

# MODELING LINK QUALITIES IN A SENSOR NETWORK

Jure Leskovec, Purnamrita Sarkar, Carlos Guestrin  
Center for Automated Learning and Discovery,  
School of Computer Science,  
Carnegie Mellon University  
{jure, psarkar, guestrin}@cs.cmu.edu

## ABSTRACT

*Sensor networks are ad-hoc wireless networks of small, low-cost sensors, which can measure characteristics of their environment. Autonomous low-cost sensors often have limited battery life, and are prone to failures and communication losses. It is thus important to devise efficient power usage, communication and message routing schemes. In this work, we concentrate on estimating the link qualities between pairs of sensors in a natural environment. The estimation is a basic component for algorithms that optimize the power of radio transmission signal, communication schedules, and a routing scheme. Our results show that simple regression models give estimates with only 6% error. We also show the dimensionality reduction techniques help us understand the topology of the communication network and identify potential bottlenecks in the network.*

## 1 INTRODUCTION

A sensor network node is a small autonomous unit, often running on batteries, with hardware to sense environmental characteristics, such as temperature, vibrations and humidity. Such nodes usually communicate using a wireless network. A sensor network is composed of a large number of sensors deployed in a natural environment. The sensors gather environmental data and transfer the information to the central base station with external power supply [11].

Owing to the limited battery power of these sensors a very common strategy to maximize the expected lifetime is to use a better communication strategy. For this strategy to be globally optimized, we must model link qualities (LQs) between pairs of sensors. More precisely, the probability that sensor  $j$  will receive a message transmitted by node  $i$ .

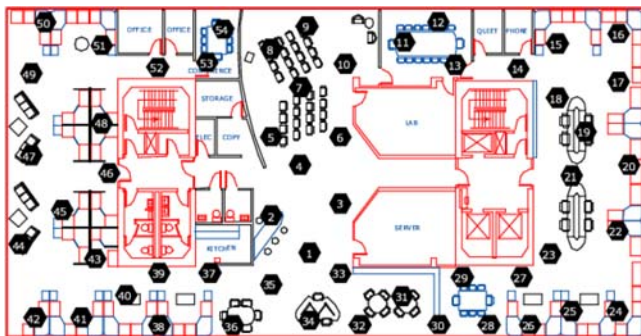


Figure 1: Sensors in the Intel Berkeley Research lab

Precise models of link qualities are the basis of many optimization and networking algorithms. For example, these models can be used to refine the communication protocols, or to decrease the number of packet collisions by tuning radio power appropriately. A proper model for link quality can also be used to select the density and positions of the sensors to ensure efficient communication. These models can also help us ensure robustness of the underlying network by finding the most unstable parts of the network, and the sensors which are critical for communication.

## 2. THE DATASET

The data comes from a deployment of 54 sensors positioned inside the Intel lab in Berkeley [3]. We have 33 days worth of data. For every 30 seconds we have a reading of binary link qualities between all pairs of sensors. There are more than 2.3 million readings in total (500MB of data). During the data collection period some nodes died and there are about 1% of readings with missing values. The readings from sensors are highly noisy and skewed due to power failures, crashes of base station, sensor failures and rapid changes in environmental conditions.

Figure 1 shows the map and the positions of 54 sensors inside the Intel Berkeley lab. The lab has a ring structure. The two ‘holes’ on the map correspond to the kitchen and the elevators. Near to the upper right corner of the map there was a cell phone base station. For this reason link qualities in the upper right part of the building are lower and decay faster with the distance than the link qualities for other parts of the building.

## 3. ANALYSIS OF LINK QUALITY

There are two obvious variables influencing the link quality: one is time and the other is location. Figure 2 shows the variance of link qualities over time of sensor 34 to all other sensors. We can observe that the variance is very low on the average. From figure 2, other experiments and measurements we concluded that link qualities do not change significantly over time.

So our work mainly concentrates on spatial link qualities. For spatial link qualities we try to relate physical position of the 2 sensors with the link quality. Having the (x,y) positions of the sensors we model the decay of link quality with the distance and build a link quality map.

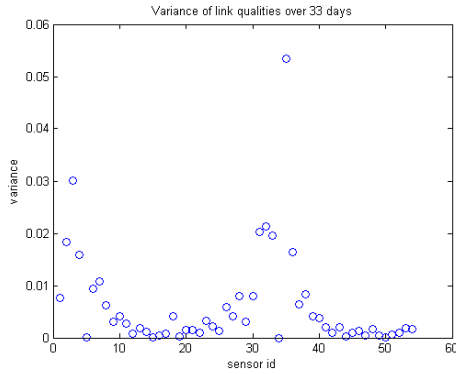


Figure 2: Variance of link quality over time of sensor 34 to all other sensors. Notice the small variance.

### 3.1 Link quality versus distance

Theoretically the strength of radio signal should drop with the square of distance, so we expected link quality to follow the same law. We analyzed a typical situation and tried to fit a function to the link qualities. We took sensor number 38 and plotted the link qualities to other sensors versus their Euclidean distance. A quadratic function had a very poor quality fit ( $R^2=0.55$ ), a power function ( $LQ=distance^{-c}$ ), with  $c$  around 2 performed even worse with  $R^2=0.19$ . On the average the quadratic regression was the best, with the average  $R^2$  around 60%. The fits for power function were not at all very satisfying, having an average  $R^2$  of 20%. Investigating even further we tried to fit a second degree polynomial to the link quality vs. distance between each pair of sensors. Figure 3 shows the amount of noise in the data. The results were also not at all promising. We also tried to use different distance measures, especially ones motivated by the connectivity graph, but this also didn't help.

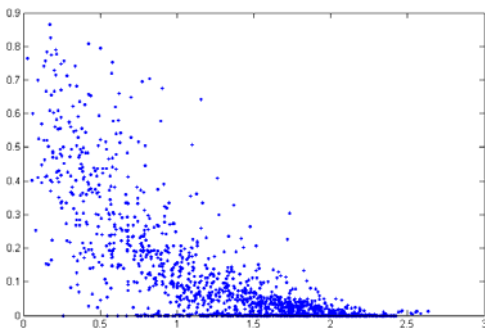


Figure 3: For all possible pairs of nodes we plot link quality versus the distance between the sensors.

### 3.2 Dimensionality reduction

So far we have been working with full 54 by 54 who-talks-to-whom link quality matrix. One way to reduce the amount of noise in the data is by using dimensionality reduction techniques. We use Principal Component Analysis (PCA) [2]. We perform PCA on 54 by 54 link quality matrix. We essentially do a metric multidimensional scaling [1] on the

data to learn the underlying coordinates in a 3-dimensional Euclidean space, namely the latent space. The first 3 eigenvectors explain around 70% of the variance of the data. Increasing to 4 dimensions covers additional 5% more variance so we decided to continue experimenting with 3 dimensional latent space.

Figure 4 shows the latent coordinates of the sensors in a 3-dimensional space. We can clearly observe the two rings we had seen on the map of the lab (Figure 1). This means we are able to reconstruct the map of the lab using only the link quality data. This also implies that the sensor locations should be good attributes for modeling the link qualities. Notice also the big gaps in the rings. This shows the “hole” and a potential bottle neck in the communication network and suggests deploying more sensors in that part of the lab. A close inspection of figure 6 reveals a set of nodes outlining a big hole in the graph. If any two of these nodes dies the communication between two halves of the network will seriously interrupted. These nodes are critical for the communication of the network. This suggests that one needs to deploy more sensors in this part of the lab to increase the robustness of the network. Thus we see that the multidimensional scaling approach reveals some very important and interesting patterns in the data, besides matching the true map of the sensor locations.

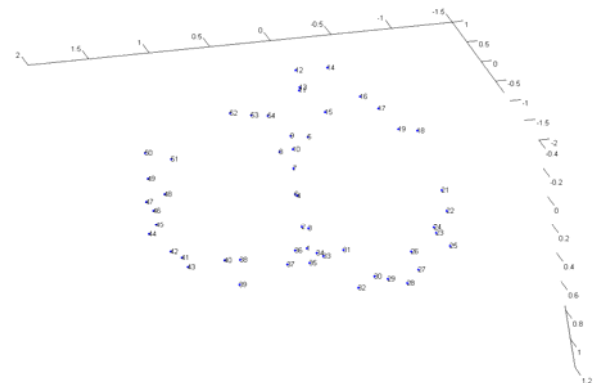


Figure 4: The projection of link quality data on first 3 principal components. Notice the two rings very similar to actual map of the lab.

### 3.3. Link qualities via dimensionality reduction

Now we use the notion of latent space to construct a 2 level model for link quality prediction. We will first learn a regression model to map from sensor  $(x, y)$  lab coordinates to the 3D latent space position. We then use the principal components to map from 3 dimensional latent space to the original 53 dimensional vector of link qualities. Figure 5 more clearly depicts our idea.

Note that we learn 3 separate regression models, each from mapping from  $(x,y)$  lab location to of the sensors to a particular latent space dimension. We use linear, quadratic and cubic regression. Also note that we have only 54

training instances. We performed leave one out cross validation and report the mean absolute error. Table 1 shows results on test and training set for the 3 models. Notice that quadratic model performs best and the cubic model overfits the data.

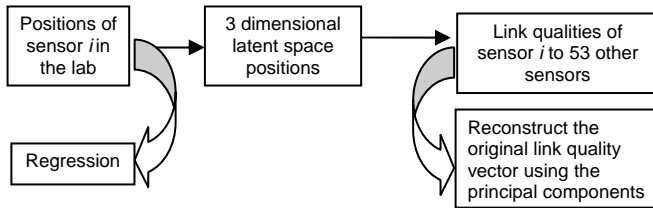


Figure 5: The 2 level model. We learn 3 regression models to map from  $(X,Y)$  positions to latent space positions. We then use principal components to map from the latent space to the link qualities.

Regression type	Training Set Mean Error	Test Set Mean Error
<b>Linear</b>		
LS dimension 1	0.073	0.078
LS dimension 2	0.229	0.244
LS dimension 3	0.124	0.131
<b>Quadratic</b>		
LS dimension 1	0.045	0.051
LS dimension 2	0.078	0.090
LS dimension 3	0.071	0.083
<b>Cubic</b>		
LS dimension 1	0.038	0.050
LS dimension 2	0.077	0.098
LS dimension 3	0.062	0.099

Table 1: Performance of the regression mapping from  $XY$  to the latent space

We also examined at the scatter plot of predicted latent space positions and the true latent space position of a particular sensor for the quadratic model. For all three dimensions we found that the residuals are well distributed, and concluded that the quadratic model is suitable.

So far we build the model to map from physical sensor positions to 3 dimensional latent space positions. Now we perform the last step where we use principal components to map from 3 dimensional latent space to original 54 dimensional vectors of link qualities. Using quadratic regression model and 3 dimensional latent space the final mean square error of link qualities is 0.14. If we increase the number of dimensions in the latent space to 4, the error increases to 0.145. Increasing the number of dimensions further to 10, gives the average mean error of 0.20.

As we pick more latent space dimensions the mapping from latent space to the link quality gets more accurate. On the other hand the mapping from  $(x,y)$  positions gets less accurate and the combination is worse performance. The problem with learning mapping to higher latent space dimensions is that they contain more noise, so the regression

gets unstable with large errors. Using cross validation we get the best results when using 3 dimensional latent space.

### 3.4 Direct approach

We also tried a more direct approach of learning link quality between a pair of sensors given the lab coordinates of them, using a regression model. In this case we have 2862 (54 squared) training examples each having 4 real attributes (locations of the two sensors). As in previous section we perform 10 fold cross validation and report average mean error on training and on test set.

We compared 3 classes of algorithms: normal least squares polynomial regression, a variant of logit transform and regression Support vector machines [4] using polynomial and radial kernels. For the logit transformation our idea was to transform the link qualities (which are probabilities and thus reside on interval  $(0,1)$ ) to the whole real space. Our hypothesis was that it may be easier to learn the link qualities spread out over the whole real space. In this case we transformed the link quality  $LQ$  with the equation  $LQ' = \log(LQ/(1-LQ))$ . We then learned the regression model, performed the inverse logit transform and measured the mean error.

Regression type	Training set Mean Error	Test set Mean Error
<b>Normal</b>		
Linear	0.108	0.108
Quadratic	0.087	0.088
Cubic	0.086	0.088
<b>Logit transform</b>		
Linear	0.409	0.409
Quadratic	0.412	0.412
Cubic	0.411	0.411
<b>SVM</b>		
Linear	0.119	0.119
Quadratic	0.093	0.093
Cubic	0.090	0.090
6 deg polynomial	0.082	0.083

Table 2: The performance of various regression techniques

Table 2 shows the results for the 3 classes of regression algorithms we tested. Our first observation is that even simple linear regression outperforms our 2 level model by 4%. We observe a 2% improvement of quadratic and cubic model over the linear model. Next observation is that logit transform performs far the worse. It performs a bit better than random guessing which has the mean error of 0.5. The SVM with polynomial kernels have similar performance as normal least squares regression using the same degree polynomial as in SVM kernel. We observe that even very high degree polynomial kernel of degree 6 does not help to fit the data very well.

The radial kernel beats all other techniques with a mean error of around 6% on both training and test set. Radial

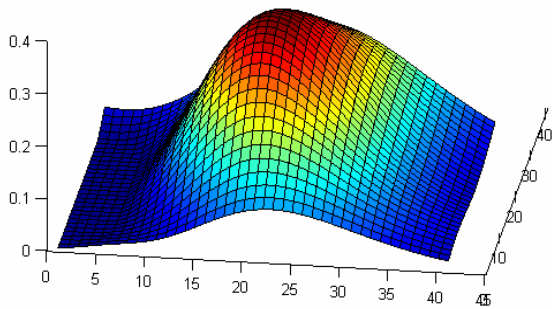


Figure 6(a): Link quality map for a sensor in the center of the Intel lab.

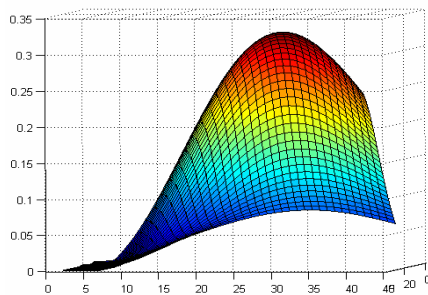


Figure 6(b): Link quality map for a sensor in the corner of the lab.

kernel is especially appropriate for this task, since it has a bell shape, which means that a link quality basically decays in a bell shape with the distance.

Having built the model we can now look at the link quality map for a particular sensor. We fix the location of the first sensor and then for every position of the second sensor we use the model to obtain the link quality. We call this the link quality map.

Figure 6 shows the two examples of link quality maps generated using the SVM radial kernel. Figure 6(a) shows the case when we positioned the sensor in the center of the lab. We observe a bell shape decay of link qualities. Notice how the link qualities are very low on the left middle part of the figure. Notice also that on the left side in the top corner link quality is better than left middle and left bottom corner. This is because left bottom part is further away and better hidden behind the wall. There is also a cell-phone base station on the left part of the map, which further decreases link qualities. One would falsely expect that link qualities in the holes of the two rings (kitchen and the elevator) should be close to zero. This is not the case since we have no data points inside those rings and the link quality just gets interpolated over that empty space. Figure 6(b) shows the case when the sensor is positioned into a corner of the lab. We observe a similar bell like decay of link quality away from the sensor.

#### 4. RELATED WORK

The power efficiency plays a central role in sensor networks. People investigated scalable and power-efficient protocols [5], power management [6], efficient routing [7]

and querying in sensor networks [8]. The most related to our work are [9] and [10]. Our findings are in accordance to conclusions in [9] that the link characteristics are far from the theoretical models. However, these works survey the detailed link stability but not its effect on positioning, while our work concentrates in modeling link qualities in a natural environment and how they change with positions of the sensors.

#### 5. CONCLUSIONS

In this work, we showed exploratory results on the modeling of link qualities in a sensor network. Since link qualities are often invariant with respect to the time, we focused on the spatial aspect. In our experiments simple regression techniques were quite effective. However, in our comparisons, support vector regression with radial kernel was the best performing approach. Intuitively, link qualities decay with distance, a property captured effectively by this model. We also showed how dimensionality reduction techniques can be used to analyze link qualities, identifying critical nodes and sparsely connected parts of the sensor network.

#### References

- [1] R. Sibson. Studies in the robustness of multidimensional scaling : Perturbational analysis of classical scaling. *J. Royal Stat. Soc. B, Methodological*, 41:217, 1979.
- [2] G.H. Golub and C.F. Van Loan *Matrix Computations*. The John Hopkins University Press, 1996
- [3] <http://db.lcs.mit.edu/labdata/labdata.html>
- [4] N. Cristianini and J. Shawe-Taylor. Support Vector Machines. *Cambridge University Press*, 2000.
- [5] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy efficient wireless sensor networks. In *Mobile computing and networking*, 2001.
- [6] M. Bhardwaj, A. Chandrakasan, and T. Garnett. Upper bounds on the life time of sensor networks. In *IEEE International Conference on Communications*, 2001.
- [7] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Mobicom*, 1999.
- [8] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-Driven Data Acquisition in Sensor Networks. In *VLDB 2004*.
- [9] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks. *Technical Report UCLA/CSD-TR 02-0013*, 2002.
- [10] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *Sensor and Adhoc Communication and Networks*, 2004.
- [11] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. *Tech. R. IRB-TR-02-006*, Intel Research, 2002.