A "Fault Bucket" for Time-Series Prediction with Uncharacterized Faulty Observations

Abstract

We provide a proposal for performing both online prediction and retrospective inference of signals from observations that are potentially rendered less informative than normal due to a faulty observation mechanism. The proposed model uses Gaussian processes and a general "fault bucket" for *a priori* uncharacterized faults, along with an approximate method for marginalizing the potential faultiness of all observations. This gives rise to an efficient, flexible algorithm. We demonstrate our method's relevance to problems drawn from environmental-monitoring applications.

1. Introduction

We consider the problem of inferring a signal $y\colon \mathbb{R} \to \mathbb{R}$ from noisy measurements of it. Typical algorithms for this purpose often assume that the data is linear, i.i.d., Markovian, or Gaussian. In some cases, knowledge about the problem can enable the explicit specification of parametrized nonlinear models. This approach, however, is problematic for two reasons. First, estimating the model parameters is often difficult. More crucially, collected observations in real-world applications are often corrupted in non-trivial ways due to, for example, faulty sensing mechanisms. Such effects often inhibit effective environmental monitoring, where data can be corrupted in unknown ways, rendering the *a priori* construction of parametric model impossible. The motivating example for this paper is the fast-growing field of water-quality monitoring (Wagner & US Geological Survey, 2006). Despite the enormous importance of such monitoring, there is a lack of sound machine-learning solutions for this problem. Here, we hope to release some of this data to the public domain (following the reviewing process in order to satisfy anonymous reviewing) and to present new techniques to meet the demands of the field.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Our proposed method for predicting signals with faulty observation mechanisms will rely on Gaussian processes (GPs) to perform inference about the underlying latent function. Previous work has approached this problem by creating observation models that specify the anticipated potential fault types *a priori* (Garnett et al., 2010), but this might be an unreasonable assumption in highly variable or poorly understood environments. Here we suggest the use of a catch-all "fault bucket," which can identify and treat appropriately data readings suspected of being corrupted in some way. The result is a method for data-stream prediction that can manage a wide range of faults without requiring significant domain-specific knowledge.

The collection of literature on similar topics is vast, under labels such as fault detection (de Freitas et al., 1996; Eciolaza et al., 2007; Isermann, 2005; Ding, 2008), novelty detection (Markou & Singh, 2003), anomaly detection (Chandola et al., 2009), or one-class classification (Khan & Madden, 2010). Despite this, most of the techniques are too simple (e.g. linear or Gaussian) or fail to produce good uncertainty estimates. Additionally, many problems in anomaly detection and one-class classification are of a very different nature and, therefore, the techniques developed there not immediately applicable to our domain of interest. Uncertainty estimates are key in order to provide the user of a system with reliable monitoring signals. Green-tech areas, including environmental monitoring and energy-demand prediction, are still far from full automation. Currently, the most important requirement of such systems is to provide the user with signals that he or she can use to reach a decision, making the uncertainty of predictions of the utmost importance. For this reason, we focus on developing GP-based techniques to build probabilistic nonlinear models of the signal and thereby deliver reliable reports. In addition to providing posterior probabilities of observation faultiness, we are able to perform effective prediction for the latent process even in the presence of faults.

GPs have been used previously for fault detection (Eciolaza et al., 2007). However, the nature of the data and setup in that domain is different from ours, forcing

us to develop new GP-based techniques for detection and prediction in the presence of faults.

2. Gaussian Processes

Gaussian processes provide a simple, flexible framework for performing Bayesian inference about functions (Rasmussen & Williams, 2006). A Gaussian process is a distribution on the functions $y\colon \mathcal{X} \to \mathbb{R}$ (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 $y\colon \mathcal{X} \to \mathbb{R}$, and a set of input points $\mathbf{x}$, the prior distribution on $\mathbf{y} \triangleq y(\mathbf{x})$ is

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

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 $z(x)$ represent the value of an observation of the signal at x and $y(x)$ 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

$$p(z \mid y, x, \sigma_n^2) \triangleq \mathcal{N}(z; y, \sigma_n^2), \tag{1}$$

which represents additive i.i.d. Gaussian observation noise with variance σ_n^2. Note that this model is inappropriate when sensors can experience faults, which complicate the relationship between z and y.

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

$$p(y_\star \mid \mathcal{D}, \theta) = \mathcal{N}\big(y_\star; m(y_\star \mid \mathcal{D}, \theta), C(y_\star \mid \mathcal{D}, \theta)\big),$$

where the posterior mean and covariance are

$$\begin{aligned} m(y_\star \mid \mathcal{D}, \theta) &\triangleq \mu(x_\star;\theta) + \\ &\quad + K(x_\star, \mathbf{x};\theta)\big(K(\mathbf{x},\mathbf{x};\theta) + \sigma_n^2\mathbf{I}\big)^{-1}\big(\mathbf{z} - \mu(\mathbf{x};\theta)\big) \\ C(y_\star \mid \mathcal{D}, \theta) &\triangleq K(x_\star, x_\star;\theta) - \\ &\quad - K(x_\star, \mathbf{x};\theta)\big(K(\mathbf{x},\mathbf{x};\theta) + \sigma_n^2\mathbf{I}\big)^{-1}K(\mathbf{x}, x_\star;\theta). \end{aligned}$$

We now make some definitions for the sake of readability. Henceforth, we assume that our observations $\mathbf{z}$ 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 $K_{m,n} \triangleq K(\mathbf{x}_m, \mathbf{x}_n)$. 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.

3. Fault Bucket

Rather than specifying explicit parameters for every possible fault type, we propose a single catch-all "fault bucket" that can identify and treat appropriately measurements that are suspected of being faulty. The basic idea is to model faulty observations as being generated from a Gaussian distribution with a very wide variance; points that are more likely under this model than under the normal predictive model of the Gaussian process 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 formalize this idea, we choose an observation noise distribution to replace that in (1) that models the noise as independent but not identically distributed with separate variances for the non-fault and fault cases.

$$\begin{aligned} p(z \mid y, x, \neg\,\text{fault}, \sigma_n^2) &\triangleq \mathcal{N}(z; y, \sigma_n^2) \\ p(z \mid y, x, \text{fault}, \sigma_f^2) &\triangleq \mathcal{N}(z; y, \sigma_f^2), \end{aligned}$$

where $\text{fault} \in \{0, 1\}$ is a binary indicator of whether the observation $z(x)$ 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 any given observation will be faulty. Unfortunately, managing our uncertainty about the "faultiness" of all available observations is a challenging task. With N observations available, there are 2^N possible assignments of faultiness. It quickly becomes computationally infeasible to marginalize over all these possible values.

Instead, we propose a sequential approach, applicable for ordered data such as time series. For time series, the value to be predicted $y_\star$ typically lies in the future, and less-recent observations are typically less pertinent for this task than more-recent ones. The intuition behind our approach is to, at a given time step, "merge" our current model with the uncertainty present in the inferred faultiness of the most recent observation. In effect, we represent each observation as having a known variance lying between between σ_n^2 and σ_f^2. The more likely an observation's faultiness, the closer its assigned variance will be to the (large) fault variance and the less relevant it will become for inference about the latent process. This approximate observation is then used for future predictions; we need never consider the full sum over all observations. Nonetheless, this approximate marginalization over faultiness is preferable to heuristics that would designate all observations as either faulty or not; our method acknowledges the uncertainty that may exist in our belief about faultiness.

More formally, imagine that we have partitioned our observations $\mathcal{D}_{a,b}$ into a set of old observations $\mathcal{D}_a \triangleq (\mathbf{x}_a, \mathbf{z}_a)$ and a set of newer observations $\mathcal{D}_b \triangleq (\mathbf{x}_b, \mathbf{z}_b)$. Define σ_a to be the (unknown) vector of all noise variances at observations $\mathbf{z}_a$, and define σ_b similarly. Because we have to sum over all possible values for these vectors, we will index the possible values of σ_a by i (each given by a different combination of faultinesses over $\mathcal{D}_a$) and the values of σ_b similarly by j. We now define the covariances

$$\begin{aligned} V_a^i &\triangleq K_{a,a} + \operatorname{diag} \sigma_a^i, \\ V_b^j &\triangleq K_{a,a} + \operatorname{diag} \sigma_a^i, \\ V_{a,b}^{i,j} &\triangleq K_{\{a,b\},\{a,b\}} + \operatorname{diag}\{\sigma_a^i, \sigma_b^j\}, \end{aligned}$$

where $\operatorname{diag} \sigma$ is the diagonal matrix with diagonal σ.

To initialize our algorithm, imagine that a identifies a small set of data, such that we can readily compute

$$p(\mathbf{z}_a) = \sum_i p(\mathbf{z}_a | \sigma_a^i) p(\sigma_a^i) = \sum_i \mathcal{N}(\mathbf{z}_a; 0, V_a^i) p(\sigma_a^i) \quad (2)$$

(which is the likelihood of our hyperparameters), so

$$p(\sigma_a | \mathcal{D}_a) = \frac{p(\mathbf{z}_a | \sigma_a) p(\sigma_a)}{p(\mathbf{z}_a)} = \frac{\mathcal{N}(\mathbf{z}_a; 0, V_a) p(\sigma_a)}{p(\mathbf{z}_a)}.$$

This distribution also specifies the probability of our observations $\mathcal{D}_a$ being faulty; for a single observation $\mathcal{D}_a$, $p\big(\text{fault}(\mathcal{D}_a) \mid \mathcal{D}_a\big) = p(\sigma_a = \sigma_f \mid \mathcal{D}_a)$.

If we were to perform predictions for some $y_\star$ using $\mathcal{D}_a$ alone, we would need to evaluate

$$\begin{aligned} p(y_\star \mid \mathcal{D}_a) &= \sum_i p(\sigma_a^i \mid \mathcal{D}_a) p(y_\star \mid \mathcal{D}_a, \sigma_a^i) \\ &= \sum_i p(\sigma_a^i \mid \mathcal{D}_a) \mathcal{N}\big(y_\star; m(y_\star \mid \mathcal{D}_a, \sigma_a^i), C(y_\star \mid \mathcal{D}_a, \sigma_a^i)\big), \end{aligned}$$

the weighted sum of Gaussian predictions made using the different possible values for σ_a. Now, we will perform the "merging" that we discussed earlier, by approximating this sum of Gaussians as a moment-matched single Gaussian. It is our hope that our predictions for $y_\star$ are not so sensitive to the noise in our observations that all the Gaussians in this sum become dramatically different. In any case, the quality of this approximation will improve over time—if $y_\star$ is far removed from our old data $\mathcal{D}_a$, then our predictions really will not be very sensitive to σ_a. So, we take

$$p(y_\star \mid \mathcal{D}_a) \simeq \mathcal{N}\big(y_\star; \tilde{m}(y_\star \mid \mathcal{D}_a), \tilde{C}(y_\star \mid \mathcal{D}_a)\big),$$

where we have

$$\tilde{m}(y_\star \mid \mathcal{D}_a) \triangleq K_{\star,a} \tilde{V}_a^{-1} \mathbf{z}_a, \quad (3)$$

$$\begin{aligned} \tilde{C}(y_\star \mid \mathcal{D}_a) \triangleq\ & K_{\star,\star} - K_{\star,a}(\tilde{V}_a^{-1} - \tilde{W}_a^{-1}) K_{a,\star} \\ & - \tilde{m}(y_\star \mid \mathcal{D}_{a,b})^2, \end{aligned} \quad (4)$$

where

$$\begin{aligned} \tilde{V}_a^{-1} &\triangleq \sum_i p(\sigma_a^i \mid \mathcal{D}_a) (V_a^i)^{-1}, \\ \tilde{W}_a^{-1} &\triangleq \sum_i p(\sigma_a^i \mid \mathcal{D}_a) (V_a^i)^{-1} \mathbf{z}_a \mathbf{z}_a^\mathsf{T} (V_a^i)^{-1}. \end{aligned} \quad (5)$$

Note that for $\tilde{W}_a$, explicitly computing (unstable) matrix inverses can be avoided by solving the appropriate linear equations using Cholesky factors. For $\tilde{V}_a$, we can rewrite $(A^{-1} + B^{-1})^{-1} = A(A+B)^{-1}B$. If $i \in \{0, 1\}$ (as it would be if a identified a single observation which could be either faulty or not),

$$\tilde{V}_a = V_a^0 \big(p(\sigma_a^1 \mid \mathcal{D}_a) V_a^0 + p(\sigma_a^0 \mid \mathcal{D}_a) V_a^1\big)^{-1} V_a^1. \quad (6)$$

If i takes more than two values, we can simply iterate using the same technique. Having used (6) to compute $\tilde{V}_a$, we can then calculate

$$\begin{aligned} \tilde{R}_a &\triangleq \operatorname{chol} \tilde{V}_a \\ \tilde{T}_a &\triangleq \operatorname{chol}(\tilde{R}_a)^{-1} \mathbf{z}_a \end{aligned}$$

and use them, along with (5), to efficiently determine (3) and (4).

Now, with these calculations performed, we can imagine receiving further data $\mathcal{D}_b$. Our predictions are now

$$p(y_\star \mid \mathcal{D}_{a,b}) = \sum_i \sum_j p(y_\star \mid \mathcal{D}_{a,b}, \sigma_{a,b}^{i,j}) p(\sigma_{a,b}^{i,j} \mid \mathcal{D}_{a,b}).$$

The full sums here can easily become too large to actually evaluate. To simplify, we assume that our later observations are independent of the noise in our earlier observations. To be precise, we approximate as

$$p(\sigma_{a,b}^{i,j} \mid \mathcal{D}_{a,b}) \simeq p(\sigma_a^i \mid \mathcal{D}_a)\, p(\sigma_b^j \mid \mathcal{D}_{a,b}), \tag{7}$$

giving

$$\begin{aligned} &p(y_\star \mid \mathcal{D}_{a,b}) \\ &\quad\simeq \sum_j p(\sigma_b^j \mid \mathcal{D}_{a,b}) \sum_i p(\sigma_a^i \mid \mathcal{D}_a) p(y_\star \mid \mathcal{D}_{a,b}, \sigma_{a,b}^{i,j}) \\ &\quad= \sum_j p(\sigma_b^j \mid \mathcal{D}_{a,b}) \sum_i p(\sigma_a^i \mid \mathcal{D}_a) \times \\ &\qquad \mathcal{N}\big(y_\star; m(y_\star \mid \mathcal{D}_{a,b}, \sigma_{a,b}^{i,j}), C(y_\star \mid \mathcal{D}_{a,b}, \sigma_{a,b}^{i,j})\big). \end{aligned} \tag{8}$$

Before trying to manage these sums, we will determine $p(\sigma_b \mid \mathcal{D}_{a,b})$. As before, this distribution gives us the probability of the observations $\mathcal{D}_b$ being faulty. For example, if we have only a single observation $\mathcal{D}_b$, $p\big(\text{fault}(\mathcal{D}_b) \mid \mathcal{D}_{a,b}\big) = p(\sigma_b = \sigma_f \mid \mathcal{D}_{a,b})$. We define

$$\begin{aligned} \tilde{m}(\mathbf{z}_b \mid \mathcal{D}_a) &\triangleq K_{b,a} \tilde{V}_a^{-1} \mathbf{z}_a \\ \tilde{C}(\mathbf{z}_b \mid \mathcal{D}_a, \sigma_b) &\triangleq V_b - K_{b,a}(\tilde{V}_a^{-1} - \tilde{W}_a^{-1}) K_{a,b} \\ &\quad - \tilde{m}(\mathbf{z}_b \mid \mathcal{D}_{a,b})^2, \end{aligned}$$

where both $\tilde{V}_a$ (or its Cholesky factor) and $\tilde{W}_a^{-1}$ were computed previously. By using (7) and again approximating a sum of Gaussians as a single Gaussian,

$$\begin{aligned} p(\sigma_b \mid \mathcal{D}_{a,b}) &= \frac{\sum_i p(\mathbf{z}_b \mid \mathcal{D}_a, \sigma_{a,b}^i) p(\mathbf{z}_a, \sigma_{a,b}^i)}{p(\mathbf{z}_{a,b})} \\ &\simeq \frac{\sum_i p(\mathbf{z}_b \mid \mathcal{D}_a, \sigma_{a,b}^i) p(\sigma_a^i \mid \mathcal{D}_a) p(\sigma_b)}{p(\mathbf{z}_b \mid \mathcal{D}_a)} \\ &\simeq \frac{\mathcal{N}\big(\mathbf{z}_b; \tilde{m}(\mathbf{z}_b \mid \mathcal{D}_a), \tilde{C}(\mathbf{z}_b \mid \mathcal{D}_a, \Sigma_b)\big) p(\sigma_b)}{p(\mathbf{z}_b \mid \mathcal{D}_a)}, \end{aligned}$$

where we have

$$\begin{aligned} &p(\mathbf{z}_b \mid \mathcal{D}_a) \\ &\quad= \sum_i \sum_j p(\mathbf{z}_b \mid \mathcal{D}_a, \sigma_{a,b}^i) p(\sigma_{a,b}^{i,j} \mid \mathcal{D}_a) \\ &\quad\simeq \sum_i \sum_j p(\mathbf{z}_b \mid \mathcal{D}_a, \sigma_{a,b}^i) p(\sigma_a^i \mid \mathcal{D}_a) p(\sigma_b^j) \\ &\quad\simeq \sum_j \mathcal{N}\big(\mathbf{z}_b; \tilde{m}(\mathbf{z}_b \mid \mathcal{D}_a), \tilde{C}(\mathbf{z}_b \mid \mathcal{D}_a, \sigma_b^j)\big) p(\sigma_b^j). \end{aligned} \tag{9}$$

Note that the product of (9) and (2) gives the likelihood of our hyperparameters.

Now, returning to (8), we will once again approximate a sum of Gaussians as a moment-matched single Gaussian. Our goal here is to reuse our previously evaluated sums over i to resolve future sums over i. As we gain more data, the faultiness of very old data becomes less important. We arrive at

$$p(y_\star \mid \mathcal{D}_{a,b}) \simeq \mathcal{N}\big(y_\star; \tilde{m}(y_\star \mid \mathcal{D}_{a,b}), \tilde{C}(y_\star \mid \mathcal{D}_{a,b})\big), \tag{10}$$

where we have

$$\begin{aligned} \tilde{m}(y_\star \mid \mathcal{D}_{a,b}) &\triangleq K_{\star,a,b} \tilde{V}_{a,b}^{-1} \mathbf{z}_{a,b} \\ \tilde{C}(y_\star \mid \mathcal{D}_a) &\triangleq K_{\star,\star} - K_{\star,a}(\tilde{V}_{a,b}^{-1} - \tilde{W}_{a,b}^{-1}) K_{a,\star} \\ &\quad - \tilde{m}(y_\star \mid \mathcal{D}_{a,b})^2. \end{aligned}$$

where

$$\begin{aligned} \tilde{V}_{a,b}^{-1} &\triangleq \sum_j p(\sigma_b^j \mid \mathcal{D}_{a,b}) \sum_i p(\sigma_a^i \mid \mathcal{D}_a) (V_{a,b}^{i,j})^{-1} \\ \tilde{W}_{a,b}^{-1} &\triangleq \sum_j p(\sigma_b^j \mid \mathcal{D}_{a,b}) \sum_i p(\sigma_a^i \mid \mathcal{D}_a) (V_{a,b}^{i,j})^{-1} \\ &\quad \times \mathbf{z}_{a,b} \mathbf{z}_{a,b}^\mathsf{T} (V_{a,b}^{i,j})^{-1}. \end{aligned}$$

Now, using the inversion by partitioning formula (Press et al., 1992, Section 2.7),

$$(V_{a,b}^{i,j})^{-1} = \begin{bmatrix} S_a^{i,j} & -S_a^{i,j} K_{a,b} (V_b^j)^{-1} \\ -(V_b^j)^{-1} K_{b,a} S_a^{i,j} & (V_b^j)^{-1} + (V_b^j)^{-1} K_{b,a} S_a^{i,j} K_{a,b} (V_b^j)^{-1} \end{bmatrix},$$

where $S_a^{i,j} \triangleq (V_a^i - K_{a,b}(V_b^j)^{-1} K_{b,a})^{-1}$.

Note that $(V_{a,b}^{i,j})^{-1}$ is affine in $S_a^{i,j}$, so that when

$$V_a \gg K_{a,b} V_b^{-1} K_{b,a}, \tag{11}$$

$(V_{a,b}^{i,j})^{-1}$ is effectively affine in $(V_a^i)^{-1}$. 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 particularly well correlated. On this basis, (11) seems reasonable for our application. Defining the affine map $f\colon (V_a^i)^{-1} \mapsto (V_{a,b}^{i,j})^{-1}$, then, noticing $\sum_i p(\sigma_a^i \mid \mathcal{D}_a) = 1$,

$$\sum_i p(\sigma_a^i \mid \mathcal{D}_a) f\big((V_a^i)^{-1}\big) \simeq f\Big(\sum_i p(\sigma_a^i \mid \mathcal{D}_a)(V_a^i)^{-1}\Big).$$

Therefore, for

$$\begin{aligned} \hat{V}_{a,b}^j &\triangleq \begin{bmatrix} \tilde{V}_a & K_{a,b} \\ K_{b,a} & V_b^j \end{bmatrix}, \\ \tilde{V}_{a,b}^{-1} &\simeq \sum_j p(\sigma_b^j \mid \mathcal{D}_{a,b}) (\hat{V}_{a,b}^j)^{-1}, \\ \tilde{V}_{a,b} &\simeq \begin{bmatrix} \tilde{V}_a & K_{a,b} \\ K_{b,a} & \tilde{V}_{b|a} + K_{b,a} \tilde{V}_a^{-1} K_{a,b} \end{bmatrix}, \end{aligned}$$

where

$$\tilde{V}_{b|a}^{-1} \triangleq \sum_j p(\sigma_b^j \mid \mathcal{D}_{a,b})(V_b^j - K_{b,a}\tilde{V}_a^{-1}K_{a,b})^{-1}$$

can be computed using the same trick as in (6) if b identifies a single observation and $j \in \{0, 1\}$. Note that the lower right hand element of $\tilde{V}_{a,b}$ defines the noise variance to be associated with observations $\mathcal{D}_b$. As before, we determine our predictions (10) by solving linear equations using the quantities

$$\tilde{R}_{a,b} \triangleq \operatorname{chol} \tilde{V}_{a,b}, \tag{12}$$

$$\tilde{T}_{a,b} \triangleq \operatorname{chol}(\tilde{R}_{a,b})^{-1}\mathbf{z}_a; \tag{13}$$

both of which can be efficiently determined (Osborne, 2010, Appendix B) using the evaluated $\tilde{R}_a$ and $\tilde{T}_a$.

We now turn to $\tilde{W}_{a,b}^{-1}$. Unfortunately, even if (11) were true, $\tilde{W}_{a,b}^{-1}$ is quadratic in $(V_a^i)^{-1}$. We will nonetheless assume that $\tilde{W}_{a,b}^{-1}$ is affine in $(V_a^i)^{-1}$. The quality of our approximation for $\tilde{W}_{a,b}^{-1}$ is much less critical than for $\tilde{V}_{a,b}^{-1}$, because the former only influences the variance of our predictions for the current predictant; any flaws in that approximation will not be propagated forward. Further, of course, if one probability dominates,

$$p(\sigma_a^i \mid \mathcal{D}_a) \gg p(\sigma_a^{i'} \mid \mathcal{D}_a) \quad \forall i' \neq i,$$

then the approximation is valid. With this,

$$\tilde{W}_{a,b}^{-1} \triangleq \sum_j p(\sigma_b^j \mid \mathcal{D}_{a,b})(\hat{V}_{a,b}^j)^{-1}\mathbf{z}_{a,b}\mathbf{z}_{a,b}^{\mathsf{T}}(\hat{V}_{a,b}^{i,j})^{-1}\,. \tag{14}$$

and we can solve for $K_{\star,(a,b)}\tilde{W}_{a,b}^{-1}K_{(a,b),\star}$ by efficiently updating using the previously computed quantity $\tilde{T}_a$.

3.1. Discussion

We return to the management of our hyperparameters θ. Unfortunately, analytically marginalizing θ is impossible. Most of the hyperparameters of our model can be set by optimizing 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 marginalize these hyperparameters online using Bayesian Monte Carlo (Osborne, 2010, Chapter 7), taking a fixed set of samples in their values and using the hyperparameter likelihoods (computed in (2) and (9)) 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.

The proposal above can be extended in several ways. First, we may wish to sum over more than one fault variance, which could be useful if observations are prone to faultiness in more than one mode. Note that if, instead of summing over a small number of known variances, we wished to marginalize 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 take appropriate, simple forms. These extensions would allow our algorithm to tackle the general problem of heteroskedasticity.

Our proposed algorithm steps through our data one at a time, so that $\mathcal{D}_b$ always contains only a single observation. However, it would be possible in general 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 of faultiness. However, within the framework afforded by our approximation (7), we are free to consider noise variances that are expected to change over time.

In some contexts it might be useful to perform inference about the fault contribution, rather than the signal of interest. This task is trivial; 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 a hard decisions as required.

4. Results

We test the effectiveness of the fault bucket 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 characterized 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 & US Geological Survey, 2006).

For a quantitative assessment, we used two semi-synthetic datasets where a typical fault has been in-

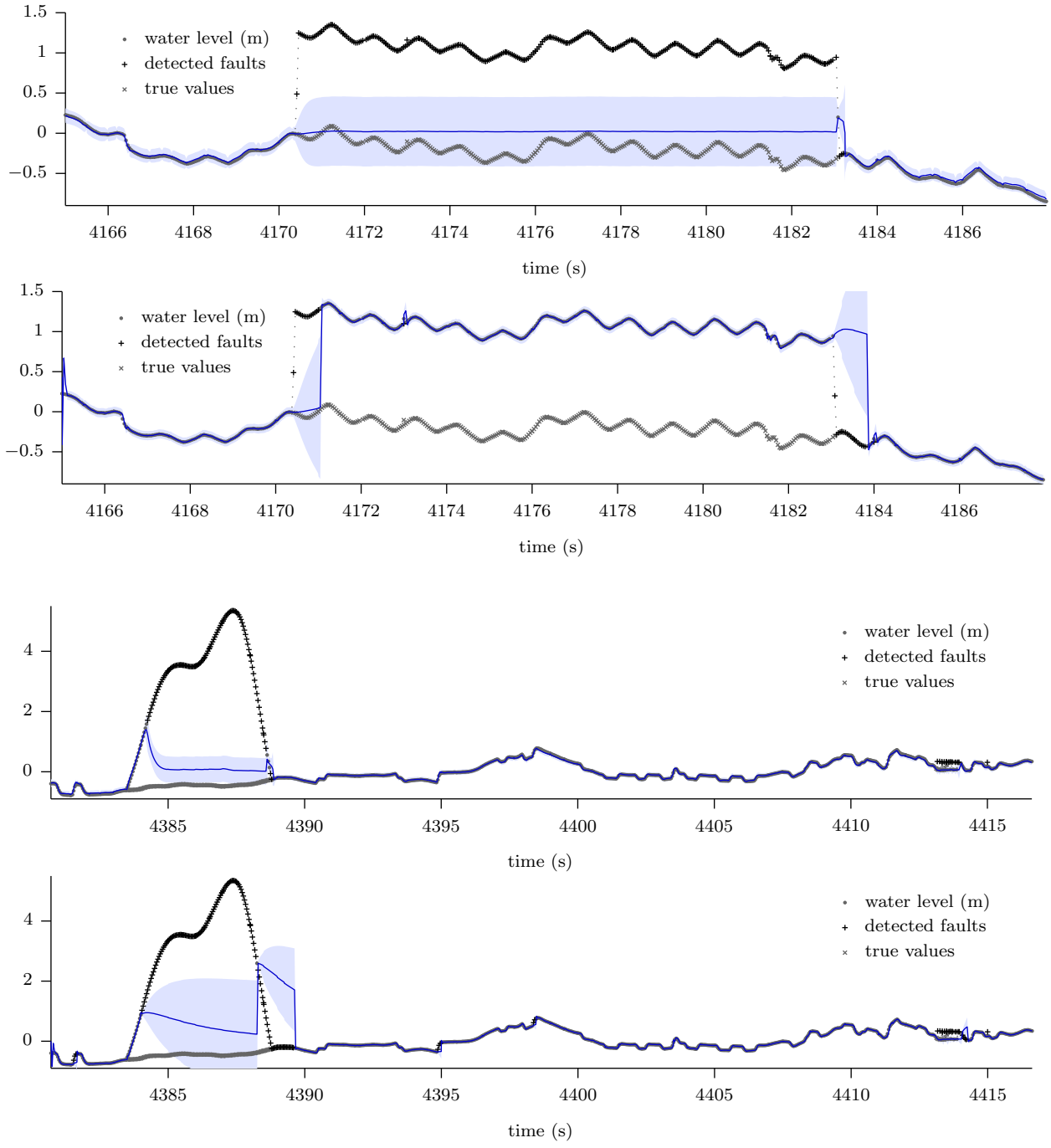

Figure 1. Mean and $\pm 3\sigma$ standard-deviation bounds for the predictions of (first and third) the fault-bucket algorithm and (second and fourth) the Kalman filter algorithm on (top two) the synthetic bias dataset and (bottom two), the synthetic anomaly dataset. Detected faults are marked in black crosses, and the unobserved true values are marked in grey circles.

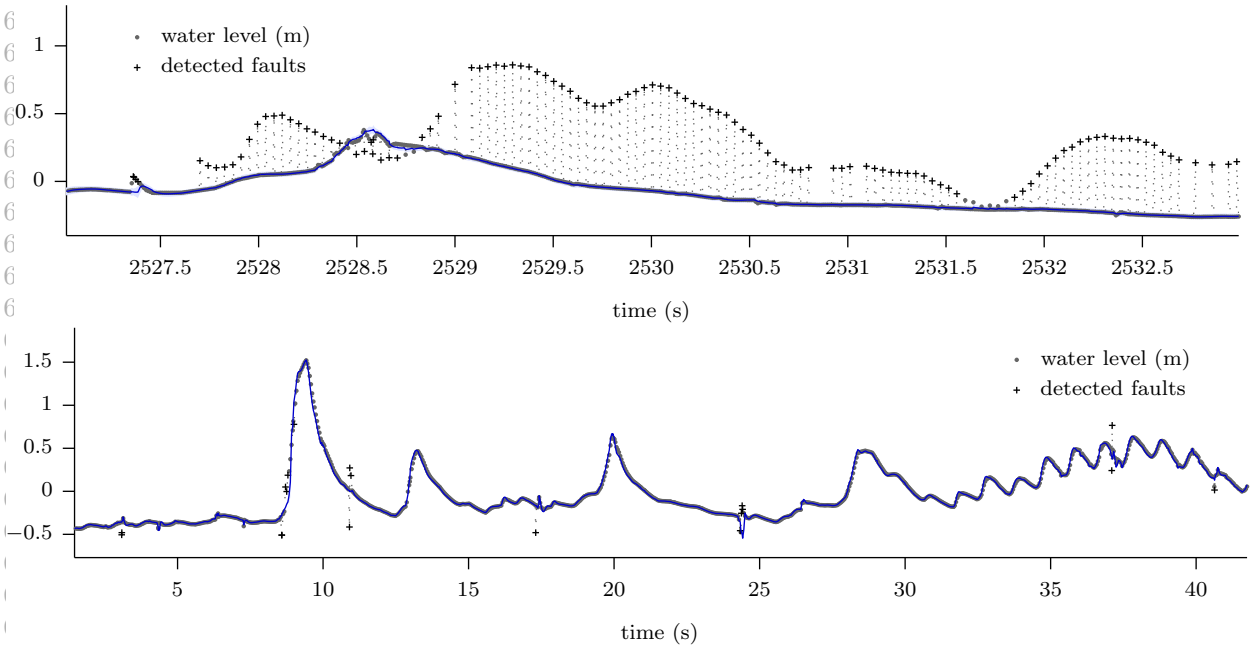

Figure 2. Mean and $\pm 3\sigma$ standard-deviation bounds for the predictions of the fault-bucket algorithm on (top), the painting dataset and (bottom), the "fishkiller" dataset. Detected faults are marked in black crosses, and the unobserved true values are marked in grey circles.

jected into clean sensor data. We then analyzed qualitative performance on two real data sets with actual faults. All measurements 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. The next dataset contains a fault type called "painting," which is an error that occurs when ice builds on a sensor obscuring some of the readings. It is characterized by frequent sensor spikes interlaced with the original, and still accurate, signal. Our final dataset, which we dub "fishkiller", comes from a sensor near a dam on a river in British Columbia, Canada. It contains an otherwise normal water level-reading that is occasionally interrupted by a short period of rapid oscillation. This occurs when dam operators open and close the floodgates too quickly. When this happens, the water level on the other side of the dam experiences a rapid drop during which time salmon can become trapped on the shores and stranded, leading to suffocation. Detecting these events is critical to proper regulation of dams in order to preserve the lives of these fish.

We implemented the algorithm described in Section 3 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 & Williams, 2006). The hyperparameters for this covariance, including the normal observation noise variance σ_n^2 were learned using training data nearby but not included in the test datasets. The unknown fault noise variance σ_f^2 was marginalized using Bayesian Monte Carlo, with 7 samples in this parameter. The prior probability of an observation being faulty was set to a constant value of 1% throughout.

As a basis for comparison, we employed the widely used Kalman filter. We first trained the parameters on a set of clean data (the same as that used for the fault-bucket approach), and on this dataset we attempted to make one-step-lookahead predictions in an online fashion. We then take the mean-squared error and set a threshold at 3 times the standard deviation on the training set. Using this threshold, when processing test data we look to see at each point if the Kalman filter prediction deviated more than this amount from the expected prediction. If it did, we label it a fault and

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 ($\log p(\mathbf{y} \mid \mathbf{x}, \mathcal{M})$), and the true-positive and false-positive rates of detection for faulty points (TPR and FPR), respectively. The top half of the table refers to the bias dataset; the bottom half to the change-in-dynamics dataset. The better of each pair of results is highlighted in bold.

Method	MSE	$\log p(\mathbf{y} \mid \mathbf{x}, \mathcal{M})$	TPR	FPR
KF	0.894	-7.11×10^5	0.053	0.073
FB	**0.034**	**-360**	**0.997**	**0.011**
KF	0.360	-3.54×10^4	**0.835**	0.046
FB	**0.075**	$\mathbf{-1.30 \times 10^4}$	0.798	**0.006**

discard the point for future predictions.

Figures 1 and 2 show the performance of the fault-bucket algorithm on the four datasets, as well as the performance of the Kalman filter on the synthetic datasets for comparison. In all cases, the fault-bucket algorithm did an excellent job of identifying faults when they occur, and made excellent predictions. The faults in the synthetic bias, painting, and fishkiller datasets were identified almost perfectly, with a small number of false negatives in the latter two. Most of the fault in the synthetic anomaly dataset was identified; however, the algorithm required a number of readings at the beginning of the smooth fault interval before it was able to conclude conclusively that the sharp rise was not part of the normal signal.

Table 1 displays quantitative measures of performance for the fault-bucket and Kalman filter algorithms on the synthetic datasets. In addition to superior predictive performance, our detection rates for the faulty points is also excellent, with the fault bucket algorithm identifying the fault in the synthetic anomaly dataset somewhat slower than the Kalman filter.

5. Conclusions

We have proposed a novel algorithm which we called the "fault bucket," for managing time-series data corrupted by faults unknown ahead of time. The chief theoretical contribution of the paper is a sequential algorithm for marginalizing over the possible faultiness of all observations. This allows for fast, principled prediction in the presence of unknown faults.

References

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

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

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

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

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

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

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

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

Osborne, M. A. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, University of Oxford, 2010. Available at www.robots.ox.ac.uk/~mosb/full_thesis.pdf.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY , USA, 1992.

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

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