of frame errors given the received power level, and
2. The possibility of multiple-access interference during a particular transmission.

To discuss issue (1), consider that if we are given received power $P_{i,j}$, connectivity $Q_{i,j} \in \{0, 1\}$ is a binary random variable, such that

$$\mathcal{P}[Q_{i,j} = 1 | P_{i,j}] = \mathcal{P}[\text{No Packet Error} | P_{i,j}] \quad (12)$$

where the probability of a packet error is a function of the type of signalling and forward error correction (FEC) used, packet length, and whether the receiver is coherent or non-coherent. If all of these details of the transceiver implementation were known, a more accurate model could be defined using (12). However, note that the uncertainty in the RSS given distance is much more significant than the uncertainty in error-free packet reception given RSS. In typical digital receivers, there is a large range of received powers for which the probability of packet error is very close to zero, and a large range of power

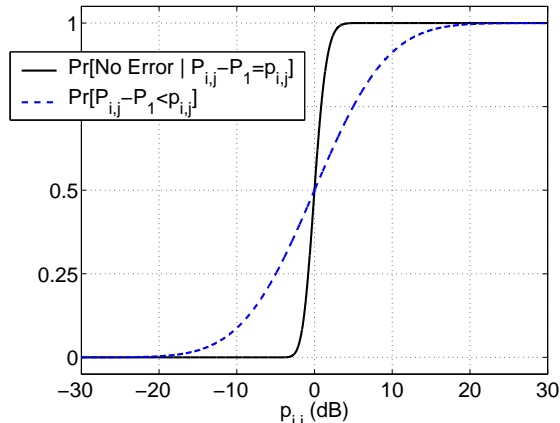


Fig. 2. Two plots relating to the variation in proximity measurements when sensors i and j are separated by the threshold distance d_1 from (11): (---) the CDF of $P_{i,j}$ in dB above the receiver threshold P_1 from (10), and (—) the probability of no packet error given $P_{i,j}$ in dB above P_1 (for a packet of 200 bits and a coherent BPSK receiver without FEC).

for which the probability is very close to one. The range of power for which the probability of packet error is neither close to one or zero is small in comparison. Fig. 2 plots $\mathcal{P}[\text{No Packet Error}|P_{i,j}]$ from (12) under the following conditions: a packet of 200 bits, a coherent BPSK receiver, with no FEC. For comparison, Fig. 2 also plots the CDF of received power for sensors separated by d_1 under a log-normal model with $\sigma_{dB} = 7.38$ dB (as measured in [11]).

Regarding issue (2), outside interference or multiple-user interference (MUI) from other sensors will cause packets to be lost at random times. The interference power raises the noise floor and increases the required power threshold for the desired signal. However, if sensors send multiple packets over time, especially for networks of mostly stationary sensors, it will be likely that a packet received with RSS greater than P_1 will be received without MUI during some transmission, and thus the sensors will measure that they are connected.

Statistical Model

Given the binary quantization model for connectivity in (10) and the log-normal model for $P_{i,j}$ in (2), it can be shown that the probability mass function of $Q_{i,j}$ given the coordinates of devices i and j is

$$\mathcal{P}[Q_{i,j} = s | \mathbf{z}_i, \mathbf{z}_j] = \begin{cases} \Phi \left[\sqrt{\gamma} \log \frac{\|\mathbf{z}_i - \mathbf{z}_j\|}{d_1} \right], & s = 0 \\ 1 - \Phi \left[\sqrt{\gamma} \log \frac{\|\mathbf{z}_i - \mathbf{z}_j\|}{d_1} \right], & s = 1 \end{cases} \quad (13)$$

where $\Phi[\cdot]$ is the cumulative distribution function (CDF) of the univariate zero-mean unit-variance Gaussian distribution, and d_1 is the threshold distance given in (11), and γ is given in (8).

3 Numerical Examples

For the purposes of obtaining an intuitive understanding of the variability of RSS and connectivity measurements, it is valuable to show some numerical values for particular cases. The following examples show that RSS and connectivity are in fact highly variable measurements. While these numbers might scare away system designers, this is certainly not the desired effect. In fact, these models are used to generate RSS and connectivity in Section 5 and demonstrate the accuracy of location estimation algorithms. The ability to achieve localization, even given the variability of the measurements, indicates the robustness of the ‘cooperative’ localization concept.

3.1 RSS-based Distance Estimates

Consider the log-normal RSS measurement $P_{i,j}$ between devices i and j . Consider two different channels, with channel parameters $\sigma_{dB}/n_p = 1.70$ and $\sigma_{dB}/n_p = 2.48$, which correspond to results from measurement campaigns reported in [12] and [11], respectively. (The lower channel parameter was a result of using a very wideband measurement of RSS, while the higher measured the RSS of a continuous wave (CW) signal.)

As a numerical example, let’s calculate the probability that the bias-corrected estimate of distance, $\delta_{i,j}^{BC}$, is within an interval around the correct distance. Specifically, consider the interval,

$$0.5\|\mathbf{z}_i - \mathbf{z}_j\| < \delta_{i,j}^{BC} < 2.0\|\mathbf{z}_i - \mathbf{z}_j\| \quad (14)$$

From (6), we can rewrite the interval from (14) as,

$$\begin{aligned} 0.5\|\mathbf{z}_i - \mathbf{z}_j\| &< \frac{\|\mathbf{z}_i - \mathbf{z}_j\|}{C} 10^{-\frac{\eta_{i,j}}{10n_p}} < 2.0\|\mathbf{z}_i - \mathbf{z}_j\| \\ \frac{\log(0.5C)}{\log 10} &< -\frac{\eta_{i,j}}{10n_p} < \frac{\log(2.0C)}{\log 10} \\ -10n_p \frac{\log(2.0C)}{\log 10} &< \eta_{i,j} < -10n_p \frac{\log(0.5C)}{\log 10} \end{aligned}$$

Since $\eta_{i,j}$ is zero-mean Gaussian with standard deviation σ_{dB} , the probability that $\eta_{i,j}$ falls in this interval is just the area under a normal curve, excluding the tails. Specifically,

$$\mathcal{P} \left[\frac{\|\mathbf{z}_i - \mathbf{z}_j\|}{2} < \delta_{i,j}^{BC} < 2\|\mathbf{z}_i - \mathbf{z}_j\| \right] = \Phi \left[\sqrt{\gamma} \log \frac{2}{C} \right] - \Phi \left[\sqrt{\gamma} \log \frac{1}{2C} \right] \quad (15)$$

where γ is given in (8).

Numerical Solutions

For the two measured channel parameters, $\sigma_{dB}/n_p = 1.70$ and $\sigma_{dB}/n_p = 2.48$, the numerical solutions to (15) are 0.92 and 0.76, respectively.

If instead of using half and twice the actual distance as the interval min and max in (14) we had used $2/3$ and $3/2$ of the actual distance, specifically,

$$0.667\|\mathbf{z}_i - \mathbf{z}_j\| < \delta_{i,j}^{BC} < 1.5\|\mathbf{z}_i - \mathbf{z}_j\|,$$

then these probabilities would be reduced to 0.69 and 0.51 for the $\sigma_{dB}/n_p = 1.70$ and $\sigma_{dB}/n_p = 2.48$ channels, respectively.

What is the difference between the results using the bias-corrected estimator and the MLE estimator of distance? Deriving from (5), we would arrive at the same formula as (15), but with $C = 1$. The probabilities that the MLE distance estimate would be in either the wider or narrower interval are slightly higher than for the bias-corrected estimate (by about 0.01), using either of the two channel parameter ratios. Thus bias correction may help with aligning the estimator mean to the true mean, but it doesn't help reduce the confidence intervals.

Clearly, if RSS is to be used as an estimate of distance, system robustness must be considered. Even in the better channel, the probability of getting a decent estimate of distance (within $2/3$ and $3/2$ of the actual distance) is 69%. A sensor localization system in a wireless network must be designed to make many RSS measurements between many pairs of devices, such that the worst of the errors can be discarded or down-weighted.

3.2 Connectivity and Coverage Area

Equation 13 gives the probability of connectivity for a pair of devices separated by distance $\|\mathbf{z}_i - \mathbf{z}_j\|$, given the channel parameter σ_{dB}/n_p and the distance threshold d_1 . To make this more concrete, two examples are provided in Fig. 3.

As indicated by the figure, regardless of the channel parameter, the probability of having received power higher than the threshold when separated by the threshold distance is 50%. But when the ratio σ_{dB}/n_p is higher, it is more likely that a distant receiver *will* be connected. As a comparison, in Fig. 3(a), the 10% probability contour line is about $1.65d_1$ away from the transmitter, while in 3(b), the 10% probability contour line is about $2.08d_1$ away. Similarly, when the ratio σ_{dB}/n_p is higher, it is more likely that a nearby receiver will *not* be connected. To show this, in Fig. 3(a), the 90% probability contour line is about $0.60d_1$ away from the transmitter, while in 3(b), this line is about $0.48d_1$ away. In sum, as the ratio σ_{dB}/n_p increases, we are less certain of *either* nearby sensors measuring that they are connected or distant sensors measuring that they are not connected.

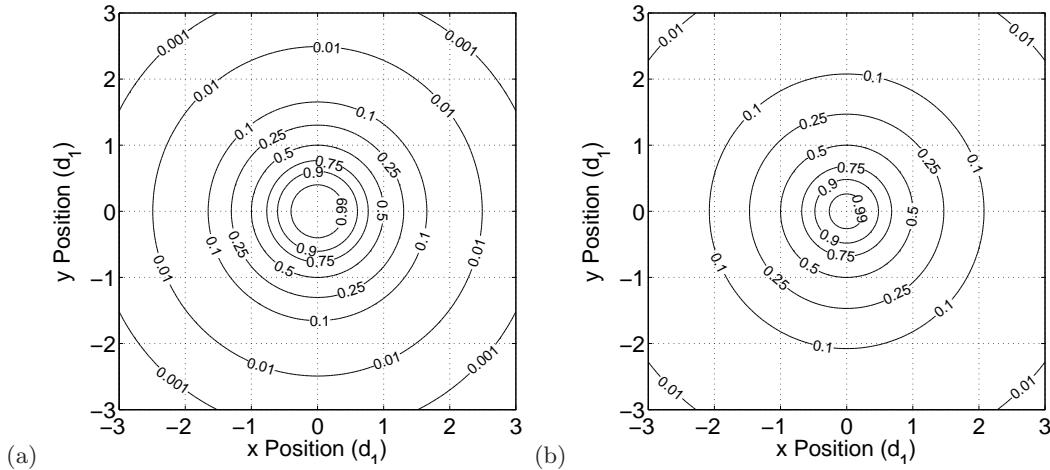


Fig. 3. For a transmitter at the center $(0, 0)$, the probability that the received power will be above the receiver threshold, as a function of the location of the receiver, given a channel parameter σ_{dB}/n_p of (a) 1.70 [12] and (b) 2.48 [11]. Units are in terms of the threshold distance d_1 .

4 Manifold Learning Localization Algorithms

In cooperative sensor localization, we have a network of N sensors, and we want to estimate the coordinates $\{\mathbf{z}_i\}_{i=1}^n$ of sensors $1 \dots n$, which have unknown location. A few sensors, $n + 1 \dots N$ are assumed to have perfect *a priori* knowledge of their coordinates, $\{\mathbf{z}_i\}_{i=n+1}^N$. We are given a mesh of many pair-wise distance or connectivity estimates, from RSS or connectivity, as discussed above. While not all pairs will make measurements, we assume that many neighboring sensors will.

Algorithms have been developed by applying manifold learning techniques to the sensor location estimation problem [1, 2, 3, 13, 14, 15, 16, 17]. Manifold learning is a class of non-linear dimension reduction methods. These are an extension of linear dimension reduction methods such as multi-dimensional scaling (MDS) or principle components analysis (PCA). In PCA, low dimensional coordinates are found by projecting the high dimensional data to the linear subspace which best fits the data. When the high dimensional data don't lie in a linear subspace, the results are inaccurate. For example, for the 3-D data in Fig. 5, PCA would attempt to find a 2-D plane that, when the data were projected to the plane, would best fit the data. Since the 3-D data actually lie on a curved surface, no single plane would serve to fit all of the data. In comparison, in manifold learning, the subspace is only assumed to be locally linear. When reducing the dimension of the data, only the relationships between neighboring high dimensional data points are preserved.

In sensor localization, the manifold learning framework is applicable for two reasons:

1. Measurements between nearby sensors are often more precise and less biased than those between further apart sensors, and
2. Using only measurements between nearby sensors reduces bandwidth, energy, and computational requirements.

Determination of the Neighbor Graph

The first step in manifold learning algorithms is to determine the neighbor graph. In the manifold learning literature, a pair are considered to be neighbors if the Euclidean distance between their high-dimensional data points is less than a threshold. For pair-wise RSS measurements, we might threshold the measured $\delta_{i,j}$ with a pre-determined radius R , or an adaptive threshold set to ensure at least K neighbors. These two neighbor selection methods are called the ‘ R -radius’ and the ‘ K -nearest-neighbors’ (KNN) methods, respectively. Note that the KNN method essentially sets a dynamic radius R for each device depending on its local sensor density, so in either case, we can refer to the threshold distance. For connectivity, the receiver’s power threshold decides on neighbors – if $Q_{i,j} = 1$, then we believe that $\delta_{i,j} < d_1$ for the receiver threshold distance d_1 given in (11), and thus we consider i and j to be neighbors. For connectivity, our pre-determined radius R is equal to d_1 , and for RSS, we may select $R \leq d_1$.

In wireless sensor networks, distance or connectivity between sensors is measured in a noisy channel, so there is an additional complication – it isn’t known which sensors are *actually* within R of each other. Neighbor selection in noise is discussed in Section 4.5 and in more detail in [2, 3].

In sum, the selection of neighbors determines a graph in which neighboring sensors’ nodes are connected, and non-neighboring sensors’ nodes are not. This neighborhood graph is the key input into the next step.

Low-dimensional Coordinate Estimation

The second step in manifold learning algorithms is to find the low-dimension coordinates which best represent the neighbor relationships. This search can be represented as the minimization a cost function or as a constrained optimization problem. Generally, these optimization approaches are of two types: *distance-based* and *similarity-based* approaches. These are contrasted by analogy in Fig. 4. As the name would indicate, the distance-based methods encode information regarding the distances between points in the graph. The similarity-based methods encode inverse distance, or some decreasing function of distance. These cost functions are described in detail in the next sections.

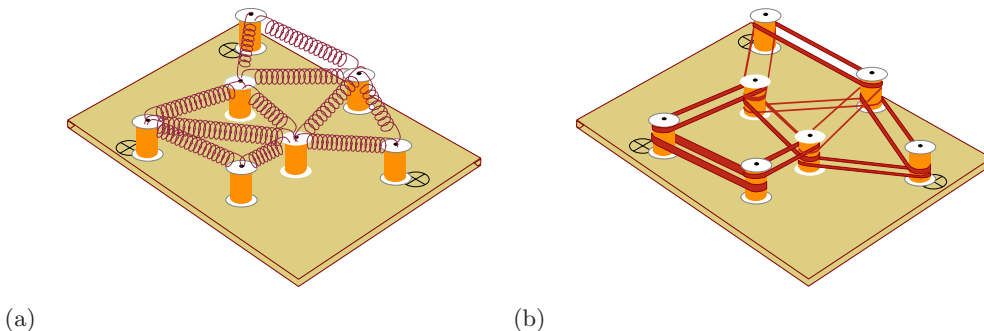


Fig. 4. Physical analogy of manifold learning algorithms based on (a) distance or (b) similarity between sensors. Sensors (spools) are connected by (a) springs which have natural length equal to the measured distance and can push *and* pull their neighbors, or by (b) rubber bands with different weights (thickness of the band) which can *only* pull sensors together. Scaling and rotation are constrained to match the *a priori* known coordinates \otimes .

	Isomap / MDS-MAP	Laplacian Eigenmap / LEAN	dwMDS
Preserves	Distance	Similarity	Distance
Algorithm	Eigen-decomposition	Eigen-decomposition	Iterative, distributed majorization
Minimize	$\sum_{i,j} (\ \mathbf{z}_i - \mathbf{z}_j\ ^2 - \delta_{i,j}^2)^2$	$\sum_{i,j} w_{i,j} \ \mathbf{z}_i - \mathbf{z}_j\ ^2$	$\sum_{i,j} w_{i,j} (\ \mathbf{z}_i - \mathbf{z}_j\ - \delta_{i,j})^2 + r_i \ \mathbf{z}_i - \bar{\mathbf{z}}_i\ ^2$
Computational Complexity	$\mathcal{O}(N^3)$	$\mathcal{O}(KN^2)$	$\mathcal{O}(KLN)$
Post-processing	Rotation	Rotation and Scaling	None
Notes	Sensitive to large range errors	Natural for connectivity	Directly incorporates prior info

4.1 Isomap

In Isomap [18], the distances $\delta_{i,j}$ measured between non-neighbors are ignored. Instead, Isomap replaces distances $\delta_{i,j}$ in (16) with $\tilde{\delta}_{k,l}$, which is set equal to the sum length along the shortest path on the neighbor graph between sensors k and l , for all pairs $(k, l) \in \{1, \dots, N\}^2$. The general idea is that the shortest path on the neighborhood distance graph is a good approximation to the shortest distance on the manifold, as demonstrated in Fig. 5. Isomap then minimizes the cost,

$$S_{Isomap} = \sum_{i=1}^N \sum_{j=1}^N \left(\tilde{\delta}_{i,j}^2 - \|\mathbf{z}_i - \mathbf{z}_j\|^2 \right)^2. \quad (16)$$

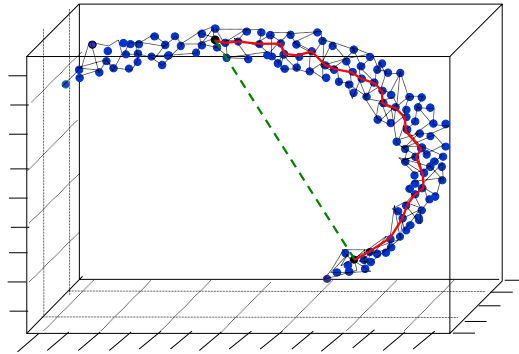


Fig. 5. In Isomap, the distance between two devices is not the direct path in the higher dimensional space (---); rather, it is the shortest path distance between the two points along the edges of the nearest neighbor graph (—).

An algorithm called MDS-MAP, introduced by Shang et. al. [1, 13, 14], applies the Isomap algorithm, when measurements are connectivity, to sensor localization.

Computation: Because the distances are squared before taking the difference, the cost is a quadratic function of the individual coordinates. The minimum of S_{Isomap} can be found directly from the singular value decomposition of the appropriate transform of the $N \times N$ distance matrix $D^2 = [[\delta_{i,j}^2]]_{i,j}$, as derived in detail in [19]. This eigen-decomposition operation has computational complexity $\mathcal{O}(N^3)$. The Isomap algorithm also requires finding of shortest path between each pair of sensors in the network. Using Dijkstra's algorithm, this is an $\mathcal{O}(N^2)$ operation, and this calculation can be performed in a distributed manner in a wireless network.

Post-Processing: Isomap produces the relative, but not absolute map of all N devices in the network. As a post-processing step, all coordinate estimates are transformed (rotate, scale, and translate) by the transformation that makes the coordinate estimates of the known-location sensors, $\{\hat{\mathbf{z}}_i\}_{i=n+1}^N$, best match their *a priori* known coordinates in a least-squares sense.

4.2 Laplacian Eigenmap Adaptive Neighbor (LEAN)

The LEAN algorithm combines Laplacian Eigenmap (LE), a similarity-based manifold learning method, with an an adaptive neighbor weighting algorithm. This section first discusses the LE method given a set of weights $\{w_{i,j}\}$. Then, Section 4.4 discusses the initial selection of the weights, and Section 4.5 discusses the iterative adjustment of the weights in a 2-step adaptive algorithm.

The LE method considers the minimization of the cost S_{LE} :

$$S_{LE} = \sum_{i,j} w_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 \quad (17)$$

subject to the translation and scaling constraints,

$$\sum_i \mathbf{z}_i = \mathbf{0} \quad \text{and} \quad \sum_i \|\mathbf{z}_i\|^2 = 1. \quad (18)$$

where $w_{i,j}$ are weights representing the ‘similarity’ of devices i and j . In the LEAN algorithm, these weights depend on the RSS or connectivity measurements, and are given explicitly in Sections 4.4 and 4.5. The key fact is that $w_{i,j} = 0$ when i and j are non-neighbors (*i.e.*, when $Q_{i,j} = 0$ or $\delta_{i,j} > R$).

The minimum of cost S_{LE} without the constraints in (18) would occur when all the coordinates \mathbf{z}_i were equal. The constraints remove the translation ambiguity by setting the origin as the center, and counteract the tendency to put all points at the origin by mandating a unit norm average coordinate.

Computation: The benefit of the formulation in (17) and (18) is that the globally optimum solution is found via eigen-decomposition. Defining the $N \times N$ weight matrix $W = [[w_{i,j}]]_{i,j}$ and its column sums (or row sums, since W is symmetric) $u_i = \sum_{j=1}^N w_{i,j}$, the graph Laplacian L_W is given by,

$$L_W = \text{diag}[u_1, \dots, u_N] - W, \quad (19)$$

where $\text{diag}[u_1, \dots, u_N]$ is the diagonal matrix with $\{u_i\}$ on its diagonal. Matrix L_W is sparse, since $w_{i,j} = 0$ for non-neighbors, and each row or column has at most $K_{max} + 1$ non-zero elements, where $K_{max} = \max_i K_i$, and K_i is the number of neighbors of sensor i . The eigen-decomposition of L_W is the set of $(\lambda_k, \mathbf{v}_k)$, for eigenvalues λ_k and eigenvectors \mathbf{v}_k , $k = 1 \dots N$. Here, it is assumed w.l.o.g. that the eigenvectors are sorted in increasing order by magnitude of eigenvalue. As presented in detail by Belkin and Niyogi in [20], the \mathbf{v}_k for $i = 2 \dots r+1$ provide the optimal lowest-cost, r -dimensional solution to (17). Specifically,

$$\hat{\mathbf{z}}_i = [\mathbf{v}_2(i), \dots, \mathbf{v}_{r+1}(i)], \quad (20)$$

where $\mathbf{v}_k(i)$ is the i th element of the k th eigenvector.

Finding the smallest eigenvalues and eigenvectors of a sparse and symmetric matrix is a computational problem which has been studied for decades for problems in physics and chemistry [21, 22], and can be solved using distributed algorithms for parallel processing. In particular, if sensors select local cluster-heads, the distributed algorithm can use data-distribution techniques and block-Jacobi preconditioning methods to reduce communication. Due to the sparsity of the graph Laplacian matrix, the computational complexity of the eigen-decomposition is $\mathcal{O}(KN^2)$, where K is the average number of neighbors of each sensor.

Post-Processing: As in the Isomap algorithm, the LE method produces a relative map of the coordinates. The same method as in the previous section is used to determine the best scaling, rotation, and translation of the coordinates based on the *a priori* coordinate information.

Related Research: The locally linear embedding (LLE) [23] and the Hessian-based LLE (HLLE) methods are also similarity-based manifold learning algorithms. The HLLE method [24] expands the optimization to attempt to preserve the local Hessian, *i.e.*, 2nd-order differences within local neighborhoods, within the final low-dimensional coordinate embedding.

4.3 Distributed Weighted Multi-dimensional Scaling (dwMDS)

The dwMDS method minimizes a cost function S_{dwMDS} in a distributed manner, by having each sensor i , for $i = 1 \dots n$, iteratively minimize its own local cost function, S_i [2, 16]. The global cost is additive, *i.e.*,

$$S_{dwMDS} = \sum_{i=1}^n S_i. \quad (21)$$

Using the dwMDS algorithm to optimize S_i at each unknown-location sensor $i = 1 \dots n$ acts to optimize S_{dwMDS} . In the dwMDS method, S_i are given by

$$S_i = r_i \|\mathbf{z}_i - \bar{\mathbf{z}}_i\|^2 + \sum_{j=1}^N \tilde{w}_{i,j} (\delta_{i,j} - \|\mathbf{z}_i - \mathbf{z}_j\|)^2, \quad (22)$$

where $\bar{\mathbf{z}}_i$ represents the mean coordinate of the *a priori* coordinate distribution for sensor i , r_i is the confidence in that mean coordinate, and $\tilde{w}_{i,j}$ is a weight corresponding to the expected accuracy in the $\delta_{i,j}$ measurement. When $r_i = 0$, it indicates no prior information exists for i , and any $0 < r_i < \infty$ indicates imperfect but partial prior knowledge of i 's location. Also, let $\tilde{w}_{i,j} = 2w_{i,j}$ if either $i > n$ or $j > n$, and $\tilde{w}_{i,j} = w_{i,j}$ otherwise. This is done so that each measurement $\delta_{i,j}$ is treated equally, whether or not it was measured between two unknown-location sensors or between and unknown-location and known-location node. Weights $w_{i,j}$ are similar to those used in Section 4.2 and their selection is discussed in Section 4.4.

Computation: In the dwMDS algorithm, sensors serially optimize their own coordinate, given their neighbors most recent coordinate estimate. Sensor i , during its turn, improves its estimate of \mathbf{z}_i . This improvement is done by optimizing a quadratic majorization function for S_i , which guarantees that each iteration of the optimization reduces the global cost S_{dwMDS} . The update function for i is simply a linear function of its neighbors' most recent coordinate estimates. We leave the detailed derivation and presentation of the calculation to the references [2, 16]. Each sensor requires calculation on the order of its number of neighbors K , in each of the L iterations required for convergence, so the total network-wide computational complexity is $\mathcal{O}(LKN)$. The dwMDS also has a slower increase in communication requirements than centralized localization algorithms as N increases [16]. Furthermore, since prior information is included directly in (22), there is no need to do post-processing.

4.4 Weight Selection

In this section, we describe the selection of weights $w_{i,j}$ in the LEAN and dwMDS methods. We consider separately measurements of RSS and connectivity.

RSS Measurements When distance estimates are available from RSS, weights $\{w_{i,j}\}$ in the LE and dwMDS methods are set as follows. The LOESS method, a non-parametric scheme, is used to set $w_{i,j}$:

$$w_{i,j} = \begin{cases} \exp\{-\delta_{i,j}^2/h_{i,j}^2\}, & \text{if } i \text{ and } j \text{ are neighbors} \\ 0, & \text{otherwise} \end{cases}, \quad (23)$$

where $h_{i,j}$ is the maximum distance $\delta_{k,l}$ measured by either sensor i or j .

Connectivity Measurements Since no pure distance estimate is available when only connectivity is measured, it would seem that any $w_{i,j}$ between connected sensors i and j should have identical weight. However, this scheme is not the best approach because it tends to give too much ‘pull’ to sensors with many neighbors. This ‘pull’ serves to bias other sensors’ coordinates towards itself, as described in [6]. Here, we use a simple symmetric weight scheme that sets weights so that the total weights for sensor i , $u_i = \sum_j w_{i,j} \approx 1$:

$$w_{i,j} = (1/K_i + 1/K_j)/2, \quad (24)$$

where K_l is the total number of neighbors of sensor l . While u_i is not identical for all i , it is slightly higher for sensors with more neighbors than their neighbors. This behavior is desirable and helps to improve bias performance.

4.5 Adaptive Neighbor Selection:

Typically, we select neighbors closer than a threshold distance R . For RSS measurements, this distance R can be set less than d_1 , the receiver threshold distance from (11), but in connectivity, note that $R = d_1$ is the only option. The key issue that this section addresses is that when distances estimated from noisy RSS or connectivity measurements are used to select neighbors, the neighbor selection process can be the source of significant bias. The act of selecting the neighbors with $\delta_{i,j}$ less than a threshold has a tendency to select the $\delta_{i,j}$ which are, on average, less than the actual distances $\|\mathbf{z}_i - \mathbf{z}_j\|$. And for connectivity, we know from Section 3.2 that sensors may include as neighbors many sensors which are further than R and may ignore sensors closer than R .

Both the dwMDS algorithm and the LE-based algorithm specifically counter this bias effect by using a two-stage adaptive algorithm:

Stage 1: First, distances $\{\delta_{i,j}\}$ are measured, and those (i,j) with $\delta_{i,j} < R$ (or $Q_{i,j} = 1$ for connectivity) are selected as neighbors. The dwMDS or LEAN algorithm then computes $\{\tilde{\mathbf{z}}_i\}_{i=1}^n$, which are referred to as ‘interim’ coordinate estimates.

Stage 2: Next, the neighborhood is adjusted based on the interim coordinate estimates:

- In the dwMDS algorithm, the neighborhood graph is completely re-calculated based solely on the interim coordinates. Sensors with interim coordinates closer than R , *i.e.*, $\|\tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_j\| < R$, are selected as neighbors. The weight matrix W is re-calculated as given in Section 4.4 using the new neighbor graph.
- In the LE-based localization algorithm, define K_i to be the number of neighbors of sensor i . Also define \tilde{K}_i to be the number of sensors with interim coordinates closer than R to $\tilde{\mathbf{z}}_i$, *i.e.*, $|\{j : \|\tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_j\| < R\}|$. Next, set the new weights $w_{i,j}$ so that the new sensor weight sums \tilde{u}_i are given by $\tilde{u}_i = u_i \sqrt{K_i / \tilde{K}_i}$, as detailed in [3]. Essentially, increase a sensor's pull if its estimate is in a less dense area than it should be.

Using the new neighborhood structure, the dwMDS algorithm or the LE algorithm is re-run to estimate final coordinate estimates $\{\hat{\mathbf{z}}_i\}$. Sections 5.1 and 5.2) show the dramatic effects of the adaptive neighbor selection in sensor localization.

5 Simulation Results

Geometries: The performance of these three algorithms is first demonstrated on a grid network, of 7×7 sensors arranged on a uniform grid of unit area, as shown in Fig. 6, in which the four corner devices are reference nodes and the remaining 45 are unknown location devices, and $L = 1$ m. The grid geometry is chosen first because it shows the geometric bias effects very well. Subsequently, in Section 5.4, random geometries are explored via simulation. Note that the choice of $L = 1$ m is arbitrary, since any scaling of L would proportionally scale the simulation errors and lower bound on standard deviation. Essentially, all distances, estimator biases and standard deviations can be taken in units of L .

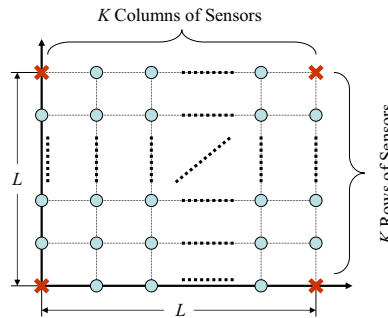


Fig. 6. Grid geometry layout of sensors.

Simulation Input and Output: For all experiments, $R = 0.4$ or 0.3 is chosen as the threshold distance, and the channel parameter ratio σ_{dB}/n_p is set to either 1.70 or 2.48. Independent Monte Carlo trials (200) are run to determine uncertainty ellipses and bias performance (per sensor) of the location estimates. The results are displayed in many figures in this section. Each figure plots the estimator mean (\blacktriangledown) and 1-standard deviation uncertainty ellipse of the estimator (—), compared to the actual device location (\bullet). One-standard deviation uncertainty ellipses, as a coarse generalization, can be seen as a 2-D confidence interval, within which most of the coordinate estimates will lie.

Also plotted in the simulation figures are the Cramér-Rao lower bound (CRB) on the uncertainty ellipses (- - - -), which were derived and presented for RSS and connectivity measurements in [12] and [25], respectively. Any unbiased estimator must have 1- σ uncertainty ellipse larger than that given by the CRB. Note that the CRB may only provide a loose lower bound on the best performance achievable by any unbiased estimator.

For comparing different estimators, let the mean bias \bar{b} and the RMS standard deviation $\bar{\sigma}$ of the estimator be defined as:

$$\bar{b} = \frac{1}{n} \sum_{i=1}^n \|\bar{\mathbf{z}}_i - \mathbf{z}_i\|, \quad \bar{\sigma} = \sqrt{\frac{1}{n} \text{tr} \mathbf{C}} \quad (25)$$

where n is the number of unknown-location sensors, $\bar{\mathbf{z}}_i$ is the mean of all of the estimates of sensor i over all trials of the simulation, \mathbf{z}_i is the actual location of sensor i , and \mathbf{C} is the covariance of the coordinate estimates over all trials.

5.1 RSS Results from dwMDS in Grid

To show the benefit of adaptive neighborhood selection from Section 4.5, we show in Fig. 7 the performance of the dwMDS algorithm with and without its 2nd adaptive stage, and with different values of σ_{dB}/n_p and threshold R . Fig. 7(a) stops the algorithm after Stage 1, using the interim coordinates as the final estimates. The biasing effect of neighborhood selection in noise results in $\bar{b} = 0.12$, by effectively shortening the average distance estimates and thus forcing a smaller sensor location estimate map. In contrast, Fig. 7(b) allows the completion of Stage 2, and is nearly unbiased with $\bar{b} = 0.02$, and has $\bar{\sigma} = 0.09$. Except at the edge nodes, the estimator variance is visibly close to the lower bound. The dwMDS algorithm (and also the lower bound) degrade with increasing ratio σ_{dB}/n_p as shown in Fig. 7(c), which has $\bar{\sigma} = 0.16$. The algorithm is shown to be robust the change to $R = 0.3$ in Fig. 7(d) which results in $\bar{\sigma} = 0.11$, and we note without showing additional plots that the error performance is robust to a wide range of R .

5.2 Connectivity Results from LEAN in Grid

The same tests as performed in Section 5.1 are now performed on the LEAN algorithm using measurements of connectivity. We should expect that the

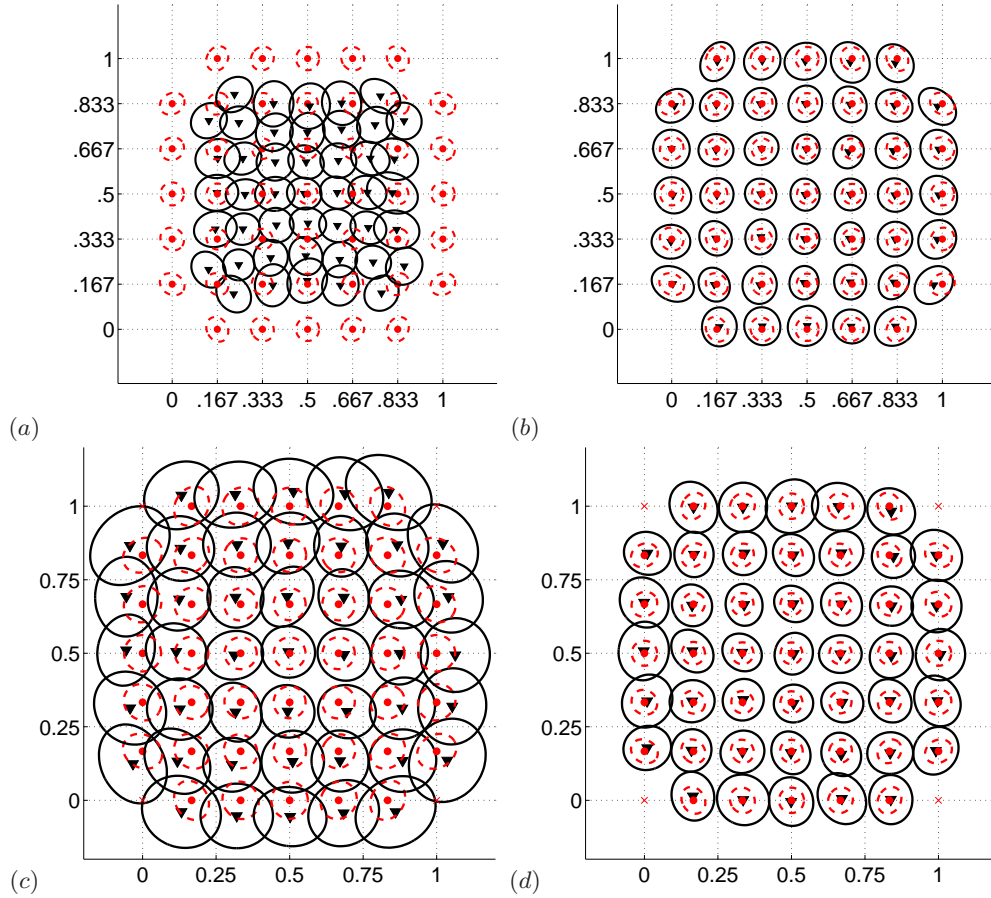


Fig. 7. Comparison of RSS-based dwMDS estimators using (a) single-stage or (b) adaptive, 2-stage neighbor selection, both with $\sigma_{dB}/n_p = 1.70$ and $R = 0.4$. Also shown is the adaptive dwMDS performance (c) when $\sigma_{dB}/n_p = 2.48$ and $R = 0.4$ or (d) when $\sigma_{dB}/n_p = 1.70$ and $R = 0.3$.

variance will increase, and we do see this in Fig. 8. Figs. 8(a) and 8(b) compare the LEAN performance with and without the second stage of the adaptive neighbor selection algorithm. In this comparison, the improvement is only marginal - both the bias and standard deviation decrease only slightly, so that $\bar{b} = 0.03$ and $\bar{\sigma} = 0.14$ in (b). The real benefit of the 2-stage adaptive LEAN algorithm is its ability to keep the bias very low over a wide range of σ_{dB}/n_p and R , a much wider range than possible without the adaptive weighting. Figs. 8(c) and 8(d) show the simulation results when the 2-stage adaptive LEAN algorithm is used with (c) $\sigma_{dB}/n_p = 2.48$ and $R = 0.4$; and (d) $\sigma_{dB}/n_p = 1.70$ and $R = 0.3$. The bias is nearly constant in all figures (a-d), but in (c), $\bar{\sigma} = 0.20$ and in (d), $\bar{\sigma}$ is the same as in (b) at 0.14.

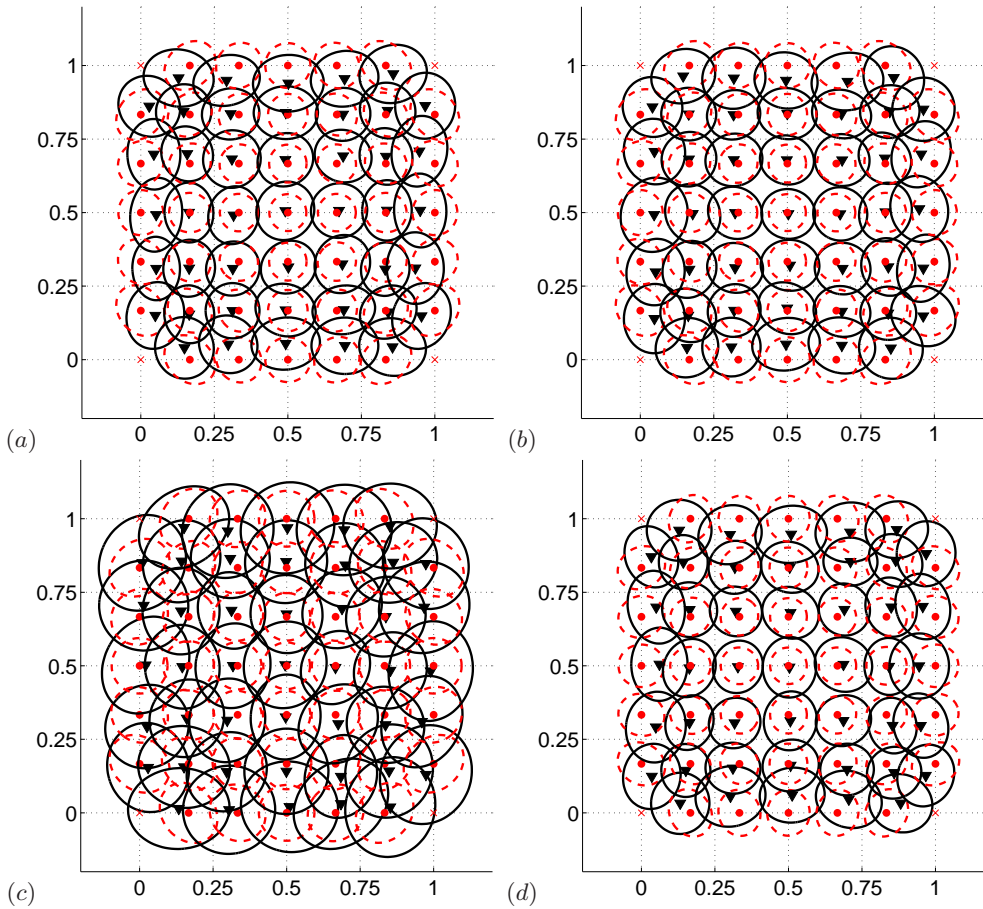


Fig. 8. Comparison of LEAN estimators using (a) single-stage or (b) adaptive, 2-stage neighbor selection, both with $\sigma_{dB}/n_p = 1.70$ and $R = 0.4$. Also shown is the adaptive dwMDS performance (c) when $\sigma_{dB}/n_p = 2.48$ and $R = 0.4$ or (d) when $\sigma_{dB}/n_p = 1.70$ and $R = 0.3$.

5.3 Connectivity and RSS Results from Isomap/MDS-MAP Method in Grid

Isomap can be run either from RSS or connectivity measurements. When RSS measurements are available, $\delta_{i,j}$ is calculated using the MLE in (5). When only connectivity measurements are available, $\delta_{i,j} = 1$ when i and j are connected (as in [1]). Since Isomap computes a relative map and then scales it to match the prior information, this choice of $\delta_{i,j} = 1$ for two connected devices is arbitrary and irrelevant. Simulation results are shown in Fig. 9(a-d), for the cases of measurements of connectivity or RSS, and for the channels with

σ_{dB}/n_p equal to 1.70 and 2.48. These changing parameters and the simulation results are shown in Table 1.

	Simulation Inputs		Outputs	
	Measurements	σ_{dB}/n_p	\bar{b}	$\bar{\sigma}$
Fig. 9(a)	RSS	1.70	0.03	0.15
Fig. 9(b)	Connectivity	1.70	0.02	0.21
Fig. 9(c)	RSS	2.48	0.04	0.23
Fig. 9(d)	Connectivity	2.48	0.05	0.28

Table 1. Isomap simulation inputs and outputs.

In all simulations, $R = 0.4$. The estimator is largely unbiased, but the variance is significantly larger than the lower bound. In general, this can be attributed to the form of the cost in (16), which is a function of the squared difference between squared distances, rather than just the squared difference between distance itself. The squaring of the distance before taking the difference enables solution via eigen-decomposition as discussed in Section 4.1, but the algorithm is more sensitive to the tails of the density of $\delta_{i,j}$.

5.4 Simulation Results in a Random Deployment

Fig. 10 shows the performance of the dwMDS and LEAN algorithms in two different random deployments of sensors. The 49 coordinates for the random deployment were chosen independently from a uniform distribution on the square area. Then, the four sensors closest to the corners are selected as reference devices. All simulations use $\sigma_{dB}/n_p = 1.70$ and $R = 0.4$. In the first randomly-generated geometry, Fig. 10(a) shows dwMDS performance to be $\bar{b} = 0.03$ and $\bar{\sigma} = 0.10$; and Fig. 10(b) shows LEAN performance to be $\bar{b} = 0.04$ and $\bar{\sigma} = 0.12$. In the second randomly-generated geometry, Fig. 10(c) shows dwMDS performance to be $\bar{b} = 0.02$ and $\bar{\sigma} = 0.09$, while Fig. 10(d) shows the LEAN performance to be $\bar{b} = 0.05$ and $\bar{\sigma} = 0.12$.

Compared to the grid geometry, there are a few sensors with very high bias in the LEAN algorithm results. However, the dwMDS algorithm results show very low bias, even for the most isolated sensors. Variance is slightly higher for the dwMDS algorithm, while the LEAN algorithm has lower variance than in the grid geometry. The LEAN algorithm may actually benefit, in terms of variance, from the non-uniform density of sensors across the randomly-deployed network.

6 Discussion and Conclusion

The simulation results show us particular advantages of the dwMDS and LEAN methods compared to the Isomap-based methods, when comparing

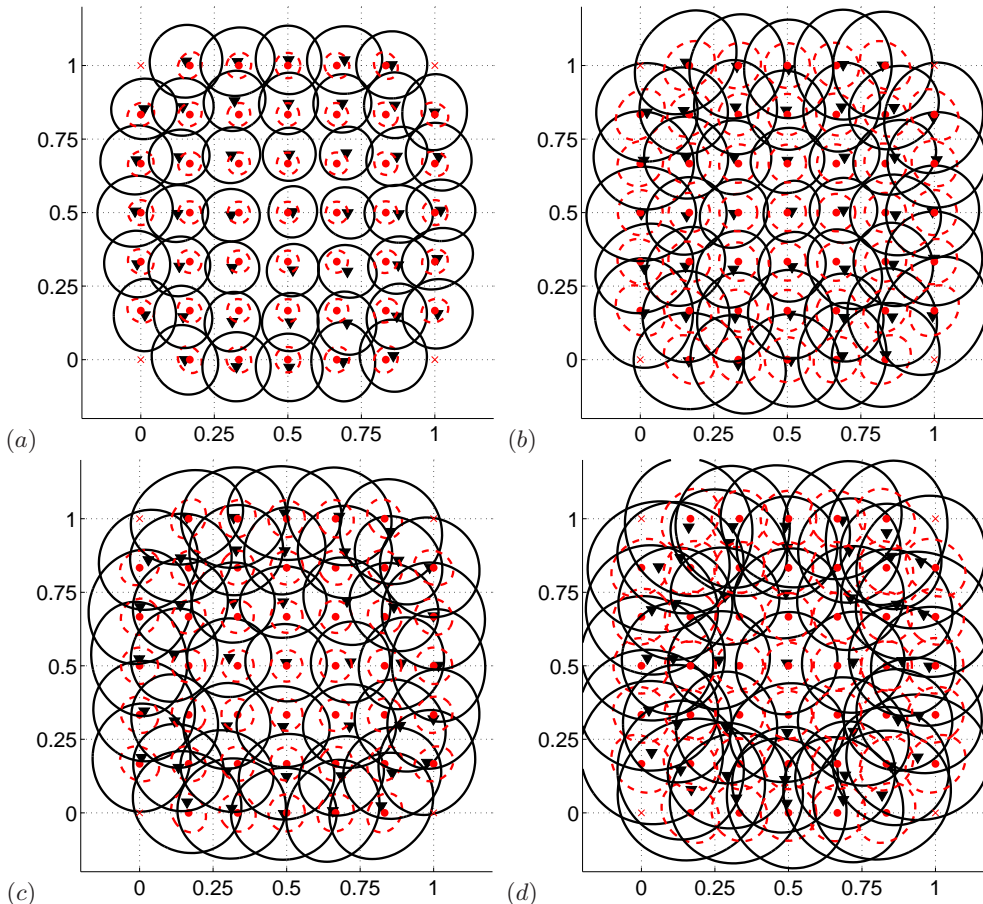


Fig. 9. Comparison of Isomap estimators using (a)&(c) RSS measurements and (b)&(d) connectivity measurements. All use $R = 0.4$, but for (a)&(b) $\sigma_{dB}/n_p = 1.70$ and for (c)&(d) $\sigma_{dB}/n_p = 2.48$.

estimator covariance. Furthermore, the standard deviation of localization using connectivity measurements (via LEAN) is almost 60% higher than with RSS measurements (via dwMDS). This increase in variance is approximately the same as the increase in the lower bounds from RSS to connectivity, as related in [25]. Both estimators, except for sensors at the edge of the networks, perform reasonably close to the lower bound.

One perspective on this analysis is that a high path loss exponent n_p is actually desirable in terms of localization accuracy. If a system operates in an environment in which n_p is high, but σ_{dB} is not proportionally as high, then more location information is possible. When antennas are very close to the ground, or when center frequencies are in oxygen absorption bands, path loss

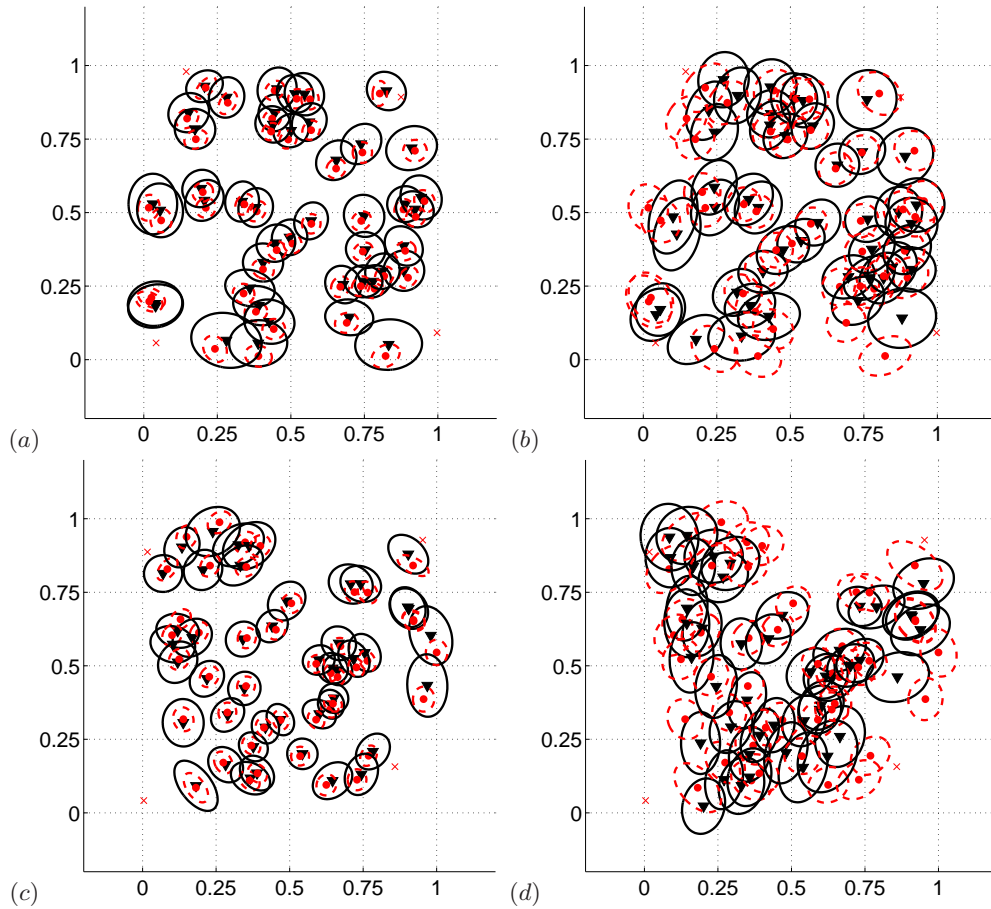


Fig. 10. Simulation of estimators (a)&(c) dwMDS with RSS and (b)&(d) LEAN with connectivity measurements, in two different non-uniform, random deployment of sensors. All use $R = 0.4$ and $\sigma_{dB}/n_p = 1.70$.

exponents are typically higher. Of course, higher path loss exponents mean that transmit powers must be higher in order to achieve the same coverage.

Adaptive methods help dramatically improve both estimator performance (in the dwMDS algorithm) and help increase the variety of environments and network densities in which the estimator performs with low bias and variance close to the lower bound (as seen in the LEAN algorithm). These benefits of adaptive neighborhoods are demonstrated by the simulation results.

Opportunities for future research are many. While LEAN is distributable because of its reliance on finding a few of the smallest eigen-vectors of a sparse symmetric matrix, algorithms from the parallel processing literature must be optimized to perform well in a energy-limited sensor network. Other

manifold learning-based methods are possible, and can be explored for their ability to reduce the computational or communication requirements of sensor localization.

Regardless, though, the statistics of RSS and connectivity measurements are critical to the determination of many measures of ‘how well’ a particular localization system will perform. This chapter has provided a thorough background in these models, and shown how they can be used in simulation of localization algorithms.

References

1. Yi Shang, Wheeler Ruml, Ying Zhang, and Markus P. J. Fromherz, “Localization from mere connectivity,” in *Mobihoc '03*, June 2003, pp. 201–212.
2. Jose A. Costa, Neal Patwari, and Alfred O. Hero III, “Achieving high-accuracy distributed localization in sensor networks,” in *IEEE Intl. Conf. Acoustic, Speech, & Signal Processing (ICASSP'05)*, March 2005, pp. 642–644.
3. Neal Patwari and Alfred O. Hero III, “Adaptive neighborhoods for manifold learning-based sensor localization,” in *IEEE Workshop on Signal Processing Adv. Wireless Commun. (SPAWC'05)*, June 2005, pp. 1098–1102.
4. Koen Langendoen and Niels Reijers, “Distributed localization in wireless sensor networks: A quantitative comparison,” *Elsevier Science*, May 2003.
5. Bhaskar Krishnamachari, *Networking Wireless Sensors*, Cambridge University Press, Cambridge UK, 2005.
6. Neal Patwari, Joshua Ash, Spyros Kyperountas, Robert M. Moses, Alfred O. Hero III, and Neiyer S. Correal, “Locating the nodes: Cooperative localization in wireless sensor networks,” *IEEE Signal Processing Mag.*, vol. 22, no. 4, pp. 54–69, July 2005.
7. Theodore S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall Inc., New Jersey, 1996.
8. Homayoun Hashemi, “The indoor radio propagation channel,” *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943–968, July 1993.
9. Greg Durgin, Theodore S. Rappaport, and Hao Xu, “Measurements and models for radio path loss and penetration loss in and around homes and trees at 5.85 GHz,” *IEEE Journal on Sel. Areas in Comm.*, vol. 46, no. 11, pp. 1484–1496, Nov. 1998.
10. Alan J. Coulson, Allan G. Williamson, and Rodney G. Vaughan, “A statistical basis for lognormal shadowing effects in multipath fading channels,” *IEEE Trans. on Veh. Tech.*, vol. 46, no. 4, pp. 494–502, April 1998.
11. Neal Patwari, Robert J. O’Dea, and Yanwei Wang, “Relative location in wireless networks,” in *IEEE Vehicular Technology Conf. (VTC)*, May 2001, vol. 2, pp. 1149–1153.
12. Neal Patwari, Alfred O. Hero III, Matt Perkins, Neiyer Correal, and Robert J. O’Dea, “Relative location estimation in wireless sensor networks,” *IEEE Trans. Signal Processing*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.
13. Yi Shang, Wheeler Ruml, Ying Zhang, and Markus P. J. Fromherz, “Localization from connectivity in sensor networks,” *IEEE Trans. Parallel & Distr. Syst.*, vol. 15, no. 11, pp. 961–974, Nov. 2004.

14. A.A. Ahmed, Yi Shang, and Hongchi Shi, "Variants of multidimensional scaling for node localization," in *11th Int. Conf. Parallel & Distr. Syst.*, July 2005, vol. 1, pp. 140–146.
15. Xiang Ji and Hongyuan Zha, "Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling," in *IEEE INFOCOM 2004*, March 2004, vol. 4, pp. 2652–2661.
16. Jose A. Costa, Neal Patwari, and Alfred O. Hero III, "Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks," *IEEE/ACM Transactions on Sensor Networks*, May 2006, (to appear).
17. Hyuk Lim and Jennifer C. Hou, "Localization for anisotropic sensor networks," in *IEEE INFOCOM 2005*, March 2005, vol. 1, pp. 138–149.
18. Joshua B. Tenenbaum, Vin de Silva, and John C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, Dec 2000.
19. T. Cox and M. Cox, *Multidimensional Scaling*, Chapman & Hall, London, 1994.
20. Mikhail Belkin and Partha Niyogi, "Laplacian Eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, June 2003.
21. Earnest R. Davidson, "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices," *J. Comput. Phys.*, vol. 17, no. 1, pp. 87–94, Jan. 1975.
22. Luca Bergamaschi, Giorgio Pini, and Flavio Sartoretto, "Computational experience with sequential and parallel, preconditioned Jacobi Davidson for large, sparse symmetric matrices," *J. Computational Physics*, vol. 188, no. 1, pp. 318–331, June 2003.
23. Sam T. Roweis and Lawrence K. Saul, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol. 290, pp. 2323–2326, Dec 2000.
24. David L. Donoho and Carrie Grimes, "Hessian Eigenmaps: new locally linear embedding techniques for high-dimensional data," Tech. Rep. TR2003-08, Dept. of Statistics, Stanford University, March 2003.
25. Neal Patwari and Alfred O. Hero III, "Using proximity and quantized RSS for sensor localization in wireless networks," in *2nd ACM Workshop on Wireless Sensor Networks & Applications (WSNA'03)*, Sept. 2003, pp. 20–29.