ERROR MODELS FOR LIGHT SENSORS BY STATISTICAL ANALYSIS OF RAW SENSOR MEASUREMENTS

F. Koushanfar[1], M. Potkonjak[2], A. Sangiovanni-Vincentelli[1]

[1]EECS Department, UC Berkeley, Berkeley, CA 94720, USA
[2]CS Department, UC Los Angeles, Los Angeles, CA 90095, USA

ABSTRACT

Error modeling is a procedure of quantitatively characterizing the likelihood that a particular value of error is associated with a particular measured value. Error modeling directly affects accuracy and effectiveness of many tasks in sensor-based systems including calibration, sensor fusion and power management. We developed a system of statistical techniques that calculate the likelihood that error of a particular value is part of a measurement. The error modeling approach has three steps: (i) data set partitioning; (ii) constructing the error density model; and (iii) learn-and-test and resubstitution-based procedures for validating the models. The data set partitioning identifies a specified percentage of measurements that have the highest negative discrepancy between sensor and standard measurements. The partitioning step employs data fitting models to identify compact curves that represent the partitioned subsets. The error density modeling uses the compact curves to build the probability density function (PDF) of the error. For validation purposes, we use a resubstitution-based paradigm.

1. INTRODUCTION

We developed a new approach for derivation of error models. While the standard procedure is to use error models to enable calibration, in a variant of our approach, we use calibration-based techniques to obtain error models. We demonstrate our procedure on a set of measurement pairs from uncalibrated light sensors vs. an exact light meter sensor at the same position. All of the error model-building techniques are demonstrated on a set of light sensors. Neverthless, the techniques are general and can be easily retargeted to sensors of different modalities.

Calibration is the process of mapping the individual sensor response to a desired standardized response. In general, individual sensor calibration (micro-calibration) is a type-specific task, where the type of the sensor involved and the accuracy requirements change the calibration problem. In an ad-hoc sensor networks consisting of a set of inexpensive, energy and storage constrained sensor nodes, the calibration method should satisfy the application constraints before optimizing for the accuracy of the calibration map. Most calibration methods in sensor networks have addressed calibration of the devices used in location discovery. Whitehouse and Culler [7] propose iterative use of linear least square regression to fit the experimental data from the location sensors for individual sensors and for a system of location discovery sensors. Bychkovskiy et al [1] attempt to calibrate the sensors locally. Their assumption is that physically close sensors have high temporal correlations. They start by calibrating pairs of close sensors and then formulate a global problem to find the best way to satisfy all pair-wise relationships simultaneously. Ihler and Fisher [4] use graphical models to formulate the self-calibration model assuming that the calibration information is spatially local. Aside from calibration, error models are also useful in a number of tasks in sensor networks such as sensor fusion [6].

2. PRELIMINARIES

Our experiment concerns two different light sensors. The first sensor is a photovoltaic light detector. A photovoltaic light detector is a miniature silicon solar cell that converts light impulses directly into electrical charges that can easily be amplified, using a transistor, to activate a control mechanism. Unlike a conventional photo diode or transistor, it generates its own power and does not require any external bias. The silicon cell is mounted on a 0.31" x 0.23" x 0.07" thick plastic carrier and has pc leads on 0.2" centers. The second sensor is a light meter. The light meter has a resistance of 20W in the light and 5kW in the dark. It is mounted on a 1" x 0.85" x 0.07" thick plastic encapsulated ceramic package with 0.57" long leads on 0.75" centers. The accuracy of the light meter is within +/- 1%. The photovoltaic light sensor is widely used in a number of wireless networked systems, especially in sensor platforms that are based on MICA-I and MICA-II nodes [8].

We positioned the photovoltaic sensor and the meter sensor in close proximity to each other and with the same angle in a dark room. We positioned a point light source at a far distance from the sensors. This point light source was moved at slow speed in an arbitrary direction. Since the distance from the source was changed during this movement, both light sensors were recording a pairs of measurements of different intensities. We conducted these measurements on six photovoltaic sensors.

Figure 1 shows the overall flow of our approach. The input to our error modeling system is a set of measurements that consist of pairs of data recorded by the photovoltaic sensor and the light meter at identical time instances. The primary goal during this step is to obtain pairs of measurements that cover as uniformly as possible the complete dynamic range of both sensors. Once the data is available, the first step of our procedure is to identify subsets that consists of exactly *k* points. These points, according to a measure specified by the user, have the largest negative discrepancy between the values at the light meter and at the photovoltaic sensor. For this task, we implement and

0-7803-8692-2/04/$20.00 ©2004 IEEE.

experiment with measures that are based on the relative value of the residuals and the level of consistency in terms of both weighted and rank-based criteria.

Once we have a dataset, we use either parametric or nonparametric data fitting schemes to provide a compact representation of the subset of the points in a two dimensional space. The photovoltaic and light meter sensor readings form the two axes. We experimented with a number of techniques, including linear fit, linear fit after transformation and several non-parametric schemes. The last step of our approach is derivation of the quality of the proposed models using resubstitution techniques. For the sake of brevity, the details are presented only in our technical report [5].

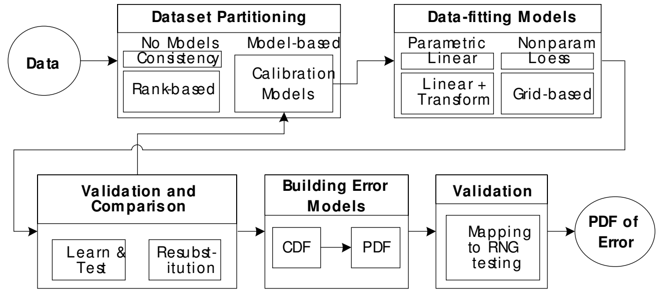

Figure 1 - Global flow of the error modeling procedure

3. SUBSET IDENTIFICATION AND MODEL BUILDING

In this section, we show techniques for partitioning the data into subsets. These subsets are used in later sections for building the error density models. In the first subsection, we introduce a number of subset identification methods. After that, we show a detail analysis and experimental results for model-based subset identification.

3.1 Subset Identification Methods

The first step of our procedure for building of an error model is the identification of the subsets of data where each subset contains *2K%* of data (where *K* is user-defined) in such a way that the points in the subsets correspond to points that have the lowest predicted values with respect to developed error models. More specifically, the input to this step is a set of points placed in a two dimensional space. The *y*-coordinate of each point corresponds to the measured value in a specific sensor measurement, while the *x*-coordinate indicates the correct corresponding value measured by the meter. The final goal is to produce a curve that partitions data into two subsets in such a way that *2K%* of data is below the curve and the reminder is above. We start by identifying *2K%* of the data points and remove them from the dataset. If we perform the procedure again, we can iteratively identify all the subsets, where each contains *2K%* of original points. The strategic objective is to place a curve in such a way as to maximize the likelihood that the future measurements correspond to points that would be placed exactly in that proportion above and below the curve, regardless of the amplitude of the actual measurement. To address the partitioning problem, we have developed three generic methods: (a) residuals-of-calibration method, (b) consistency-based method, and (c) rank-based method. For the sake of brevity, we present only the first method. The two other models are described in our technical report [5].

The first method employs the residuals of the calibration curve in order to identify the subsets. The intuition behind the method is that the points that could not be fit well by calibration curve and have high residuals are most likely the ones that are below the specified *2K%* points. Therefore, this procedure first establishes a calibration curve and consequently sorts all points in an increasing order with respect to their residuals of the calibration model. Note that before sorting, we can apply preprocessing steps depending on the intuition provided by the exploratory data analysis. For example, one can analyze all the residuals vs. the measured or the predicted value. In Figure 2, we show the partitions generated by two data fitting models, where the partitions divide the space into 10 subsets, each containing 10% of data. We also superimpose the data fitting lines on the partitions as an intermediate step toward obtaining the PDF. The figure shows data fitting lines by linear model (left) and by linear model with logarithmical transform (right).

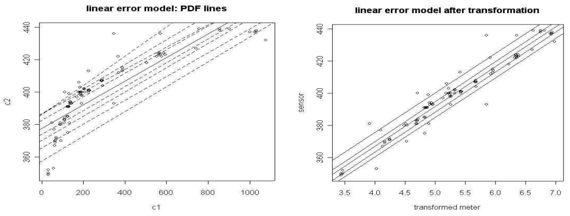

Figure 2 – Calibration lines are dividing the dataset into subsets each with 10% of the original data: Linear model (left), linear model+logarithmic transform (right).

3.2 Modeling of Sensor Readings vs. the Meter

In this subsection, we analyze a system of parametric and non-parametric statistical modeling techniques to find a model of the readings from the photovoltaic sensors vs. the meter.

Analysis of linear model is shown in [5]. We also examined several different transformations and partitioning on the data in conjunction with the linear model. In Figure 3, we show the fitted model for three different such methods on one of the sensors in the experiment. The leftmost plot shows the case where we use two partial linear models to describe the data. A vertical line on the plot shows the breakdown point for the two linear models. The middle plot shows the linear model after we get the square root of the meter data. The rightmost plot shows the linear model after we apply a logarithmic transform on the meter data. Among all transforms, the logarithmic transform yielded the best results for all sensors in terms of the R^2 value of the linear model and the summary statistics of the resulting errors. The R^2 value of the linear model using all the available data was in the range of [83%, 95%] for all of our sensors.

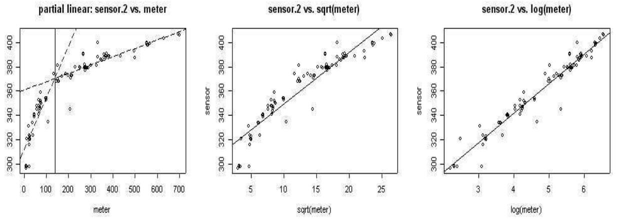

Figure 3 – Advanced linear models: two partial linear fits (left), linear fit with square root transform (middle), linear fit with logarithmic transform (right).

We use two non-parametric modeling approaches. The first method for building a non-parametric model starts with obtaining the estimates of the conditional probabilities of the signal that we are trying to predict (response variable) vs. the predictor (explanatory variable). The sensor readings at time i are shown by *x[i]* and *y[i]* for the explanatory (meter) and response (photovoltaic sensor) variables respectively. Since our data was discrete, we will present the approach assuming a discrete data model. Note that the techniques are generic and can be easily adapted to the case of continuous data in a straightforward way. We start by building a 3D histogram of the conditional probability *p(y|x)* of *y* for a given value of *x*. For each pair of sensors and meter *(y,x)*, we use some number of data points to build up this histogram. During the testing phase, for each observed value of *x*, we can use this histogram to produce an estimate of *y*. There are a number of ways in which the histogram can be used to find the prediction model of *y* for the given values *x*. We build two models; the first uses all data points and constitutes an optimal prediction model since this yields a provable upper bound on the achievable prediction. The second model uses just *P%* of data from our learning phase, which we then use for prediction on test data. Note that *P* is a user defined value and is the ratio (in percentage) of the number of points used in the learning set to the number of all points.

We considered two predictors based on the histogram for the conditional probability. First, while predicting the value of *(y|x=k)*, we will minimize the L_1 error measure if we calculate the median of all *y[i]*'s that occur in the stream when *x[i]=k*. A second predictor, that minimizes the L_2 error norm, is to use the average value of the corresponding *y[i]*. Other common error models such as relative L_1, L_2, and L_∞ can be similarly computed with minimal preprocessing. For a given error norm, no other predictor would explain the variations of y due to x with less error. Finally, we also use loess non-parametric modeling paradigm [2].

We compare the different data fitting methods using learn-and-test and resubstitution-based paradigms [3]. In Figure 4, we show the boxplots of the average relative absolute error for each prediction model for a fixed percentage of learn data (*P*=65%). Each boxplot consists of a 100 datapoints, where each datapoint corresponds to the average error of a model built with the resampled learn data. Furthermore, the left plot shows the average absolute relative learning error for 65% learn data. The right plot shows the average absolute testing error for 35% testing data. The boxes on each plot show the relative error for (1) simple linear model, (2) linear model with logarithmic transform, (3) non-parametric histogram-based model with L_1 error measure, (4) non-parametric histogram-based model with L_2 error measure, (5) non-parametric loess fit with a span of 0.2, (6) non-parametric loess fit with a span of 0.3, and finally (7) non-parametric loess fit with a span of 0.4.

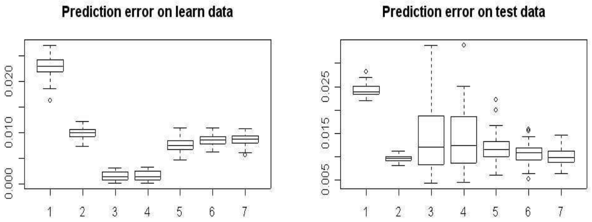

Figure 4 – Boxplots of the prediction errors: (1) Linear model, (2) linear model and logarithmic transform, (3) non parametric –L_1, (4) non parametric – L_2, (5) loess – span=0.2, (6) loess – span=0.3, (7) loess – span=0.4.

The simple linear model has the highest error among all of our models for both learning and testing phases. As we can see on both plots, the error in both learning and testing phases dramatically decreases after applying the logarithmic transform. The lower bound for the learning phase error is given by the non-parametric histogram-based method. As we can see in the left-side plot, both L_1 and L_2 measures of error give comparable lower bound results for the case of histogram-based method. However, the testing phase errors for both histogram-based models have large fluctuations. The reason is that the number of points in the experiment was not high enough to build a smooth model that has a value for all possible outcomes in the experiment. A possible way to suppress the testing phase error is to perform smoothing in conjunction with histogram building. The loess nonparametric modeling approach automatically performs smoothing within its span window. We show the error performance of the loess method for three different spans. As we increase the window sizes, we lose the granularity of the model and the learning phase error tends to increase, while the smoother curves resulting from wider span windows perform better in the testing phase. Overall, by comparing the different models we conclude that the best overall calibration model for fitting the test data are the linear model with logarithmic transform and the loess non-parametric fit.

4. ERROR DENSITY MODELS

We explain how we make a transition from a summary of partitioned data as presented in section 3.1 to the complete error model. The conversion is accomplished in the following way: we first superimpose a set of lines y=a$_i$ (i=1,.., N) over the two dimensional plot of the calibrated models. The plots are placed in the coordinate system where we have measured sensor values on the y-axis and we have the correct meter values on the x-axis. Figure 5 shows a case where 10 lines were superimposed on the top of our light sensor measurements vs. the meter readings for both simple and transformed cases.

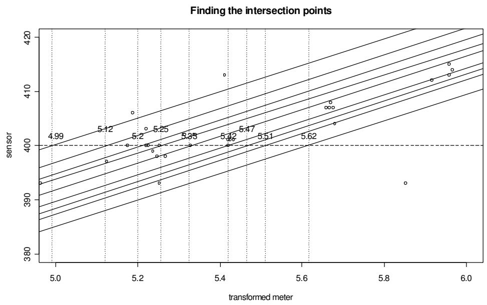

Figure 5 – Finding the intersection of the line y=400 with the K% lines from Figure 2 (right).

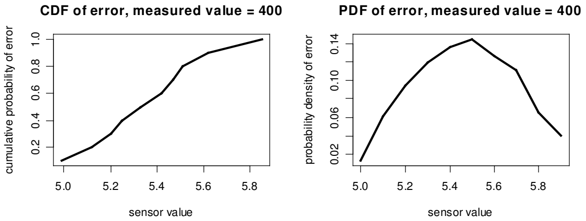

Figure 6 – CDF of the error (left) was formed from the intersection points shown in Figure 5. PDF of the error is formed by differentiation and smoothing of CDF (right). Both cases are shown for sensor value of 400.

Once when we have the superimposed lines, for each line we repeat the following procedure. We find the intersection of the line $y=a_i$ with each of the fitted plots using either analytic of numerical techniques. We build an error model for a specific $y=a_i$ as shown in Figure 6. Specifically, we first build a CDF by placing values of $x_j=b_j$ on x-axis, where b_j is the x-coordinate of intersection of each of the plots with the line $y=a_i$. On the y-axis of Figure 6 (left), we assign the value between 0 and 1 that corresponds to the percentage of the data that is modeled using a specific curve that has that percentage of data points below the curve. If we connect the lines using either analytic (e.g. piecewise linear) or statistical interpolation techniques (e.g. least linear square polynomial fit), we obtain error models for our measurements $y=a_i$. In Figure 6, we use a simple piecewise linear method to form the CDF. In both cases, we enforce that the starting point for the approximation has the value 0, while the end point has a value one. PDF is obtained from the CDF using numerical differentiation. Figure 6 (right), illustrates the PDF for light measurements for the sensed value y=400. Finally, after we repeat this procedure for each sensed value a_i, we obtain a three dimensional PDF that has the measurement on y-axis, the meter value on the x-axis, and the PDF function of the error model on the z-axis. In Figure 7, we show a three-dimensional error model for one of the sensors that was developed using the rank-based partitioning criteria along with linear model and logarithmic transform.

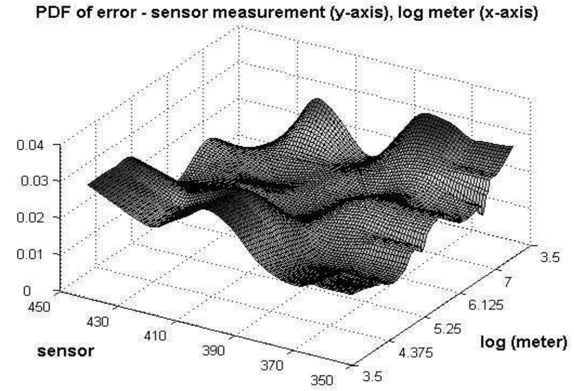

Figure 7 – PDF of error (z-axis) for different values of sensor measurement vs. meter measurement.

We applied all combinations of three subsets identification techniques and five subsets of data fitting schemes on each set of sensor data. In all cases, the superior performances in terms of low prediction errors were achieved by the combination where the discrepancy of residuals and linear model after logarithmic transform were used. The results for these two methods are shown in Figure 5, Figure 6 and Figure 7.

5. REFERENCES

[1] Bychkovskiy, V. Megerian, S. Estrin, D. Potkonjak, M. “Calibration: A Collaborative Approach to In-Place Sensor Calibration.” *IPSN*, pp. 301-316, 2003.

[2] Cleveland, W. S. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Statist. Assoc.* 74, pp. 829–836, 1979.

[3] Hastie, T. Tibshirani, R. Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, 2001.

[4] Ihler, A. T. Fisher, J.W. “Nonparametric Belief Propagation for Self-Calibration in Sensor Networks”, *International Workshop on Information Processing in Sensor Networks (IPSN)*, pp. 225-233, 2004.

[5] Koushanfar, F. Potkonjak, M. Sangiovanni-Vincentelli, A. Error Models for Light Sensors, UCB Tech Report, 2004.

[6] Suranthiran, S. Jayasuriya, S. "Optimal Fusion of Multiple Nonlinear Sensor Data." IEEE Sensors Journal, vol. 4, no. 5, pp. 651-663, October 2004.

[7] Whitehouse, K. and Culler, D. “Macro Calibration in Sensor/Actuator Networks”. *ACM MONET,* June 2003.

[8] http://www.xbow.com/Products/