Prediction and Fault Detection of Environmental Signals with Uncharacterised Faults

Michael A. Osborne
Engineering Science
University of Oxford
Oxford OX1 3PJ, UK
mosb@robots.ox.ac.uk

Roman Garnett
Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213, US
rgarnett@cs.cmu.edu

Kevin Swersky
Aquatic Informatics
Suite 1100, 570 Granville St
Vancouver, V6C 3P1, Canada
kswersky@cs.toronto.edu

Nando de Freitas
Computer Science
University of British Columbia
Vancouver, V6T 1Z4, Canada
nando@cs.ubc.ca

Abstract

Many signals of interest are corrupted by faults of an unknown type. We propose an approach that uses Gaussian processes and a general "fault bucket" to capture *a priori* uncharacterised faults, along with an approximate method for marginalising the potential faultiness of all observations. This gives rise to an efficient, flexible algorithm for the detection and automatic correction of faults. Our method is deployed in the domain of water monitoring and management, where it is able to solve several fault detection, correction, and prediction problems. The method works well despite the fact that the data is plagued with numerous difficulties, including missing observations, multiple discontinuities, nonlinearity and many unanticipated types of fault.

Introduction

Water sustainability is one of the most significant issues that humanity faces. The quality and availability of water directly affects the health and well-being of both human and natural environments. Water also has a massive economic impact, affecting industries such as agriculture, mining, power, forestry, and more. Population and industrial growth, along with climate change, are beginning to stress water supplies. Between 1994 and 1999, 26% of Canadian municipalities reported water shortages despite the fact that Canada contains 7% of the world's renewable freshwater supply (Environment Canada 2008).

One of the keys to effective water sustainability involves proper monitoring and analysis. Water is not distributed evenly in space and time, and being able to measure, predict, and respond to changes in the water supply will allow organizations to allocate supplies to where they are needed most. To this end, organizations such as Water Survey Canada (WSC) and the United States Geological Survey (USGS) have set up around 11,000 monitoring stations across North America (Wagner and US Geological Survey 2006), many of them reporting telemetry data in real-time. Analyzing this ever-increasing amount of data by hand is difficult, costly, and time-consuming; there are many opportunities for Machine Learning to automate this process in order to significantly improve the efficiency and effectiveness of current water-management systems.

In this paper, we attack the problem of fault detection, correction, and prediction in water monitoring signals. Here measurements are often corrupted in non-trivial ways by various intermittent faulty sensing and communication mechanisms, giving rise to outliers, telemetry spikes, missing data, drift, and multiple unanticipated exogenous disturbances (see Figure 1). Further, signals are not well-modelled by simple parametric approaches, such as linear or Markovian models. Despite the enormous importance of such monitoring, appropriate machine-learning techniques are yet to be deployed for this purpose. In particular, there is a clear need for flexible algorithms, able to cope with signals and faults of many different types without placing a significant model-building burden upon users. Such algorithms must also be able to run reliably in real-time on incoming data. These techniques will enable us to provide operators with high-level summaries for better decision support and, in the future, to increase the level of automation and efficiency in water-management systems.

The collection of literature on fault- (also known as novelty-, anomaly- and one-class-) detection is vast (Eciolaza et al. 2007; de Freitas, MacLeod, and Maltz 1996; Isermann 2005; Ding 2008; Markou and Singh 2003; Chandola, Banerjee, and Kumar 2009; Khan and Madden 2010; Dereszynski and Dietterich 2011). Unfortunately, the problems solved by most of these techniques are of very different character to our own, rendering such techniques inapplicable. Further, after much experimentation with those methods that are applicable (some of the results appear in our experiments section) it became clear that off-the-shelf techniques could not satisfy our requirements for reliable water monitoring. This was predominately due to excessively restrictive assumptions (e.g., that signals were linear, Markov or Gaussian), and/or a failure to produce reasonable uncertainty estimates. Green-tech areas, including environmental monitoring and energy-demand prediction, are still far from full automation; the provision of uncertainty estimates is necessary to allow human operators to make appropriate decisions. For this reason, we focus on developing probabilistic nonlinear models of the signal. In addition to providing posterior probabilities of observation faultiness, we are able to perform effective prediction for the latent process even in the

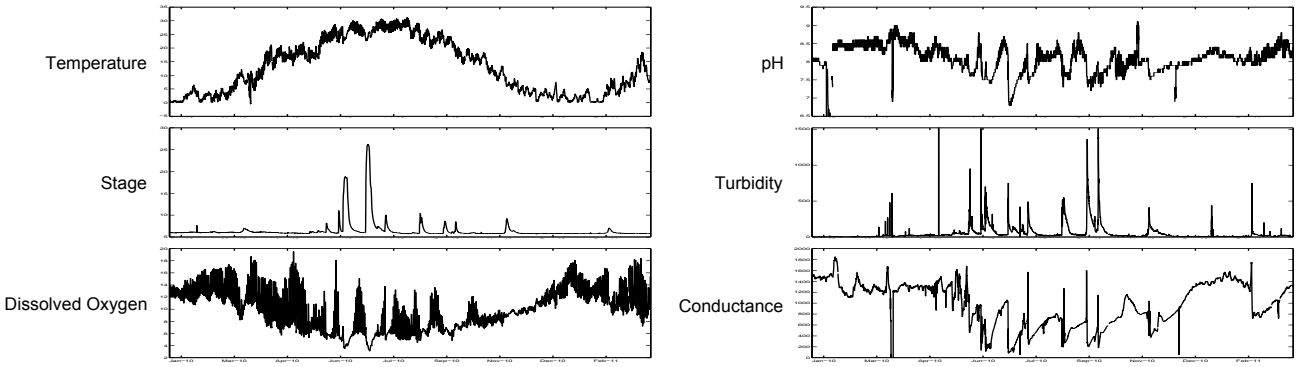

Figure 1: 16 months worth of data from six representative signals in water quality monitoring, which corresponds to approximately 11 000 measurements per series. These signals are highly nonlinear, demonstrating periodicity at different scales, intermittent pulses, and changes in dynamics. Not only do these signals exhibit a wide range of dynamics, but different signals of the same measurement type can also differ drastically if they are taken in different regions.

presence of faults.

Our proposed method will rely on Gaussian processes (GPs) due to their flexibility and widely demonstrated effectiveness. GPs have been used previously for fault detection in Eciolaza et al. (2007), but in a very different context, unsuitable for our problem. Previous work along similar lines has approached this problem by creating models that specify the anticipated potential fault types *a priori* (Garnett et al. 2010), but this is usually an unreasonable assumption in highly variable or poorly understood environments. In our proposed "fault bucket" approach, each point is assumed generated from either a nominal or generic faulty process; we do not require the specification of precise fault models. In this way, our model can simultaneously identify anomalies and robustly make predictions in the presence of sensor faults. The result is an efficient method for data-stream prediction that can manage a wide range of faults without requiring significant domain-specific knowledge.

Gaussian Processes

Gaussian processes provide a simple, flexible framework for performing Bayesian inference about functions (Rasmussen and Williams 2006). A Gaussian process is a distribution on the functions $f\colon \mathcal{X} \to \mathbb{R}$ (on an arbitrary domain $\mathcal{X}$) with the property that the distribution of the function values at a finite subset of points $F \subseteq \mathcal{X}$ are multivariate Gaussian distributed. A Gaussian process is completely defined by its first two moments: a mean function $\mu\colon \mathcal{X} \to \mathbb{R}$ and a symmetric positive semidefinite covariance function $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The mean function describes the overall trend of the function and is typically set to a constant for convenience. The covariance function describes how function values are correlated as a function of their locations in the domain, thereby encapsulating information about the overall shape and behavior of the signal. Many covariance functions are available to model a wide variety of anticipated signals.

Suppose we have chosen a Gaussian process prior distribution on the function $f\colon \mathcal{X} \to \mathbb{R}$, and a set of input points $\mathbf{x}$, the prior distribution on $\mathbf{f} = f(\mathbf{x})$ is

$$p(\mathbf{f}\,|\,\mathbf{x}, \theta) = \mathcal{N}\big(\mathbf{f}; \mu(\mathbf{x}; \theta), K(\mathbf{x}, \mathbf{x}; \theta)\big),$$

where $K(\mathbf{x}, \mathbf{x}; \theta)$ is the Gram matrix of the points $\mathbf{x}$, and θ is a vector containing any parameters required of μ and K, which form hyperparameters of the model.

Exact measurements of the latent function are typically not available. Let $y(x)$ represent the value of an observation of the signal at x and $f(x)$ represent the value of the unknown true latent signal at that point. When the observation mechanism is not expected to experience faults, the usual noise model used is

$$p(y\,|\,f, x, (\sigma^n)^2) = \mathcal{N}(y; f, (\sigma^n)^2), \tag{1}$$

which represents additive i.i.d. Gaussian observation noise with variance $(\sigma^n)^2$. Note that this model is inappropriate when sensors can experience faults, which corrupt the relationship between y and f.

With the observation model above, given a set of observations $\mathcal{D} = \big\{\big(x, y(x)\big)\big\} = (\mathbf{x}, \mathbf{y})$, the posterior distribution (itself given by a Gaussian Process) of $f_\star = f(x_\star)$ given these data is

$$p(f_\star\,|\,\mathbf{y}, \theta) = \mathcal{N}\big(f_\star; m(f_\star\,|\,\mathbf{y}, \theta), C(f_\star\,|\,\mathbf{y}, \theta)\big), \tag{2}$$

where the posterior mean and covariance are

$$m(f_\star\,|\,\mathbf{y}, \theta) = \mu(x_\star; \theta) + K(x_\star, \mathbf{x}; \theta)V^{-1}\big(\mathbf{y} - \mu(\mathbf{x}; \theta)\big)$$

$$C(f_\star\,|\,\mathbf{y}, \theta) = K(x_\star, x_\star; \theta) - K(x_\star, \mathbf{x}; \theta)V^{-1}K(\mathbf{x}, x_\star; \theta),$$

for $V = K(\mathbf{x}, \mathbf{x}; \theta) + (\sigma^n)^2\mathbf{I}$.

We now make some definitions for the sake of readability. Henceforth, we assume that our observations $\mathbf{y}$ have already been scaled by the subtraction of the prior mean $\mu(\mathbf{x}; \theta)$. We will also make use of the covariance matrix shorthand $K_{m,n} = K(\mathbf{x}_m, \mathbf{x}_n)$. Finally, for now, we'll drop the explicit dependence of our probabilities on the hyperparameters θ (it will be implicitly assumed that all quantities are conditioned on knowledge of them) and will return to them later. Similarly, we drop the dependence of our probabilities on the values of inputs x, which we assume are always known.

Fault Bucket

We propose an algorithm that is designed to deal with faults of many different, unspecified types. We use a sequential scheme, applicable for ordered data such as time series, partitioning the data available at any point into old and new halves. We then approximately marginalise the faultiness of old observations, storing and then updating our results for future use. This gives rise to an efficient and fast algorithm. In order to effect our scheme, we make four key approximations:

1. **Fault bucket:** Faulty observations are assumed to be generated from a Gaussian noise distribution with a very wide variance.
2. **Single-Gaussian marginal:** A mixture of Gaussians, weighted by the posterior probabilities of faultiness of old data, is approximated as a single moment-matched Gaussian.
3. **Old/new noise independence:** We assume that noise contributions are independent, and that the contributions for new data are independent of old observations.
4. **Affine precision:** The precision matrix over both old and new halves is assumed to be affine in the precision matrix over the old half.

Approximations 1 and 2 represent the state-of-the-art (Dereszynski and Dietterich 2011). However, using them alone will not give an algorithm that can scale to the real-time problems we consider. Our novel approximations 3-4 permit very fast, fault-tolerant inference. We will detail and justify these approximations further below, although the somewhat laborious mathematical details of the derivation of our algorithm are consigned to an appendix (Osborne et al. 2012) for brevity.

Our single, catch-all, "fault bucket" is expressed by approximation 1. It is built upon the expectation that points that are more likely to have been generated by noise with wide variance than under the normal predictive model of the GP can reasonably be assumed to be corrupted in some way, assuming we have a good understanding of the latent process. It is hoped that a very broad class of faults can be captured in this way. To formalise this idea, we choose an observation noise distribution to replace (1) that models the noise as independent but not identically distributed with separate variances for the non-fault and fault cases:

$$\begin{aligned} p(y \,|\, f, x, \neg\,\mathrm{fault}, (\sigma^n)^2) &= \mathcal{N}(y; f, (\sigma^n)^2) \\ p(y \,|\, f, x, \mathrm{fault}, (\sigma^f)^2) &= \mathcal{N}(y; f, (\sigma^f)^2), \end{aligned} \tag{3}$$

where $\mathrm{fault} \in \{0, 1\}$ is a binary indicator of whether the observation $y(x)$ was faulty and $\sigma^f > \sigma^n$ is the standard deviation around the mean of faulty measurements. The values of both σ^n and σ^f form hyperparameters of our model and are hence included in θ.

Of course, *a priori*, we do not know whether an observation will be faulty. Unfortunately, managing our uncertainty about the faultiness of all observations is a challenging task. With N observations, there are 2^N possible assignments of faultiness; it is infeasible to consider them all. Our solution is founded upon approximation 2. We approximately marginalise the faultiness of old observations, representing the mixture of different Gaussian predictions (each given by a different combination of faultiness) as a single Gaussian. This natural approach is similar to that taken in the related field of switching Kalman filters (Murphy 1998). We prefer this approximate marginalisation over faultiness to heuristics that would designate observations as either faulty or not—we acknowledge our uncertainty about faultiness.

More formally, define σ to be the (unknown) vector of all noise variances at observations $\mathbf{y}$. Because we have to sum over all possible values for these vectors, we will index the possible values of σ by i, each given by a different combination of faultiness over $\mathcal{D}$ (e.g. $\sigma_t^0 = \sigma^n$ for $\neg\,\mathrm{fault}$ and $\sigma_t^1 = \sigma^f$ for fault). Given all data $\mathcal{D}$, we need to marginalise over predictions indexed by i, as per

$$\begin{aligned} &p(f_\star \,|\, \mathbf{y}) \\ &= \sum_i p(\sigma^i \,|\, \mathbf{y}) p(f_\star \,|\, \mathbf{y}, \sigma^i) \\ &= \sum_i p(\sigma^i \,|\, \mathbf{y}) \mathcal{N}\big(f_\star; m(f_\star \,|\, \mathbf{y}, \sigma^i), C(f_\star \,|\, \mathbf{y}, \sigma^i)\big), \end{aligned} \tag{4}$$

the weighted sum of Gaussian predictions made using the different possible values for σ. By moment-matching, we collapse the weighted sum of all these predictions to a single Gaussian prediction.

In order to build a sequential algorithm, imagine that we have partitioned our observations $\mathcal{D}_{a,b}$ into a set of old observations $\mathcal{D}_a = (\mathbf{x}_a, \mathbf{y}_a)$ and a set of new observations $\mathcal{D}_b = (\mathbf{x}_b, \mathbf{y}_b)$. We will index the possible values of σ_a by i and the values of σ_b similarly by j. We now define the covariance matrices over our data $V_a^i = K_{a,a} + \operatorname{diag} \sigma_a^i$, $V_b^j = K_{b,b} + \operatorname{diag} \sigma_b^j$ and $V_{a,b}^{i,j} = K_{\{a,b\},\{a,b\}} + \operatorname{diag}\{\sigma_a^i, \sigma_b^j\}$, where $\operatorname{diag} \sigma$ is the diagonal matrix with diagonal σ. From (2), we know that our predictions have relatively simple dependence upon the inverse of V, the precision. To approximate (4) as a single Gaussian, a simple calculation reveals that we require[1] the expected values of V^{-1} and $V^{-1} y y^\mathsf{T} V^{-1}$, expectations with respect to $p(\sigma^i \mid \mathbf{y})$. We'll refer to the set of those two expected values (which are matrices) as the *marginal set*, which, given $\mathcal{D}_a$, we denote as M_a. Our approach relies upon storing M_a, and then performing simple updates in order to arrive at $M_{a,b}$, the marginal set given $\mathcal{D}_{a,b}$. Hence our approximate marginalisation from previous time steps can be efficiently used to determine an approximate marginalisation for the current time step.

This approach firstly relies upon approximation 3; we assume that faults will not persist longer than $|\mathcal{D}_b|$. To be precise, we assume

$$p(\sigma_{a,b}^{i,j}, \mathbf{y}_{a,b}) \simeq p(\sigma_a^i)\, p(\mathbf{y}_a \,|\, \sigma_a^i)\, p(\sigma_b^j)\, p(\mathbf{y}_b \,|\, \sigma_{a,b}^{i,j}, \mathbf{y}_a) \tag{5}$$

This simplifies the expectations required to evaluate $M_{a,b}$. A further simplification is afforded by approximation 4, in which we assume that $(V_{a,b}^{i,j})^{-1}$ is effectively affine in

[1] Actually, with some rearrangement, we can avoid explicitly computing (unstable) matrix inverses and instead work with Cholesky factors to solve the required linear equations.

$(V_a^i)^{-1}$. This is true if given $\mathcal{D}_b$, it is impossible to accurately predict $\mathcal{D}_a$. This might be the case if $\mathcal{D}_a$ represents a lot of information relative to $\mathcal{D}_b$ (if, for example, $\mathcal{D}_a$ is our entire history of observations where $\mathcal{D}_b$ is simply the most recent observation), or if $\mathcal{D}_b$ and $\mathcal{D}_a$ are simply not very well correlated. Together, approximations 3 and 4 allow us the efficient updates to the marginal set M required to build an online algorithm.

If we now receive further data $\mathcal{D}_c$, our existing data is simply treated as old data ($a \leftarrow \{a, b\}$, $b \leftarrow c$), and another iteration of our algorithm performed. This requires the efficient updating of the marginal set stored from previous iterations. At each iteration, our algorithm is able to return the predictions for the latent variable $p(f_\star \,|\, \mathbf{y}_{a,b})$ and the posterior probability of an observation's faultiness $p(\sigma_b \,|\, \mathbf{y}_{a,b})$. We can also return the marginal likelihood of our model's hyperparameters, $p(\mathbf{y}_{a,b})$, useful if we want to learn such hyperparameters from data.

Discussion

We return to the management of our hyperparameters θ. Unfortunately, analytically marginalising θ is impossible. Most of the hyperparameters of our model can be set by optimising their likelihood on a large training set, giving a likelihood close to a delta function. This is not true of the hyperparameters σ^n and σ^f, due to exactly the same problematic sums discussed earlier. Instead, we marginalise these hyperparameters online using Bayesian Monte Carlo (Rasmussen and Ghahramani 2003; Osborne et al. 2008), taking a fixed set of samples in their values and using the hyperparameter likelihoods $p(\mathbf{y}_{a,b})$ to construct weights over them. Essentially, we proceed as described above independently in parallel for each sample, and combine the predictions from each in a final weighted mixture for prediction. Note that we can use a similar procedure (Garnett et al. 2010) to determine the full posterior distributions of σ^n and σ^f, if desired. It would be desirable to use non-fixed samples, but, unfortunately, this would require reconstructing our full covariance matrix from scratch each time a sample is moved.

Our proposal can be extended in several ways. First, we may wish to sum over more than one fault variance, useful if observations are prone to faultiness in more than one mode. If, instead of summing over a small number of known variances, we wished to marginalise with respect to a density over noise variance, we can simply replace the sums over i and j with appropriate integrals. Obviously this will only be analytically possible if the posteriors for σ_a^i take appropriate, simple forms. In such a way our algorithm might tackle the general problem of heteroscedasticity.

Our proposed algorithm steps through our data one at a time, so that $\mathcal{D}_b$ always contains only a single observation. With approximation 3, this means that our algorithm is not expecting faults to last more than a single observation. The results that follow, however, will show that we can nonetheless manage sustained faults. It would also be possible to step in larger chunks, evaluating larger sums. Although more computationally demanding, this might be expected to improve results. It would also allow us to consider non-diagonal noise contributions.

We have so far not specified our prior for faultiness (as expressed by $p(\sigma_a)$ and $p(\sigma_b)$). Within this paper, we consider exclusively a time-independent probability π of faultiness, but our framework does not necessarily require this to be so.

In some contexts it might be useful to perform inference about the fault contribution, rather than the signal of interest. To do so, we merely switch the roles of the fault and non-fault contributions. Note that, using our full posteriors for faultiness, we can also trivially use Bayesian decision theory to make hard decisions as required.

Results

We test the effectiveness of the fault bucket algorithm on several time-series that are indicative of problems found in environmental monitoring. In particular, we test on water-level readings; such data are often characterised by complex dynamics and will therefore provide a good indicator of our algorithm's performance on real-world tasks. We aim to improve upon the simple, human-supervised approaches to fault detection used in this field (Wagner and US Geological Survey 2006). For a quantitative assessment, we used two semi-synthetic datasets where a typical fault has been injected into clean sensor data. We then analyzed qualitative performance on two real data sets with actual faults. All measurements (other than for pH) are given in meters, with samples spaced in increments of approximately 30 minutes.

Our first synthetic example, a bias fault, concerns a simple sensor error where measurements are temporarily adjusted by a constant offset, but otherwise remain accurate. This could happen if the sensor undergoes physical trauma which results in a loss of calibration. The next dataset contains a synthetic anomaly where the water level rises quickly, but smoothly, before returning back to normal. This would be indicative of a genuine environmental event such as a flash flood. Both synthetic datasets represented sustained faults, of length 335 and 161 observations respectively. Our first real dataset deals with pH measurements from the United States Geological Survey, a common indicator of water quality. In this series, a clear sensor fault can be seen in which the observations undergo a sudden, sustained decrease in value. The next (real) dataset contains a fault type called "painting," an error that occurs when ice builds on a sensor, obscuring some of the readings. It is characterised by frequent sensor spikes interlaced with the original, and still accurate, signal.

We implemented the algorithm described in Section in MATLAB to address the task of 1-step-lookahead time-series prediction. A sliding window of size 100 was used to predict the value of the next observation. Each dataset was recentered so that a zero prior mean function was appropriate, and the functions were all modeled using a Matérn covariance with parameter $\nu = {}^5\!/_2$ (Rasmussen and Williams 2006). The hyperparameters for this covariance, including the normal observation noise σ^n, were learned using training data similar to but disjoint from the test datasets. The unknown fault noise σ^f was marginalised using Bayesian Monte Carlo, with a parsimonious 7 samples used. The prior probability of an observation being faulty, π, was set to a constant value of 1% throughout.

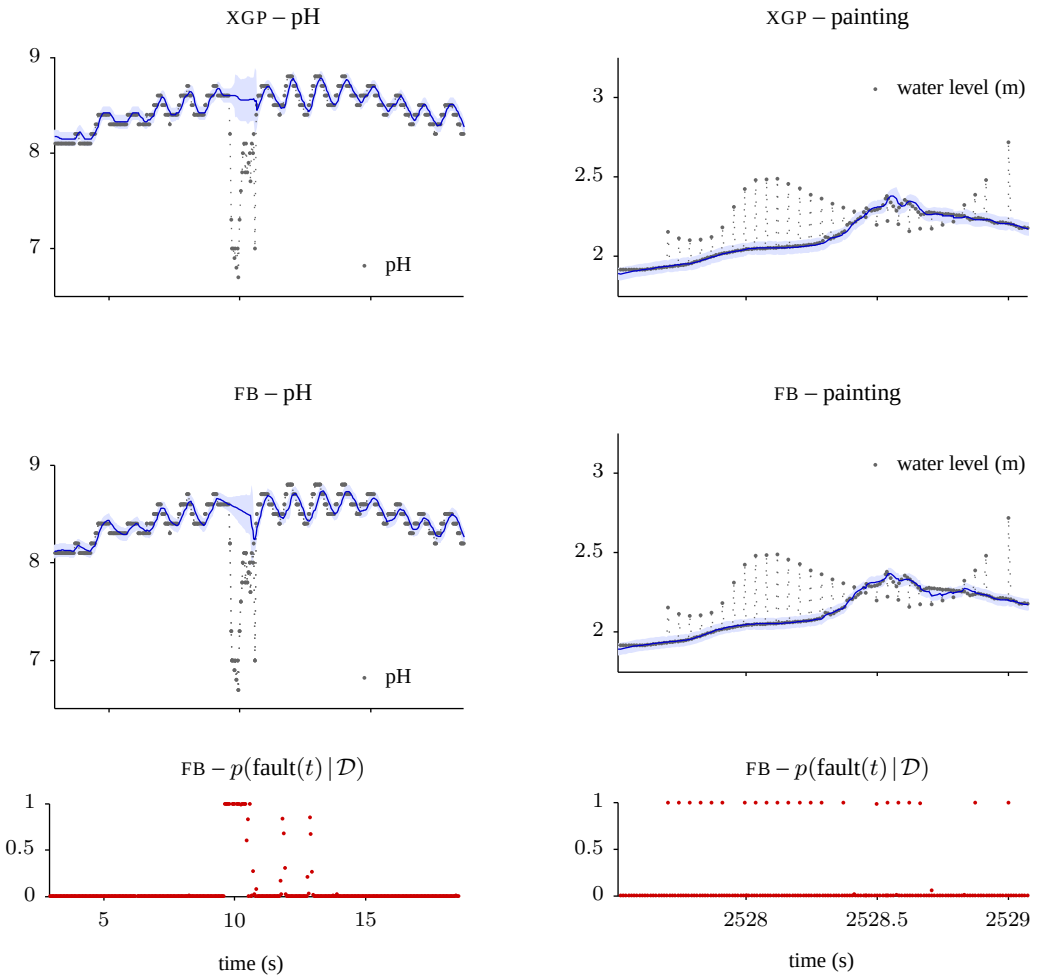

Figure 2: Mean and $\pm 3\sigma$ standard-deviation bounds for the predictions of the exhaustive (XGP) and fault-bucket (FB) algorithms on the pH and painting datasets. For the fault-bucket algorithm, the posterior faultiness of each observation is also shown underneath the predictions. Note that each column shares the same x-axis.

We tested against a number of different methods in order to establish the efficacy of the fault bucket algorithm. We focus particularly on other GP-based approaches, in order that our novel modifications of a GP model can be best evaluated. All GP-based approaches used the same hyperparameters employed by our algorithm. The training set used to learn those hyperparameters was also supplied to other methods for their respective model learning phases. Several methods identify a new observation y as a fault if

$$\left|y - m(y\,|\,\mathbf{y})\right| > 3\sigma^T\,, \tag{6}$$

where $m(y\,|\,\mathbf{y})$ is the method's *a priori* prediction for y, and σ^T is the noise standard deviation on the faultless training set. Of course, methods using (6) or similar can not provide the posterior probability of a point's faultiness, as our algorithm can. Methods tested include:

XGP: A GP in which we exhaustively search over the faultiness of the last 10 points, and approximate the noise variance of all previous points in the window as having the value $(\sigma^f)^2 p(\text{fault}\,|\,\mathbf{y}) + (\sigma^n)^2 p(\neg\,\text{fault}\,|\,\mathbf{y})$, fixed at the time the point was observed (when data $\mathcal{D}$ was available). Clearly, this method is very much more computationally expensive than the fault bucket algorithm (roughly 2^9 times more), but offers a useful way to quantify the influence of approximations 2–4.

EPGP: A GP with our observation likelihood (3) and the posterior determined by expectation propagation (Minka 2001). Note that expectation propagation will not readily give us a posterior probability of faultiness; for the purposes of explicitly identifying faults, we use (6).

TGP: A GP in which a point was flagged as a fault using (6); if faulty, a point was treated as having noise variance $(\sigma^f)^2$.

STGP: A GP with student-t likelihood with four degrees of freedom (found to optimise performance); the posterior was determined using a Laplace approximation as per Vanhatalo, Jylänki, and Vehtari (2009). To identify faults, (6) was used.

MLH: The most likely heteroscedastic GP (Kersting et al. 2007).

Table 1: Quantitative comparison of different algorithms on the synthetic datasets. For each dataset, we show the mean squared error (MSE), the log likelihood of the true data ($\log p(\mathbf{y} \mid \mathbf{x})$), and the true-positive and false-positive rates of detection for faulty points (TPR and FPR), respectively, with all methods permitted a 'burn-in' period of 50 points. The best value for each set of results is highlighted in bold.

	Bias dataset				"Flash-flood" dataset			
Method	MSE	$\log p(\mathbf{y} \mid \mathbf{x})$	TPR	FPR	MSE	$\log p(\mathbf{y} \mid \mathbf{x})$	TPR	FPR
FB	**0.024**	334	**0.997**	0.031	0.069	-5.77×10^3	**0.829**	0.016
XGP	0.037	**439**	0.982	0.022	**0.042**	$\mathbf{-1.52 \times 10^3}$	0.805	0.012
EPGP	0.879	-5.06×10^3	0.009	0.025	2.179	-2.14×10^4	0.000	**0.000**
TGP	0.033	278	**0.997**	0.031	0.075	-8.29×10^3	**0.829**	0.083
STGP	0.189	-604	0.994	0.255	0.249	-1.01×10^5	0.787	0.140
MLH	0.940	-5.43×10^7	0.065	0.031	2.369	-2.27×10^7	0.045	0.262
EKF	0.060	-1.26×10^4	0.551	0.258	0.613	-1.81×10^4	0.169	0.768
SKF	0.101	-1.04×10^4	**0.997**	**0.000**	0.162	-3.83×10^4	0.805	0.004

EKF: An autoregressive neural net trained with the extended Kalman filter to capture nonstationarity. Again, (6) was used to identify and discard faulty data.

SKF: A switching Kalman filter (Murphy 1998), which switched between the non-faulty and faulty observation models in (3). The model was trained by solving the Yule-Walker equations; the best model order found was three.

Note that for EPGP, STGP and MLH, we perform retrospective prediction (so that all data is available to make predictions about even the first predictant), as these methods are usually used. Clearly this allows these approaches an unfair predictive advantage relative to sequential methods. Note also that the multiple passes over the data effected by approximation schemes such as expectation propagation cannot be readily applied to the sequential problem without requiring a great deal of expensive computation. For N observations, the computational cost of expectation propagation is $\mathcal{O}(N^3)$, with a large constant of proportionality, rendering it impractical for real-time problems. With efficient Cholesky factor updates, our scaling is $\mathcal{O}(N^2)$. For this reason, we did not consider even more demanding, non-sequential, expectation propagation approaches such as the twinned GP (Naish-Guzman and Holden 2008). Note also that these approaches do not provide posterior probabilities of faultiness, as our method is able to.

Figures 2 show the performance of the fault-bucket algorithm and the exhaustive alternative on the two real datasets. The fault-bucket algorithm did an excellent job of identifying faults when they occur, and made excellent predictions, even for the sustained fault in the pH dataset.

Table 1 displays quantitative measures of performance for the various algorithms on the synthetic datasets. In addition to superior predictive performance, our detection rates for the faulty points are generally excellent. Note that approaches that provided comparable fault-detection rates, like the TGP and SKF, perform significantly poorer prediction. The results reveal that approximations 2–4 do not result in significant loss of performance on real data relative to exhaustive search. Our naïve approach to faults may, of course, suffer relative to better-informed models, but its probabilistic estimates of faultiness provide a human operator with an indication as to whether more sophisticated analysis is necessary. We believe our method will augment the toolbox of approaches to fault detection.

Conclusion

We have proposed a novel algorithm, the "fault bucket," for managing time-series data corrupted by faults of type unknown ahead of time. Our chief contribution is a sequential algorithm for marginalising the faultiness of observations in a GP framework, allowing for fast, effective prediction in the presence of unknown faults. Unlike most robust regression approaches (such as those using student-t likelihoods), we can also compute the posterior probability of faultiness. This capacity is crucial to its utility for the domain, serving as a means of alarming a human operator to the possible need for corrective action.

As to future work, addressing multivariate signals is of great interest. Unfortunately, this extension is not trivial and would itself require additional approximations.

Acknowledgement

All datasets have been provided by Aquatic Informatics. The data may be obtained by contacting Touraj Farahmand at `tourajf@aquaticinformatics.com`. M.A.O. was funded by the ORCHID project (`http://www.orchid.ac.uk/`).

References

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. ACM *Comput. Surv.* 41:15:1–15:58.

de Freitas, N.; MacLeod, I. M.; and Maltz, J. S. 1996. Neural networks for pneumatic actuator fault detection. *Transactions of the South African Institute of Electrical Engineers* 90:28–34.

Dereszynski, E., and Dietterich, T. G. 2011. Spatiotemporal models for anomaly detection in dynamic environmental monitoring campaigns. *ACM Transactions on Sensor Networks*.

Ding, S. X. 2008. *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer, first edition.

Eciolaza, L.; Alkarouri, M.; Lawrence, N. D.; Kadirkamanathan, V.; and Fleming, P. J. 2007. Gaussian Process Latent Variable Models for Fault Detection. In IEEE *Symposium on Computational Intelligence and Data Mining*, 287—292.

Environment Canada. 2008. Threats to Water Availability in Canada.

Garnett, R.; Osborne, M. A.; Reece, S.; Rogers, A.; and Roberts, S. J. 2010. Sequential Bayesian Prediction in the Presence of Changepoints and Faults. *The Computer Journal* 53.

Isermann, R. 2005. Model-based fault-detection and diagnosis – status and applications. *Annual Reviews in Control* 29(1):71–85.

Kersting, K.; Plagemann, C.; Pfaff, P.; and Burgard, W. 2007. Most likely heteroscedastic Gaussian process regression. In *Proceedings of the 24th international conference on Machine learning*, 393–400. ACM.

Khan, S., and Madden, M. 2010. A survey of recent trends in one class classification. In Coyle, L., and Freyne, J., eds., *Artificial Intelligence and Cognitive Science*, volume 6206 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 188–197.

Markou, M., and Singh, S. 2003. Novelty detection: a review – Part 1: Statistical approaches. *Signal Processing* 83(12):2481–2497.

Minka, T. 2001. Expectation propagation for approximate bayesian inference. In *Uncertainty in Artificial Intelligence*, volume 17, 362–369. Citeseer.

Murphy, K. 1998. Switching Kalman filters. Technical report, Compaq Cambridge Research Lab Tech Report 98-10.

Naish-Guzman, A., and Holden, S. 2008. Robust regression with twinned Gaussian processes. *Advances in Neural Information Processing Systems* 20:1065–1072.

Osborne, M. A.; Rogers, A.; Ramchurn, S.; Roberts, S. J.; and Jennings, N. R. 2008. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *International Conference on Information Processing in Sensor Networks (IPSN 2008)*, 109–120.

Osborne, M. A.; Garnett, R.; Swersky, K.; and de Freitas, N. 2012. Prediction and fault detection of environmental signals with uncharacterised faults: Appendix. available at: `http://www.robots.ox.ac.uk/~mosb/papers/fault_bucket_appendix.pdf`.

Rasmussen, C. E., and Ghahramani, Z. 2003. Bayesian Monte Carlo. In Becker, S., and Obermayer, K., eds., *Advances in Neural Information Processing Systems*, volume 15. Cambridge, MA: MIT Press.

Rasmussen, C. E., and Williams, C. K. I. 2006. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, MA, USA: The MIT Press.

Vanhatalo, J.; Jylänki, P.; and Vehtari, A. 2009. Gaussian process regression with student-t likelihood. In *Advances in Neural Information Processing Systems*, volume 22, 1910–1918.

Wagner, R. J., and US Geological Survey. 2006. *Guidelines and standard procedures for continuous water-quality monitors: Station operation, record computation, and data reporting*. US Department of the Interior, US Geological Survey.