A Spectral Clustering Approach to Validating Sensors via Their Peers in Distributed Sensor Networks

H. T. Kung Dario Vlah
{htk, dario}@eecs.harvard.edu
Harvard School of Engineering and Applied Sciences
Cambridge, MA 02138

ABSTRACT

In a distributed sensor network, the goodness of a sensor may change according to its current device status (e.g., health of hardware) and environment (e.g., wireless reception conditions at the sensor location). As a result, it is often necessary to validate periodically sensors in the field, in order to identify those which no longer work properly and eliminate them from applications' use. In this paper, we describe a spectral clustering approach of using peer sensors to identify these bad sensors. Using a simple model problem, we describe how our sensor validation method works and demonstrate its performance in simulation.

I. INTRODUCTION

For a distributed sensor network deployed in the field, we may find that sensor performance degrades over time due to a number of reasons. For example, sensors may malfunction or become damaged, get blown by the wind to a place where radio reception is poor, enter into a function-reducing mode due to low battery power, and may even be maliciously compromised. This means that we will need to check if sensors still function as expected every so often. In many circumstances it would be impractical to bring calibration instruments to the field and conduct sensor validation procedures on the spot. Fortunately, in a distributed sensor network, there are potentially many sensors in the environment, so they could check each other's validity.

In this paper, we present a spectral clustering approach to validating sensors via peers in the field. We show that we can identify bad sensors with a high degree of accuracy. After removing these bad sensors from applications, we have a smaller as well as more accurate sensor set for use by applications. This means more accurate sensing as well as reduced computing and communication. In addition, this could lead to a stealthier sensing environment and improved protection against malicious tampering. This approach of validating sensors in the field, followed by the elimination of bad sensors, departs from some other approaches in distributed sensor networks where no attempts are made to remove bad sensors, and emphasis is instead placed on being tolerant against erroneous measurements from bad sensors (e.g., those in localization with MDS [1] and SISR [2]).

II. OVERVIEW OF THE APPROACH

First, we define some basic concepts and terms. We will use multiple *reference targets*, against which measurements of sensors will be compared. Throughout the paper, the term *"target"* refers to reference targets. We assume that properly working sensors of similar properties, such as radio and antenna characteristics, node environments and power usage, will report similar measurements against the same targets. We call these working sensors *"good sensors,"* and other sensors which do not report proper measurements *"bad sensors."* Formally, we call sensors of similar properties *"nearby sensors,"* and the associated properties *"sensor indices."* Finally, we call the measurements of a sensor on a target the *"weight"* of this sensor-target pair.

To illustrate this terminology, we consider a simple example, where sensors are indexed by their antenna orientations. In this case, nearby sensors are those which have similar antenna polarizations. Suppose that the received signal strength (RSS) values reported by a sensor on a target correlate with the matching degree of their antenna polarizations. Then nearby sensors, which are in good working condition, are expected to report similar RSS measurements on the same targets. As another example, we could consider sensors to be indexed by their locations rather than antenna angles, so in this case nearby sensors are those in the same general area.

Next, we describe a graph model for clustering sensors which will reveal bad sensors. Let $\mathbf{B} = (S, T, W)$ be the weighted bipartite sensor-target graph, with the sensor nodes S on one side, target nodes T on the other side, and the sensor-target edge weights W defined above. We can represent the graph $\mathbf{B}$ by its matrix $\mathbf{A}$, that is, whose entry a_{ij} is the weight between sensor s_j and target t_i.

We expect that nearby sensors will have similar weights at the same targets. Suppose that we have a group of good sensors which have large weights with respect to some targets. Then we expect that this group of sensors will form a cluster in the weighted graph $\mathbf{B_S} = (S, W_S)$, whose edge weights W_S are given by the adjacency matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. On

the other hand, by definition, a bad sensor cannot have weights similar to those of nearby sensors for the same targets and therefore, would not be in the same cluster of nearby sensors.

From the above, it follows that, to identify bad sensors, we need to solve the problem of determining whether a sensor belongs to a cluster of nearby sensors. We assume that there are plentiful sensors so that clusters of nearby sensors will be large. A sensor is deemed to be bad if the sensor satisfies one of the two conditions:

(C1) the sensor is in a small unique cluster, or
(C2) the sensor is in a small out-of-place component of a large cluster.

Therefore the essence of the clustering problem studied in this paper is to identify these two conditions in the $\mathbf{B_S}$ graph. In the rest of this paper, we show that we can achieve this objective by examining the leading eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$.

III. A Model Problem

To help describe our approach, we use a model problem where sensors are indexed by their antenna orientations, as described earlier, and where we wish to identify the bad sensors. We deliberately make the problem simple to permit easy visualization and interpretation of clustering results, while still capturing essential issues of interest. When extending this model to real-world applications, various details may need to be added, such as path loss models.

In this model problem, sensors and reference targets are randomly placed in a region. Each uses an antenna of a certain orientation, as depicted in Figure 1.

We will solve the sensor clustering problem based on the following two premises:

1) Sensor-target bipartite arrangement, which clusters the best-matched sensors to individual targets. In our model problem, we assume that the matching of a sensor and a target is based on the degree to which their antenna orientations match. We will identify bad sensors by detecting sensors which do not fall in the cluster of nearby sensors.
2) Use of non-principal eigenvectors, in addition to the principal one, to identify clustering structures in the sensor-target bipartite arrangement. There is a large body of recent work on spectral graph partitioning for general graphs, with relation to some early literature. [3], [4] Although non-principal eigenvectors generally represent a subject of great mathematical sophistication, we have found that they are more readily analyzed in the context of sensor-target bipartite graphs. For example, under the bipartite framework, we can explain how non-principal eigenvectors assist the identification of clusters of sensors matched to individual targets. In the next subsection, we will show how the second eigenvector is instrumental in this aspect. Note the i-th eigenvector, for $i > 1$, can be readily computed after the first $i-1$ eigenvectors have been obtained. For example, if we use a Power-Method-like algorithm for computing eigenvalues, we just need to keep the additional eigenvector orthogonalized with respect to the other eigenvectors already computed. [5]

To allow for easy visualization and interpretation, we use a step function to score the degree of matching in polarization angles, as shown in Figure 2. Based on this scoring function, we can assign a matching weight for each sensor-target pair that reflects the antenna polarization difference between the sensor and target. Applying the scoring function of Figure 2 to the antenna orientations of sensors and targets of Figure 1, we derive the sensor-target weight matrix $\mathbf{A}$ in Figure 3.

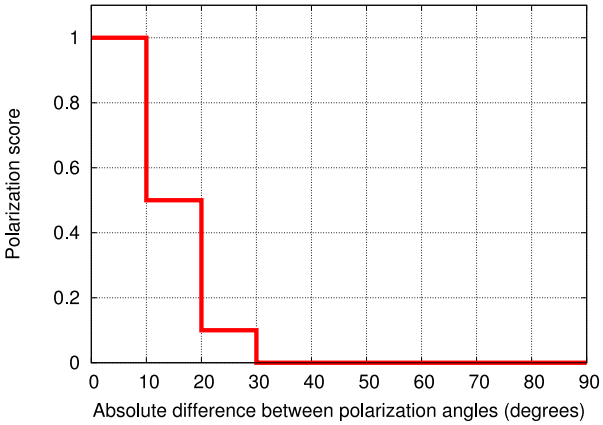
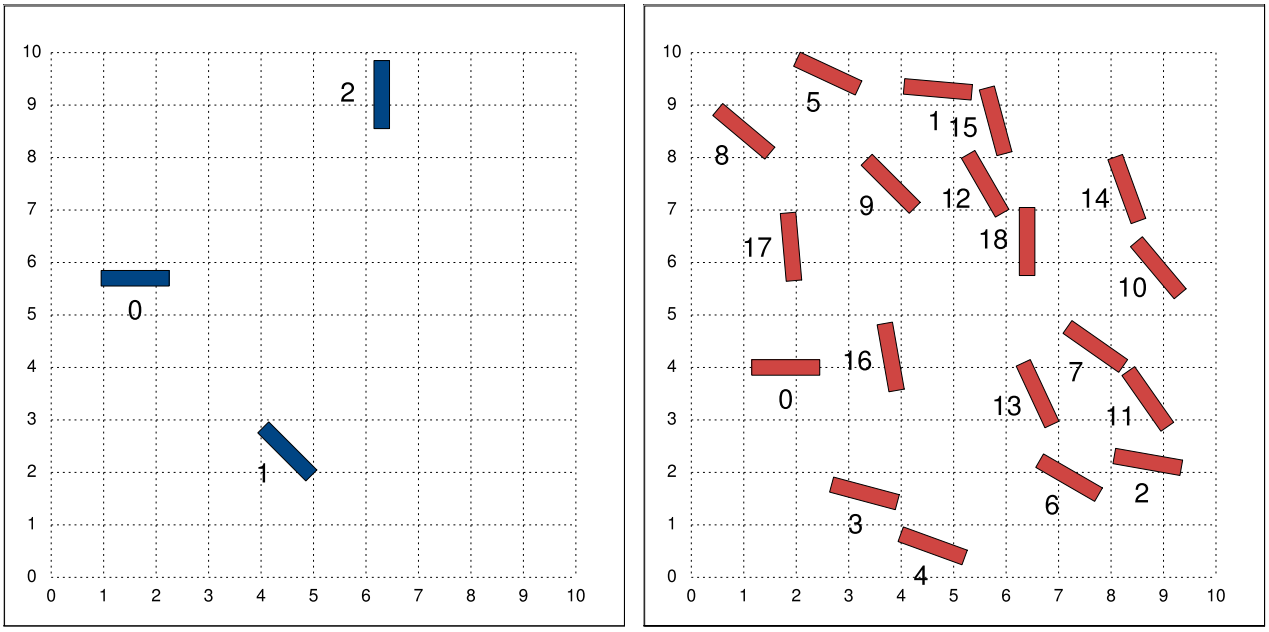


Fig. 2: Step function for scoring polarization matching.

$$\mathbf{A}=\begin{bmatrix} 1 & 1 & 0.5 & 0.5 & 0.1 & 0.1 & & & & & & & & & & \\ & & & & 0.1 & 0.1 & 0.5 & 0.5 & 1 & 1 & 1 & 0.5 & 0.5 & 0.1 & 0.1 & \\ & & & & & & & & & & & & & 0.1 & 0.1 & 0.5 & 0.5 & 1 & 1 \end{bmatrix}$$

Fig. 3: The sensor-target weight matrix corresponding to Figures 1 and 2.

Given a sensor-target weight matrix $\mathbf{A}$, we will examine some number of leading eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ to identify clusters of sensors best matched to individual targets. The number of leading eigenvectors needed is related to the number of leading eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ which are significantly larger than the rest. For example, for the weight matrix $\mathbf{A}$ in Figure 3, the leading three eigenvalues are 4.04, 2.52 and 2.52, and the rest are zero. Thus, we will examine the first and second eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. It turns out that these leading eigenvalues are related to the structure of $\mathbf{A}$, such as its hub structure, which dictates how quickly the leading eigenvalues drop in their magnitude [6].



(a) 3 targets with 3 antenna orientations

(b) 19 sensors with 19 antenna orientations

Fig. 1: Sensors and targets in the same region.

We now consider the principal and second eigenvector of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, as shown in Figures 4 and 5, respectively. At first glance, it may be difficult to see immediately how these two eigenvectors relate to polarization matching between sensors and targets. This will become clear when we look at the bipartite sensor-target graph of Figure 6, where the colored edges represent scores of Figure 2. That is, color-coded scores are blue = 1, green = 0.5 and red = 0.1.

First, let's look at the principal eigenvector of Figure 4. A common way to interpret its components is that they represent the stationary probabilities that a random traversal of the graph of Figure 6 will land at any given sensor node. Indeed, one can show (e.g., by the Perron-Frobenius theorem [7]) that all the components of the principal eigenvector are all positive (or equivalently, all negative) as in Figure 4. The values of these components, which correspond to sensor nodes, reflect connectivity as well as edge weights of the graph. However, there could be "aliases" in the sense that multiple clusters may attain the same component value in the first principal eigenvector. For example, in Figure 4, sensors 4, 5, 13 and 14 all have value .05, and thus the eigenvector implies that {4, 5, 13, 14} is a cluster. However, in Figure 6 we see that sensors 4 and 5 should form a cluster separated from that of sensors 13 and 14, since the former has a connection to target 0 while the latter has a connection to target 2. Similarly, the first principal eigenvector of Figure 4 mistakenly identifies clusters {2, 3, 15, 16} and {0, 1, 17, 18}.

Next, we look at the second eigenvector of Figure 5.

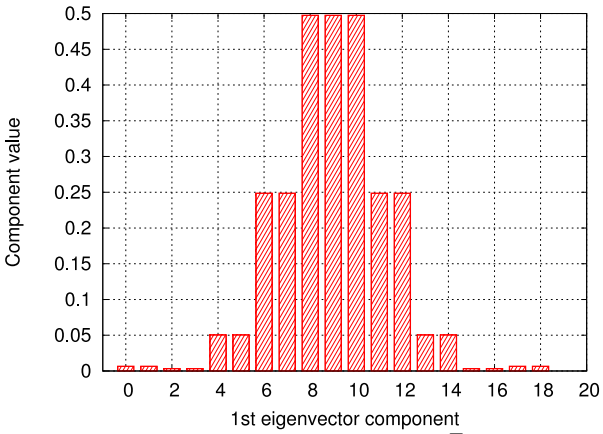


Fig. 4: The principal eigenvector of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, with values of its elements (also called components) shown. Note that components here correspond to sensor nodes.

It has seven distinct values. These imply seven clusters: {0, 1}, {2, 3}, {4, 5}, {6, 7, 8, 9, 10, 11, 12}, {13, 14}, {15, 16}, and {17, 18}. One can check that the second eigenvector has now fixed all the mistaken clusters identified by the first eigenvector, except for mistakenly identifying {6, 7, 8, 9, 10, 11, 12} as a cluster due to aliasing. We see from Figure 6 that although all these seven nodes connect to target 1, sensors 8, 9 and 10 connect to target 1 with blue connections, while sensors 6, 7, 11 and 12 with green connections. Thus {8, 9, 10} and {6, 7, 11, 12} should form two clusters. One can check that this

partition of {6, 7, 8, 9, 10, 11, 12} into these two clusters is actually implied by the principal eigenvector of Figure 4, as the two groups attain two distinct values 0.25 and .5. Thus, for this example, by using both the first and second eigenvectors together, we have obtained the correct eight clusters: {0, 1}, {2, 3}, {4, 5}, {8, 9, 10}, {6, 7, 11, 12}, {13, 14}, {15, 16}, and {17, 18}.

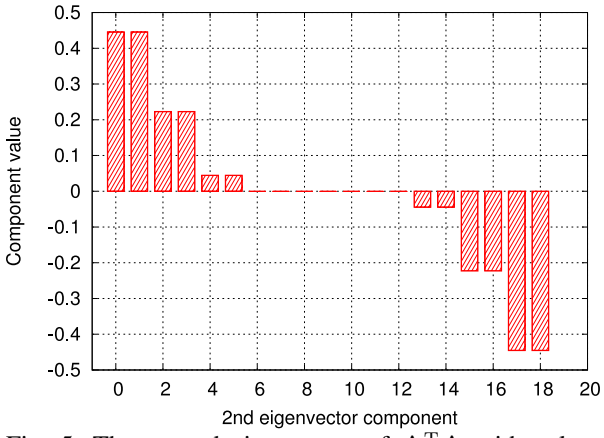


Fig. 5: The second eigenvector of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ with values of its elements (also called components) shown. Note that components here correspond to sensor nodes.

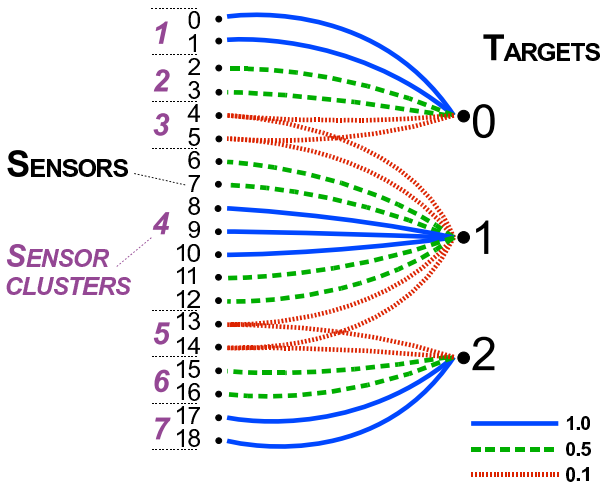


Fig. 6: Bipartite sensor-target score graph, where color-coded scores are blue = 1, green = 0.5 and red = 0.1. The seven sensor clusters shown are those found by the second eigenvector of Figure 5. Note that with the help of the principal eigenvector of Figure 4, cluster 4 will be correctly partitioned into the two clusters {8, 9, 10} and {6, 7, 11, 12}.

We have seen in this example that the principal and second eigenvectors complement each other, in the sense that each can fix the alias problems of the other. This phenomenon is not an accident. Without loss of generality, we consider the case that eigenvectors are computed by the power method. Note that because the matrix $\mathbf{A}$ has non-negative entries, using a positive initial vector for the power method results in the principal eigenvector having aliased values when the underlying graph is a symmetric bipartite graph, as in Figure 6. Being orthogonal to the principal eigenvector, the second one must contain both positive and negative components, and as a result, it does not exhibit the same alias problems as the first one. But due to cancellation resulting from combining positive and negative values, the second eigenvector may suffer from other alias problems, which the first one does not share since all its values are positive. This means that the principal and second eigenvectors can fix each other's aliases. (We note that the use of leading eigenvectors in this manner is a technique also known to multidimensional scaling, a statistical method for computing data similarity. [8])

Using this joint approach of principal and non-principal eigenvectors, we have a general method of automatically finding sensor clusters which match individual targets.

IV. Sign-based Spectral Clustering

From our arguments in the preceding section, we consider here a sign-based spectral clustering. Suppose that we use the leading k eigenvectors. Multiple sensors will be clustered together only if their corresponding components in these k eigenvectors have matching signs. (Note that k eigenvectors can specify no more than 2^{k-1} clusters, given that the principal eigenvector always contains non-negative components.) As an example, this sign-based process of clustering of the sensors in Figure 6 with $k = 3$ is shown in Figure 7. Note that for the same matrix $\mathbf{A}$ in Figure 3, sign-based clustering gives three clusters using three leading eigenvectors, while the value-based clustering described in the preceding section gives seven clusters. It is expected that the sign-based clustering generally gives coarse clustering, since only signs of component values rather than their actual values are used.

In practice, sign-based clustering is robust against those sensor measurements which may have significant variation. In this case, component values of eigenvectors will have enough variance so their precise values can not be used. Sign-based clustering is a method to address this uncertainty in sensor measurements.

V. Sensor Validation: Illustrative Example of Detecting Bad Sensors and Design Considerations

In this section, we first illustrate how our spectral clustering method can identify a bad sensor for the model problem described in Section III. We then describe some considerations in designing a spectral clustering scheme for detecting bad sensors.

Weight matrix:

Targets	Sensors																		
	1	1	0.5	0.5	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0.1	0.1	0.5	0.5	1	1	1	0.5	0.5	0.1	0.1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.1	0.5	0.5	1	1

Eigenvectors:

e_1:	0	0	0	0	0	0	0.2	0.2	0.5	0.5	0.5	0.2	0.2	0	0	0	0	0	0
e_2:	-0.4	-0.4	-0.2	-0.2	-0	-0	-0	-0	-0	-0	-0	-0	-0	0	0	0.2	0.2	0.4	0.4
e_3:	-0.4	-0.4	-0.2	-0.2	-0	-0	0	0	0	0	0	0	0	-0	-0	-0.2	-0.2	-0.4	-0.4

Cluster assignments:

0	0	0	0	0	0	2	2	2	2	2	2	2	1	1	1	1	1	1

Fig. 7: Weight matrix $\mathbf{A}$, 3 leading eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, and the resulting clustering assignment. Sensors with eigenvector component signs $\{+,-,-\}$ are assigned to cluster 0, $\{+,-,+\}$ to cluster 2, and $\{+,+,-\}$ to cluster 1.

Suppose that sensor 9 in Figure 6 malfunctions, and obtains erroneous measurements on target 0 which are similar to those of sensors 0 and 1. The corresponding bipartite graph is shown in Figure 8. The weight matrix $\mathbf{A}$ reflects this error, as shown in Figure 9. That is, with this bad sensor 9, the column in the weight matrix $\mathbf{A}$ for sensor 9 is now (1, 0, 0) as in Figure 9, as opposed to the original column (0, 1, 0) as in Figure 7. For this changed $\mathbf{A}$, Figure 9 shows the three leading eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, and the resulting sign-based clustering assignments of sensors using these leading eigenvectors. We see that now sensor 9 is clustered in a cluster labeled as 0 while its nearby sensors belong to a large cluster labeled as 2. Thus, sensor 9 is in a small out-of-place component of a large cluster, i.e., it satisfies condition C2 in Section II. This means that our spectral clustering method has correctly identified sensor 9 as a bad sensor.

From this example we notice the following considerations in designing the spectral clustering method:

- Number k of leading eigenvectors to use. As noted earlier in Section III, k need not be larger than the number of leading eigenvalues which are significantly larger than the rest of eigenvalues (say, order of magnitude larger). Moreover, using a larger k could lead to erroneous clustering results because component signs in eigenvectors corresponding to insignificant eigenvalues tend to be unstable under round-off errors. For the simple model problem being studied here, there are only three significant leading eigenvalues (see Section III). Thus it is appropriate to choose $k = 3$ in this case.
- Filter size f for the out-of-place test. From Figure 9, we see that sensor 9 belongs to cluster 0 and that other sensors in the same cluster are separated from sensor 9 by at least six positions, with respect to the angular indexing of the model problem. The parameter f specifies the minimum amount of separation required for such an out-of-place component to meet condition C2 in Section II, and thereby be declared as a cluster for bad sensors. Increasing f will reduce false positives but at the expense of decreased probability of detecting bad sensors. A proper choice of the filter size f depends on the expected size of measurement errors that a bad sensor will produce in terms of the resulting separation in sensor indices. The error in the illustrative example described above amounts to a move from sensor 9 to sensor 1. For this separation, filter size $f = 3$ is a proper choice.
- The expected number b of bad sensors to detect. As noted in Section IV, the sign-based spectral clustering can specify no more than 2^{k-1} clusters when k leading eigenvectors are used. This means that b will need to be less than or equal to 2^{k-1}.

VI. Simulation Results on Large Systems

In this section, we present simulation results on the model problem with 100 sensors and 10 targets. We assume that sensors and targets are evenly partitioned into three groups, with antenna orientations of 0, 45 and 90 degrees. A straightforward extension of Figure 2 is used to provide weights for all sensor-target pairs. Note that similar simulation results can be obtained for other uneven setups.

We assume that some randomly selected sensors are bad sensors in the sense their measurements can be off by any amount from -100% to +100%. That is, if the original weight for a sensor-target pair is w, then the erroneous weight can be any number in $(0, 2w)$. The purpose of the simulation is to assess the effectiveness of our spectral clustering method in identifying these bad sensors.

Let B be the number of bad sensors input to the simulator, T the total number of sensors deemed as bad

Weight matrix:

Targets \ Sensors	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	1	1	0.5	0.5	0.1	0.1	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0.1	0.1	0.5	0.5	1	0	1	0.5	0.5	0.1	0.1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.1	0.5	0.5	1	1

Eigenvectors:

e_1:	0.5	0.5	0.3	0.3	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0
e_2:	-0	-0	-0	-0	0	0	0.3	0.3	0.6	-0	0.6	0.3	0.3	0	0	0	0	0	0
e_3:	-0	-0	-0	-0	0	0	0	0	0	-0	0	0	0	-0	-0	-0.3	-0.3	-0.6	-0.6

Cluster assignments:

0	0	0	0	2	2	2	2	2	**0**	2	2	2	1	1	1	1	1	1	1

Fig. 9: A weight matrix obtained by introducing an error into $\mathbf{A}$ at sensor 9 on target 0, the resulting 3 leading eigenvectors, and the new clustering assignments. The cluster label of the erroneous sensor is highlighted with a box.

sensors by our clustering method, and B' the number of sensors that the clustering method correctly identified as bad. Obviously, $B' \leq B$ and $B \leq T$. The higher the corresponding accuracy ratio $R = B'/B$ is, the better the method is. We are also interested in the number of false positives, $T - B'$. The lower the corresponding false positive ratio $(T - B')/B$ is, the better the method is.

We study the performance under various numbers k of leading eigenvectors used in the spectral clustering method, and under various values for the numbers B of bad sensors input to the simulator. In the simulator we assume that the filter size $f = 3$, as defined in Section V. We also assume that the small cluster for condition C1 in Section II is of size 1.

Figure 10 shows the performance of the spectral clustering approach in validating sensors for various numbers

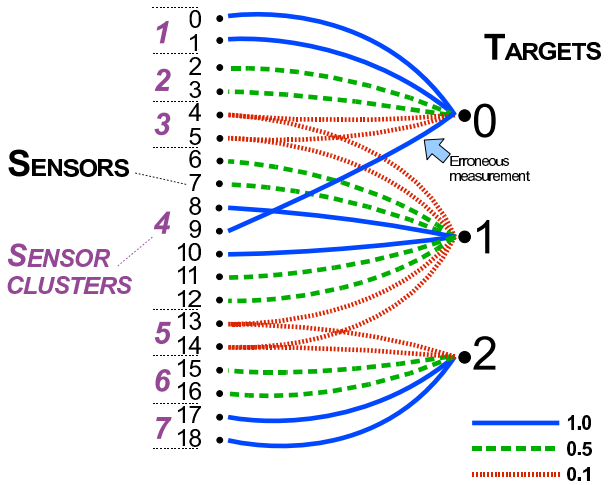


Fig. 8: Bipartite sensor-target score graph derived from that of Figure 6, but with one bad sensor (sensor 9) whose erroneous measurement induces the blue edge to target 0.

of bad sensors. We note the following from Figure 10:

1) When the number k of leading eigenvectors used increases, the performance on accuracy improves. When $k = 15$, the best performance is achieved. In this case, the B' curve on bad sensors identified is almost on the 45-degree line. Thus, almost all the bad sensors input to the simulator are successfully identified. These results are quite remarkable considering that they hold even when there are more than $2/3$ sensors out of the total of 100 sensors which are bad.
2) The number of false positives, $T - B'$, decreases when the number of bad sensors input to the simulator decreases. This is due to the fact that when there are fewer bad sensors, there will be more good sensors. This means it becomes relatively rare that condition C2 of Section II will falsely consider sensors in a small cluster as bad sensors.
3) When the number of bad sensors input to the simulation is at 5 and when five or six leading eigenvectors are used, the spectral clustering achieves almost perfect performance, namely, B' is almost equal to B and also T. Note, however, less than 7 eigenvectors should be used when the number of bad sensors are less than 10, as shown in the figure. This is because in these situations of smaller numbers of bad sensors, only a few leading eigenvalues are significant.

VII. Conclusion and Future Work

We have presented a spectral clustering approach to validating sensors via their peers. The work is motivated by the fact that it is often difficult to use instruments to validate equipment in the field and, as a result, peer-based validations can be important in these situations. We have modeled the problem as a clustering problem in the sense that sensors in the same environment will be clustered

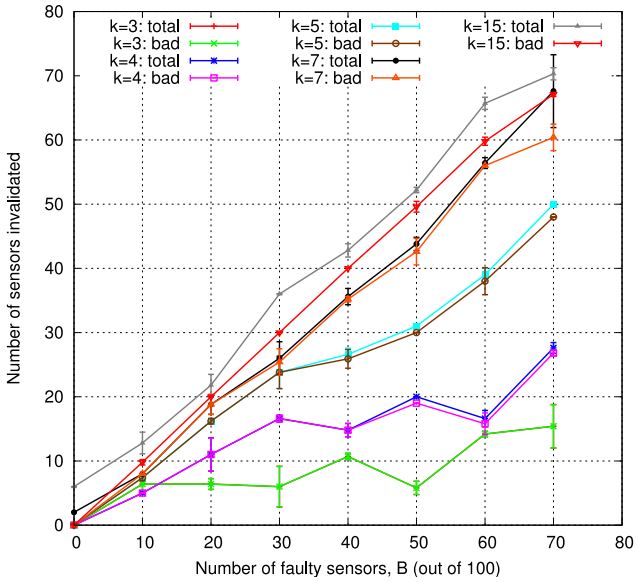


Fig. 10: Validation performance for increasing fraction of bad sensors.

together and will behave similarly as far as the reference targets are concerned, while a bad sensor can not belong to the same cluster. Thus, we can identify bad sensors through the clustering structure.

A key result of this paper is on the use of leading eigenvectors in forming clusters. Although spectral clustering is well known in the literature [9], our use in validating sensors in the field for distributed bipartite sensor networks appears to be novel. More specifically, to the best of our knowledge, our formulation of sensor indexing in support of clustering and our method of identifying bad sensors using conditions C1 and C2 of Section II, are new. We have shown analysis and examples on how this spectral clustering approach works for peer-based sensor validation. As shown in our simulation with 100 sensors and 10 reference targets, the method can identify bad targets with high accuracy and low false positive rate.

We are currently studying the performance of this spectral clustering approach against real-world measurement data such as those we have encountered in field experiments which involve tens of wireless sensors [2]. In addition, as noted earlier in the paper, our work departs from traditional approaches where no attempts are made to remove bad sensors, and emphasis is instead placed on being tolerant against erroneous measurements. We plan to investigate joint approaches which integrate both bad-sensor removal methods such as the method of this paper and bad-measurement tolerant methods such as those in localization with MDS [1] and SISR [2].

ACKNOWLEDGEMENTS

This research was supported in part by the Air Force Research Laboratory Grants FA8750-08-1-0220, FA8750-09-2-0180 and FA8750-08-1-0191.

REFERENCES

[1] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, no. 11, pp. 961–974, 2004.

[2] "Localization with Snap-Inducing Shaped Residuals (SISR) - Coping with Errors in Measurement," Harvard University, Cambridge, MA, USA, Tech. Rep. TR-03-09, Mar. 2009.

[3] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 98, 1973.

[4] A. Pothen, H. Simon, and K. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM J Matrix Anal.*, vol. 11, no. 3, pp. 340–452, 1990.

[5] D. Kempe and F. McSherry, "A Decentralized Algorithm for Spectral Analaysis," *Journal of Computer and System Sciences*, vol. 74, no. 1, pp. 70–83, 2008.

[6] H. T. Kung and B. W. Suter, "A hub matrix theory and applications to wireless communications," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 60–60, 2007.

[7] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. New York, NY: Academic Press, 1979.

[8] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling, 2nd Ed.* New York, NY: Chapman & Hall/CRC, 2001.

[9] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14*. MIT Press, 2001, pp. 849–856.