Automatic calibration of sensor-phones using gaussian processes

Richard Edward Honicky

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2007-34
http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-34.html

March 20, 2007

Acknowledgement

Thank you to Dean A. Richard Newton and Professor Peter Bartlett for guidance on this research.

Automatic calibration of sensor-phones using gaussian processes with explicit basis functions

Tech Report
Department of Computer Science
UC Berkeley

R. J. Honicky
honicky@cs.berkeley.edu

March 20, 2007

Abstract

In the context of environmental sensors embedded into location-aware cell phones, this paper explores how to automatically calibrate the bias of each sensor based on readings from nearby phones, thus exploiting the mobility of the sensors. Gaussian process regression is used interpolate values and also infer the bias of the sensors. The impact of dense vs sparse sampling is explored, and future strategies for efficiency, gain calibration, and cyclical covariance are suggested.

1 Introduction

Wireless sensor networks have continued to increase in importance as scientists find new ways to exploit the fine granularity data that these networks can provide [2]. At the same time, as cell phone processors become more integrated, better connected, faster, cheaper, and less power consuming, and as manufactures search for new uses of the space and power freed up by highly integrated processors, the cell phone increasingly looks like a full-fledged PC. Finally, localization technologies (e.g. GPS) continue to improve as demand increases for faster and more accurate location fixes, and this improvement is also making its way into cell phones.

Wireless sensor networks, however, still suffer from several problems (including power, node mortality, calibration and routing [2]) which limit their applicability to societal-scale problems. Nonetheless, demand for geo-coded data and all manner of GIS databases continues to increase as scientists, economists, and policymakers seek to guide their decisions more precisely and in ways more appropriate to specific problems' contexts [1]. My research seeks to build on the initial successes of wireless sensor networks in providing fine granularity data, while at the same time leveraging the enormous economies of scale and increasing integration of cell phones to provide a societal-scale environment-sensing solution.

The larger project on which I am working explores integrating environmental sensors into location-aware (GPS enabled) cell phones, and examines issues surrounding hardware, data collection software, data storage and dissemination, statistical techniques for gathering and understanding the state of the environment from noisy and sometimes sparse observations, the correlation of sensor readings with semantic location (e.g. "at home"), and privacy concerns.

This paper will explore two modeling aspects of this problem. The first is interpolation of the environmental state in sparsely sampled areas. The second is based on the fundamental observation that phones frequently come into close proximity with one another. We can therefore probably use the mobility of cell phones to our advantage. When different sensors are in the same location at the same time, they will give slightly different readings due to miscalibration. We can use these differences to determine the miscalibration of each sensor, which in turn will allow increased accuracy in sparsely sampled areas.

This paper focuses on gaussian process regression (GPR) because GPR provides a clear statistical interpretation and good variance estimates. Variance is both interesting in its own right and is also important for inferring sensor bias, choosing when to sample, and how to thin data. While finite element methods are often the fastest method for fitting a surface to noisy data, they usually do not provide variance estimates, and often do not correspond to the MAP estimate of the underlying function. Support vector regression offers another alternative to GPR. While support vector regression minimizes the risk of the regression, again the fitted function does not necessarily have a clear probabilistic interpretation, and does not provide a variance estimate. In fact, regularization has a close relationship with the idea of placing a prior over a space of functions. As such, an ϵ-insensitive loss function can also be applied to Gaussian processes regression to increase the sparsity of the solution.

2 Gaussian process regression

2.1 Basic Gaussian process regression

Gaussian process regression is a kernel method, and as such, shares many similarities with other kernel methods. A Gaussian process is a Gaussian distribution over functions, with a covariance function which determines the covariance between the Gaussian random variables at any to points. As with other kernel methods, we can view Gaussian processes from either a weight-space perspective, or a function-space perspective. I will summarize Rasmussen's and Edwards' well considered analysis [4].

In standard linear GP regression, we consider functions of the form

$$f(x) = \phi(x)_T w + \varepsilon,$$

for some basis function $\phi : d \mapsto n$, and a Gaussian random vector $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. By putting a zero mean Gaussian prior over the weight vector

$$w \sim \mathcal{N}(O, \sigma_p),$$

we favor solutions with low weights. This is analogous to the complexity term in a regularization method, since it causes the algorithm to favor functions with a small weight vector.

I will focus on the function space perspective, since for Gaussian processes, the function space perspective is far more intuitive. From a function space perspective, a Gaussian process can be specified in terms of its mean and covariance functions:

$$\begin{aligned} m(x) &= \mathbb{E}[f(x)] \\ k(x_p, x_g) &= \mathbb{E}[(f(x_p) - m(x_p))(f(x_q) - m(x_q))] \end{aligned}$$

Plugging in $f(x) = \phi(x)^T + \varepsilon$, and recalling that w is drawn from a zero mean Gaussian, we have

$$\begin{aligned} \mathbb{E}[f(x)] &= \mathbb{E}[\phi(x)^T w + \varepsilon] = 0 \\ \mathbb{E}[(f(x_q) - m(x_q))(f(x_r) - m(x_r))] &= \mathbb{E}[f(x_q) f(x_r)] \\ &= \mathbb{E}[(\phi(x_q)^T w + \varepsilon_q)(\phi(x_r)^T w + \varepsilon_r)] \\ &= \mathbb{E}[(\phi(x)^T w w^T \phi(x') + \varepsilon_q \phi(x_r)^T w + \varepsilon_r \phi(x_q)^T w + \varepsilon_q \varepsilon_r] \\ &= \phi(x)^T \mathbb{E}[w w^T] \phi(x') + \mathbb{E}[\varepsilon_q \varepsilon_r] \\ &= \phi(x)^T \Sigma_p \phi(x') + \sigma_n^2 \delta_{qr}, \end{aligned}$$

where δ_{qr} is the Kronecker delta.

Furthermore, since ϕ always appears as a quadratic term with σ_q (a covariance matrix, and thus positive definite), we can kernelize ϕ as

$$\begin{aligned} k(x_q, x_r) &= \phi(x_q)^T \sigma_q \phi(x_r) + \sigma_n^2 \delta_{qr} \\ &= \phi(x_q)^T \sigma_q^{1/2} \sigma_q^{1/2} \phi(x') + \sigma_n^2 \delta_{qr} \\ &= (\Sigma_q^{1/2} \phi(x)) \cdot (\Sigma_q^{1/2} \phi(x')) + \sigma_n^2 \delta_{qr} \\ &= \psi(x) \cdot \psi(x') + \sigma_n^2 \delta_{qr}, \end{aligned}$$

where $\left(\Sigma_p^{1/2}\right)^2 = \Sigma_p$. If X is a vector of observations, then we have

$$\operatorname{cov}(y) = K(X, X) + \sigma_n^p I \tag{1}$$

Writing the joint distribution as

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

Conditioning on X, y and X_*. We have

$$\bar{f}_* = \mathbb{E}[f_* | X, y, X_*] = K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1} y \tag{2}$$

$$\operatorname{cov}(f_*) = K(X_*, X_*) - K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1} K(X, X_*) \tag{3}$$

2.2 Choosing a kernel

The spacial distribution of atmospheric gasses is usually modeled as Gaussian, meaning that the correlation between readings at two locations decays quickly as the two locations become further apart. Thus we also use a radial basis function kernel for the experiments in this paper:

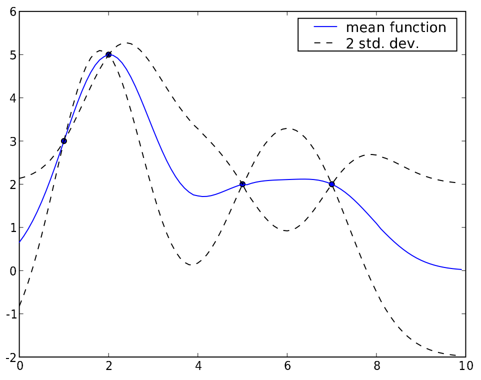
$$k(x_q, x_r) = \exp\left(-\frac{1}{2}|x_q - x_r|_2^2\right)$$

As our data set grows, however, inverting the noisy covariance matrix $[K(X, X) + \sigma_n^2 I]^{-1}$ will take $O(n^3)$ time if we are not careful. Since $[K(X, X) + \sigma_n^2 I]^{-1}$ is positive definite, we can do the inversion using a Cholesky decomposition. The Cholesky decomposition can take advantage of sparsity in K, so that the inversion grows as $O(n'^3)$, where n' is the number of non-zero elements in k.

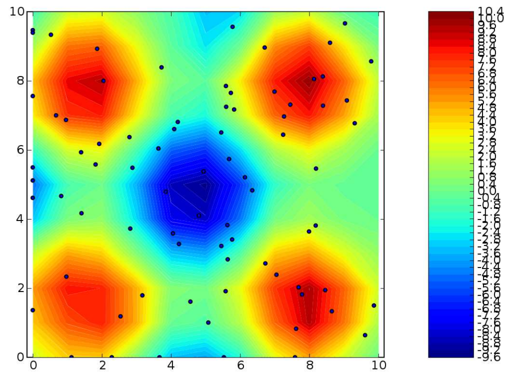
The radial basis kernel, however, has infinite support, so in order to reap the benefits of the Cholesky decomposition's efficiency, we must approximate the radial basis kernel zero at distances greater than, say, three standard deviations, $3\sigma_n$. See Section 4 below.

Figure 1(a) shows a Gaussian process inferred from four test points, with the mean function in solid blue. This function was inferred using the truncated kernel as described above. Notice how the process mean function tends towards 0 in the absence of test points. This is due to the zero mean prior process.

Figure 1(b) shows a contour plot of the mean of a 2 dimensional Gaussian process. The sample points were chosen by choosing evenly spaced points along twelve random lines across the field, and sampling from $f(x_1, x_2) = 5(sin(x_1) + sin(x_2)) + \varepsilon_i$. This crudely simulates a series of sensor-phones moving through a potential field and sampling periodically.



(a) A 1d function with minimal noise



(b) The function $5(sin(x) + sin(y))$, inferred from noisy sample points taken along random lines

Figure 1: Inferred Gaussian process functions

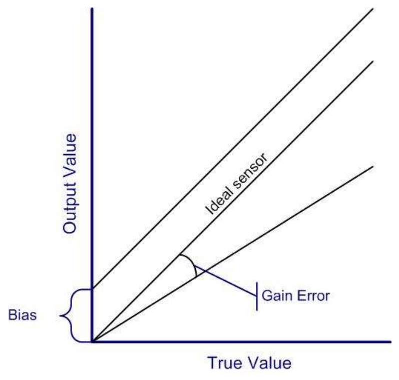


Figure 2: Gain error and bias vs an ideal sensor

3 Inferring sensor bias

Over time sensors tend to become mis-calibrated, and must be calibrated in order to remain useful. An important advantage to using a cell phone as a sensor platform arises out of the mobility of the sensor. Intuitively, when sensors are close by in space-time, they should give approximately the same reading. In other words, they have higher correlation. The extent to which they differ is a function of their distance from each other in space-time, as well as the miscalibration of their respective sensors. We should be able to infer this miscalibration and correct for it, yielding a more accurate reading, and therefore more accurate model.

There are two common types of sensor miscalibration that can be easily corrected: bias and gain error. Bias is an additive error, and gain error is a multiplicative error (See Figure 3).

Bias fits particularly well with our model, and gain error does not, so I investigate bias in this paper, and leave an investigation of gain error for future work.

Again, following from Rasmussen and Williams [4], to model bias, we first augment our x vector:

$$x = \begin{bmatrix} x_d \\ x_b \end{bmatrix},$$

where x_d is the previous x vector, and x_b is a vector of indicators which are 1 if the sample came for sensor i, and 0 otherwise.

Next we change our model slightly as follows:

$$g(x) = f(x) + h(x)^T\beta$$

where f is our original model, and $h(x) = x_b$. Now, if we assume a Gaussian prior over β:

$$\beta \sim \mathcal{N}(b, B)$$

then we can integrate out β and incorporate b and B into our model [3]:

$$g(x) \sim \mathcal{GP}(h(x)^T b, k(x, x') + h(x)^T B h(x'))$$

Plugging this into the prediction equations (2) and (3) above, we get

$$\bar{g}(X_*) = \bar{f}(X_*) + R^T\hat{\beta} \tag{4}$$

$$\operatorname{cov}(g_*) = \operatorname{cov}(f_*) + R^T(B^{-1} + HK_y^{-1}H^T)^{-1}R \tag{5}$$

where $K_y = K + \sigma_n^2 I$, $\hat{\beta} = (B^{-1} + HK_y^{-1}H^T)^{-1}(HK_y^{-1}y + B^{-1}b)$, and $R = H_* - HK_y^{-1}K_*$. $\bar{\beta}$ can be seen as trading off between the prior and the data. Since we have no apriori knowledge about each β, we can let the covariance of the prior on β go to infinity, and hence we have that $B^{-1} \to 0$. In that case we can simplify to

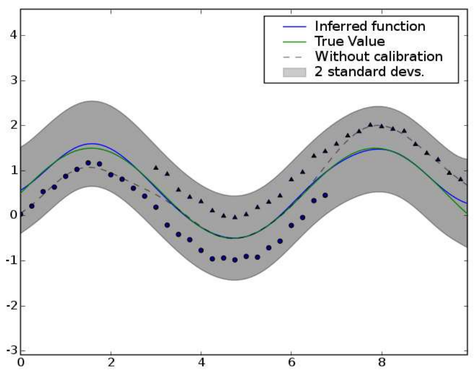
$$\operatorname{cov}(g_*) = \operatorname{cov}(f_*) + R^T(HK_y^{-1}H^T)^{-1}R \tag{6}$$

with $\hat{\beta} = (HK_y^{-1}H^T)^{-1}(HK_y^{-1}y)$. Thus b has no influence on our prediction, as we would expect from a diffuse prior.

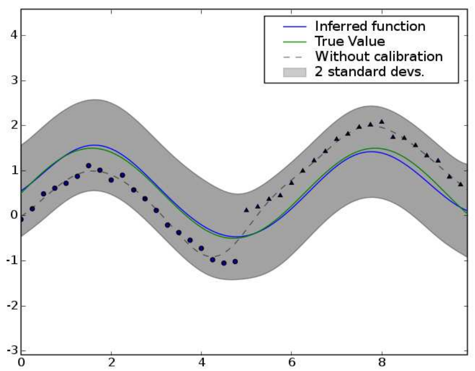
Assuming that the bias of the sensors are iid, then the maximum likelihood value of the "true" bias is the empirical mean of the inferred $\hat{\beta}$ values

$$\hat{\beta}_{ML} = \frac{1}{s}\sum_{i=1}^{s}\beta_i.$$

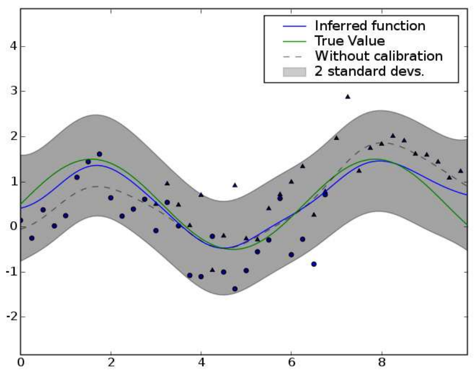
In Figure 3, we see our modified Gaussian process learning the function $y = sin(x) + \frac{1}{2}$ from the noisy observations of one sensor with bias $\frac{1}{2}$, and another sensor with bias $-x\frac{1}{2}$. The



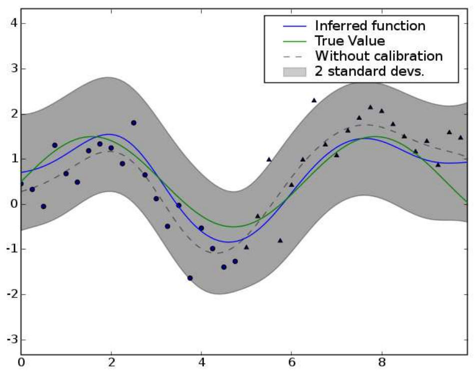
(a) $\sigma_p = 0.0625$ overlapping



(b) $\sigma_p = 0.0625$, non-overlapping

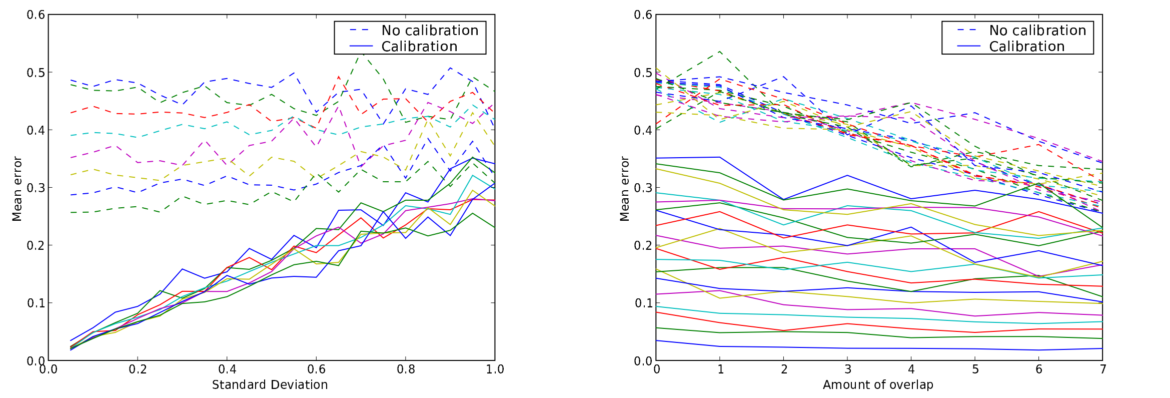


(c) $\sigma_p = 0.5$ overlapping



(d) $\sigma_p = 0.5$ non-overlapping

Figure 3: $y = sin(x) + \frac{1}{2}$ sampled from two sensors with different amounts of noise



(a) Mean error vs σ_n for various amounts of overlap (b) Mean error vs overlap for various values of σ_n

Figure 4: Inferred Gaussian process functions

standard Gaussian process fit is shown for reference. The modified Gaussian process is able to learn the mean value and avoid overfitting to the biased readings.

To investigate the impact of overlap between readings of the same location from different sensors on the accuracy of the inferred model, I generated ten data sets from each configuration of overlap between two sensors, as shown above in Figure 3, with values for overlap coming from the set $\{0, 1, \ldots, 8\}$, and σ_n from the set $\{0.05, 0.1, \ldots, 1.0\}$. The function was sampled in sufficient points so that the number of non-overlapping samples for each configuration was constant. I then calculated the mean error between the inferred function and the true value of the function $y = sin(x) + \frac{1}{2}$. I then took the mean for each configuration, and plotted it Figure 4.

Figure 4(a) show that for the calibrated model, the mean error increases with the standard deviation of the samples, as we would expect it to, but for the uncalibrated model, the mean error is swamped by the error caused by miscalibration. This means that the calibrated model does much better in places which are only sampled by a single sensor.

In Figure 4(b), we see that as the amount of overlap increases, the error remains relatively constant for the calibrated model for a given value of σ_n. This suggests that the effectiveness of the calibration only depends on a single, accurate pair of estimates in a nearby location in order to accurately infer the relative miscalibration of two sensors.

4 A kernel with finite support

As mentioned above, the radial basis kernel has infinite support, which makes the Gram matrix dense. A naive solution to this problem would be to truncate a Gaussian kernel after, say, 3 standard deviations. Unfortunately, this leads to non-PSD kernels, which can not be inverted, and thus can not be used with our algorithm.

Rasmussen and Wilson discuss a method developed by Wendland [5] for construcing PSD kernels with compact support using piecewise polynomials. One applicable example is the function

$$k(r) = \begin{cases} (1-r)^j & 0 \leq r \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

where $j = \lfloor \frac{D}{2} \rfloor + 2$ and D is the maximum number of dimensions for which the kernel is PSD. Obviously this can be scaled to any size.

This function seems to behave sufficiently similarly to a radial basis kernel, but this topic bears further investigation.

5 Conclusions and future work

5.1 Inferring gain error

Gain error is a very important source of error in many sensors, and so the techniques we utilize should be able to detect and correct these errors. This will most likely involve an iterative solution since the products of random variables are generally messy.

5.2 Modeling cyclical dependencies

This paper has ignore the temporal aspect of environmental sensing. The time dimension, however, will not behave in the same way that the spacial dimensions do. Many environmental characteristics, such as carbon monoxide (CO) levels, are highly cyclical. Specifically, CO levels will tend to be higher during the morning and evening rush hours, and lowest in the middle of the night. Similarly, they might tend to gradually increase over the course of the week and then decline over the weekend.

To model cyclical trends like this, Rasmussen and Williams suggest first mapping the cyclical dimension x to $u(x) = (\cos(x), \sin(x))$. Since

$$(\cos(x) - \cos(x'))^2 + (\sin(x) - \sin(x')) = 4 \sin^2 \left(\frac{x - x'}{2} \right),$$

then the squared exponential kernel in in u-spaces becomes

$$k(x, x') = \exp \left(- \frac{2 \sin^2 \left(\frac{x-x'}{2} \right)}{l^2} \right), \tag{7}$$

where l is the characteristic length scale of the kernel.

To model periodicity on multiple scales (say, daily and weekly), we can add parameters and take a linear combination of the (appropriately parametrized) trend kernel k_1, and the two periodic kernels k_2 and k_3:

$$k(x, x') = \theta_1 k_1(x, x') + \theta_2 k_2(x, x') + \theta_3 k_3(x, x') \tag{8}$$

The log likelihood of our trained model is given by:

$$\log p(y|X) = -\frac{1}{2} y^T (K + \sigma_n^2 I)^{-1} y - \frac{1}{2} \log |K + \sigma_n^2 I| - \frac{n}{2} \log 2\pi \tag{9}$$

Since we get the Cholesky decomposition of $K + \sigma_n^2 I$ in the process of inferring the Gaussian process mean, we can substitute $\log |K + \sigma_n^2 I| = 2 \sum_i \log L_{ii}$, where L is the Cholesky decomposition of $K + \sigma_n^2 I$. Doing so, we get

$$\log p(y|X) = -\frac{1}{2} y^T (K + \sigma_n^2 I)^{-1} y - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi. \tag{10}$$

We can then use the log likelihood to optimize the parameters of our model using an optimizer like the conjugate gradient optimizer.

I plan to explore this technique, and its applicability to real data as that data becomes available.

5.3 Bounding model complexity with epochs and approximate priors

Since data from sensors will stream in continuously, a method for bounding the model complexity will be necessary. One simple approach is to periodically mark an epoch, compress the model to a constant size representation, and then use that model as a prior for the next epoch. I plan to explore how different techniques for compressing the model impact the accuracy of the model over several epochs. I expect that the model will be robust to inaccurate approximations because mistakes will get swamped by the data over time, but this will need to be confirmed at least empirically, if not analytically.

5.4 Final comments

In this paper I have explored Gaussian processes as an integrated method for surface fitting and calibration of sensors. Gaussian processes provide concrete statistical interpretation of the generated model, and as such are an attractive component of complex models which utilize covariance information. The sensor calibration technique I have described appears to work well as long as sensors take a least one accurate reading close to one another. Significant work, however, remains to be done for the model I have developed to be usable on a large scale over a long period of time.

References

[1] Daratech reports gis revenues forecast to grow 8% to $1.75 billion in 2003; utilities and government increase spending, August 2003. http://www.directionsmag.com/article.php?article_id=403.

[2] David Culler, Deborah Estrin, and Mani Srivastava. Guest editors' introduction: Overview of sensor networks. *Computer*, August 2004.

[3] A O'hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society B*, pages 1–42.

[4] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[5] Holger Wendland. *Scattered data approximation*. Cambridge University Press, 2005.