Active multi-fidelity Bayesian online changepoint detection

Gregory W. Gundersen1  Diana Cai1  Chuteng Zhou2  Barbara E. Engelhardt1  Ryan P. Adams1

1Department of Computer Science, Princeton University
2Arm ML Research Lab

Abstract

Online algorithms for detecting changepoints, or abrupt shifts in the behavior of a time series, are often deployed with limited resources, e.g., to edge computing settings such as mobile phones or industrial sensors. In these scenarios it may be beneficial to trade the cost of collecting an environmental measurement against the quality or “fidelity” of this measurement and how the measurement affects changepoint estimation. For instance, one might decide between inertial measurements or GPS to determine changepoints for motion. A Bayesian approach to changepoint detection is particularly appealing because we can represent our posterior uncertainty about changepoints and make active, cost-sensitive decisions about data fidelity to reduce this posterior uncertainty. Moreover, the total cost could be dramatically lowered through active fidelity switching, while remaining robust to changes in data distribution. We propose a multi-fidelity approach that makes cost-sensitive decisions about which data fidelity to collect based on maximizing information gain with respect to changepoints. We evaluate this framework on synthetic, video, and audio data and show that this information-based approach results in accurate predictions while reducing total cost.

1 INTRODUCTION

Sequential data are rarely stationary. For example, a stock’s volatility might increase or a text stream’s topics might shift due to world events. A changepoint is an abrupt change in the generative parameters of sequential data. The goal of changepoint detection is to discover these structural changes, and thereby partition the data into regimes Changepoint detection is a broad class of algorithms, including the classic CUSUM algorithm [Page, 1954], hidden Markov models with a changing transition matrix [Braun and Muller, 1998], Poisson processes with varying rates [Ritov et al., 2002], two-phase linear regression [Lund and Reeves, 2002], and Gaussian process changepoint models [Saatçi et al., 2010]. The Bayesian approach is appealing due to the ability to specify priors and represent posterior uncertainty [Chib, 1998, Fearnhead, 2006, Chopin, 2007]. For streaming applications, exact filtering algorithms allow for online Bayesian detection of changepoints without retrospective smoothing [Fearnhead and Liu, 2007, Adams and MacKay, 2007].

Many applications of online changepoint detection are in real-time settings with limited resources for sensing and computation, such as content delivery networks [Akhtar et al., 2018], autonomous vehicles [Ferguson et al., 2015], and smart home and internet-of-things devices [Aminikhanghahi et al., 2018, Lee et al., 2018, Munir et al., 2019]. In such resource-constrained settings, the observations for a changepoint detector are typically environmental measurements, for example heart-rate data [Villarroel et al., 2017]. Trading the cost of collecting these data against their quality or “fidelity” may be useful, depending on how these fidelities affect changepoint estimation.

For example, since scaling up neural network capacity is an effective approach to improving model performance [Arora et al., 2018, Kaplan et al., 2020, Mahajan et al., 2018], a high-fidelity observation model might be a large but expensive-to-evaluate neural network. Retraining a smaller architecture or using compression algorithms such as distillation [Hinton et al., 2015], quantization [Gong et al., 2014, Hubara et al., 2017], or pruning [Frankle and Carbin, 2018] could produce a low-fidelity observation model. If the output of these neural networks is the input to a changepoint detector, then the fidelity of the networks will impact the quality of changepoint detection.

In such situations, the cost of Bayesian online changepoint detection (BOCD) could be reduced by making decisions about the fidelity of the observations. One view of BOCD is
as a model-based version of an exponentially-weighted moving average, estimating the weights from data rather than selecting them a priori. It determines which of the recent data matter for the current state. This view motivates our multi-fidelity approach: if changepoints are easily identified and the data can be partitioned into stationary regimes, there is no need for expensive high-fidelity observations when BOCD’s posterior confidence about changepoints is high.

In our framing of the problem, we must choose which data fidelity to use and pay a fixed cost to make this choice. In the neural network example, we can evaluate either an expensive or cheap neural network to obtain a high- or low-fidelity representation of a raw measurement. To make this choice, we propose an information-theoretic approach, similar to the active data collection strategy proposed by MacKay [1992] and to approaches used in Bayesian optimization [Hernández-Lobato et al., 2014], preference learning [Houlsby et al., 2012], and Bayesian quadrature [Gessner et al., 2020]. We choose the data fidelity with maximal weighted information rate (gain over cost) for the posterior distribution over changepoints. The weights allow modelers to specify a desired computational budget. This results in policies that use lower-fidelity data in regimes with higher posterior certainty.

Contributions. First, we formulate a new version of an important problem: online changepoint detection with multiple data sources of varying cost and quality. The task is to choose which fidelity to use at each time point to make accurate predictions while minimizing costs. Second, we propose active selection of each datum’s fidelity based on the expected informativeness of observations from each fidelity, and choose the one that maximizes the information rate for the posterior distribution over changepoints. Finally, we demonstrate the empirical performance of our algorithm on both synthetic and real-world data. We show that in many real-world scenarios, despite the extra step of computing information gain, our model reduces the total computational budget while maintaining good predictive accuracy.

2 BAYESIAN ONLINE CHANGEPONT DETECTION

We begin by reviewing the BOCD algorithm [Adams and MacKay, 2007, Fearnhead and Liu, 2007]. Our data are a contiguous sequence of observations in time, \( \mathbf{X}_{1:T} := \{ x_1, \ldots, x_T \} \) where \( x_t \in \mathbb{R}^D \). Assume that the data can be partitioned such that, within each partition, the data are i.i.d. [Barry and Hartigan, 1992], governed by partition-specific parameters \( \theta \). The transition from one partition into another results in an abrupt change from one set of parameters to another. This transition is referred to as a changepoint.

Denote the parameters at time \( t \) as \( \theta_t \). In the changepoint process, these parameters are determined in one of two ways: either a changepoint has occurred at time \( t \), in which case the parameters are drawn afresh from a prior distribution \( \Pi \), or a changepoint has not occurred and the parameters are \( \theta_t = \theta_{t-1} \), i.e., they stay the same. We model the arrival of changepoints as a discrete time Bernoulli process with hazard rate \( 1/\beta \), resulting in a geometric distribution over partition lengths with mean \( \beta \in \mathbb{R}_{>0} \).

In the online setting, the primary quantity of interest is the time since the last changepoint, which we refer to as the run length. We denote the run length at time \( t \) as \( r_t \), which takes values in the non-negative integers. Thus, a changepoint at \( t \) means \( r_t = 0 \). At time \( t \), the BOCD algorithm estimates the posterior marginal distribution over the run length \( p(r_t \mid \mathbf{X}_{1:t}) \). We refer to this distribution as the run-length posterior. Online updating of the run-length posterior is made easy via a recursion that is essentially the same as the message-passing (dynamic programming) approach to hidden Markov models [Baum and Petrie, 1966, Rabiner, 1989]:

\[
p(r_t \mid \mathbf{X}_{1:t}) \propto p(r_t, \mathbf{X}_{1:t}) = \sum_{r_{t-1}} p(r_{t-1}, \mathbf{X}_{1:t-1}) p(r_t \mid r_{t-1}, \mathbf{X}_{1:t-1})
\]

\[
= \sum_{r_{t-1}} p(r_{t-1} \mid r_{t-1}, \mathbf{X}_{1:t-1}) p(\mathbf{X}_t \mid r_{t-1}) p(r_{t-1} \mid \mathbf{X}_{1:t-1})
\]

\[
= \sum_{r_{t-1}} p(r_{t-1}) p(\mathbf{X}_t \mid r_{t-1}) p(r_{t-1} \mid \mathbf{X}_{1:t-1}), \tag{1}
\]

where the cancellations arise from Markovian assumptions we have made: 1) the probability of a changepoint at time \( t \) is independent of data before \( t \), given knowledge of \( r_{t-1} \), and 2) the predictive distribution over the data \( \mathbf{x}_t \) at time \( t \) is independent of past run lengths, given knowledge of the current run length \( r_t \). The three terms within the sum have a convenient interpretation as the prior, the predictive distribution, and the estimated joint distribution from the previous time step. These are the only ingredients necessary for a straightforward online filtering algorithm.

The Bernoulli process prior above is in an unconventional form that represents the time since the last changepoint:

\[
p(r_t \mid r_{t-1}) = \begin{cases} 
1/\beta & \text{if } r_t = 0, \\
1 - 1/\beta & \text{if } r_t = r_{t-1} + 1, \\
0 & \text{otherwise.}
\end{cases} \tag{2}
\]

In other words, the run length \( r_t \) must either increase by one from the previous time point or drop to zero.

The construction so far has not depended on the specifics of the data-generating distribution \( P_{\theta} \), which appears as a part of the posterior predictive distribution in Equation (1):

\[
p(\mathbf{x}_t \mid r_{t-1} = \ell, \mathbf{X}_{1:t-1}) = \int_{\theta_r} p_{\theta_r}(\mathbf{x}_t) \pi(\theta_t \mid X^{(t)}) \, d\theta_t, \tag{3}
\]
where \( p_{\theta_\ell}(\cdot) \) is the probability density function associated with the distribution \( P_{\theta_\ell} \), \( \pi(\theta \mid \cdot) \) is the probability density function associated with the posterior distribution w.r.t. \( \theta \), and \( X^{(\ell)} := X_{t-\ell:t-1} \) denotes the most recent \( \ell \) data. This is a key property of the BOCD algorithm: conditioning on \( r_t = \ell \) means that only the most recent \( \ell \) data need to be accounted for in the posterior distribution. When the data distribution \( P_{\theta_0} \) is chosen to allow for a conjugate prior for \( \Pi \), then the computations necessary for the recursion are relatively simple: it is only necessary to maintain a set of sufficient statistics for each \( r_t \) hypothesis. These statistics can be easily updated via addition, and the posterior predictive is often available in closed form. (See Adams and MacKay [2007] for further discussion.) When more complicated models are used, approximate inference or numerical integration are necessary.

Given the run-length posterior, we can compute a predictive distribution to make online predictions that are robust to changepoints by marginalizing out the run length, i.e., by computing a mixture of posterior predictive distributions— which are already available from the recursion—under the run-length posterior:

\[
\pi(\theta_t \mid D_{1:t}) \propto \pi(\theta_t) \prod_{i=1}^{t} p_{\theta_i}(x_i)^{\xi_i}. \tag{5}
\]

Equation (4) underscores the value of modeling the run-length in this construction: it provides a model-based approach to decide which data are currently relevant for predicting the next observation. That is, the value of \( r_t \) explicitly captures the size of the current partition, i.e., what recent data share the same parameters.

The basic framework for BOCD has been extended in a number of ways, such as learning the changepoint prior [Wilson et al., 2010], adding Thompson sampling for multi-armed bandits with changing rewards [Mellor and Shapiro, 2013], estimating uncertainty bounds on the number and location of changepoints [Ruggieri and Antonellis, 2016], and using \( \beta \)-divergences for robustness against outliers [Knoblauch et al., 2018]. While changepoint detection has been explored in the context of active data selection [Osborne et al., 2010, Hayashi et al., 2019], to our knowledge, the BOCD framework has not been considered in multi-fidelity settings.

## 3 MULTI-FIDELITY CHANGEPONT DETECTION

We now extend the BOCD framework to the multi-fidelity setting, referring to our algorithm as MF-BOCD. Our central assumption is that, at any time point \( t \), we choose the quality of our observation, with higher fidelity (lower noise) having greater cost. We generally take this cost to be computational, but it could also be quantified in terms of resources such as money or energy. Given the selected data fidelities, we can again recursively compute a run-length posterior (Section 3.2). Given this multi-fidelity run-length posterior, the algorithm then selects the data fidelity that maximizes a cost-sensitive information rate objective (Section 3.4).

### 3.1 MULTI-FIDELITY POSTERIOR PREDICTIVE

Again, suppose we have a distribution \( P_{\theta_0} \) and prior \( \Pi \), and the task is to estimate the parameter \( \theta_t \) in the presence of changepoints. Our data are again the contiguous sequence \( X_{1:T} \).

However, we now assume each observation \( x_t \) has an associated value \( \xi_t \in [0,1] \), which we call the fidelity. The fidelities \( z_{1:T} := \{\xi_1, \ldots, \xi_T\} \) are non-random and take values from a set \( Z \). In the experiments, we only consider the case when the cardinality of \( Z \) is two, i.e., we only have low- and high-fidelities, but this is not a necessary restriction. Let our sequence of observations and chosen fidelities be \( D_{1:T} := \{(x_1, \xi_1), \ldots, (x_T, \xi_T)\} \). The role of the fidelity \( \xi_t \) is to re-weight the associated probability function \( p_{\theta_\ell}(x) \) in a multi-fidelity posterior (MF-posterior). At time \( t \), the MF-posterior is:

\[
\pi(\theta_t \mid D_{1:t}) \propto \pi(\theta_t) \prod_{i=1}^{t} p_{\theta_i}(x_i)^{\xi_i}. \tag{5}
\]

Here, \( \pi(\cdot) \) is the probability density function associated with the prior distribution \( \Pi \).

Intuitively, the effect of data re-weighting on the MF-posterior is a density that concentrates as if the contribution of \( T \) samples were \( \sum_{t=1}^{T} \xi_t \) number of data points instead of \( T \) data points. Figure 1 illustrates the MF-posterior of a conjugate Gaussian model with known variance (discussed in Section 3.3). Here the data are generated from a standard normal distribution, and the MF-posterior \( \pi(\theta_t \mid D_{1:T}) \) is visualized for varying \( \xi_{LF} \) and fixed \( \xi_{HF} = 1 \). As \( \xi_{LF} \) decreases, the MF-posterior becomes less concentrated with a larger variance and increased influence from the prior.

Re-weighting terms in the likelihood has been considered under various names, such as safe Bayes [Heide et al., 2020, Grünwald et al., 2017], generalized posteriors [Walker and Hjort, 2001, Bissiri et al., 2016], coarsened posteriors [Miller and Dunson, 2018], and Bayesian data re-weighting [Wang et al., 2017]. In our framing of this model, we must choose the fidelity \( \xi_t \) of each observation \( x_t \), paying a fixed cost to make this choice.

When using a member of the exponential family with a conjugate prior, one has analytical expressions of the MF-posterior and MF-posterior predictive. Let the distributions on \( x \) and \( \theta \) have the following functional forms:

\[
p_{\theta_\ell}(x) = h_1(x) \exp \left\{ \theta_\ell^T u(x) - a_1(\theta_\ell) \right\}, \tag{6}
\]

\[
\pi_{x,\nu}(\theta_t) = h_2(\theta_t) \exp \left\{ \theta_t^T \chi - \nu a_1(\theta_t) - a_2(\chi, \nu) \right\}. \tag{7}
\]
within each panel, the percentage of (low-fidelity) weighted data likelihoods (LF%) varies. When we can compute. See Appendix where, using exponential family terminology, $\theta_i$ are now natural parameters, $u(x)$ are sufficient statistics, $a_1(\cdot)$ and $a_2(\cdot)$ are log normalizers, and $h_1(\cdot)$ and $h_2(\cdot)$ are base measures. Then the MF-posterior is

$$
\pi_{\mathbf{x}_t, u}(\theta_t) = \prod_{l=1}^{T} p_{\theta}(x_t) \zeta_{l-1}
\propto h_2(\theta_t) \exp\left\{\mathbf{\theta}_t^\top \mathbf{x}_t - \nu a_1(\theta_t)\right\},
$$

where $\mathbf{x}_t = \mathbf{x} + \sum_{t=1}^{t} \zeta_i u(x_i)$ and $\nu_t = \nu + \sum_{t=1}^{t} \zeta_i$. The effect of the $\zeta_i < 1$ is to down-weight the sufficient statistics of $x_i$. When $\zeta_i = 1$ for all $i$, Equation (8) reduces to the standard posterior for exponential family models.

We can now construct a multi-fidelity version of Equation (3): a posterior predictive distribution that depends on data fidelities. Let $\mathbf{D}^{(t)} := \mathbf{D}_{t:t-1}$ denote the most recent $t$ data and associated fidelities (i.e., run length $r_t = t$), and let the associated parameter estimates be:

$$
\mathbf{\chi}_t := \mathbf{x} + \sum_{t=1}^{t-1} \zeta_r u(x_r), \quad \nu_r := \nu + \sum_{r=t}^{t-1} \zeta_r.
$$

Then the MF-posterior predictive is

$$
p(x_t | r_t = t, \zeta_t, \mathbf{D}^{(t)}) = \int \rho_{\theta}(x_t) \zeta_{t} \pi(\theta_t | \mathbf{D}^{(t)}) d\theta_t
= h_1(x_t) \exp\left\{\frac{a_2(\zeta_r u(x_r) + \chi_r, \zeta_r + \nu_r)}{\exp(a_2(\chi_r, \nu_r))}\right\},
$$

provided $h_1(x_t) \zeta_{t}$ induces a distribution whose normalizer we can compute. See Appendix A for a proof. This result is an extension of prior work on power posteriors for the exponential family [Miller and Dunson, 2018] to multiple values of powers. Equation (10) can be interpreted as a traditional posterior predictive distribution for exponential family models but with the sufficient statistics weighted by the fidelities. Since BOCD is amenable to fast online updates for exponential families, inference using fidelities is often no harder than using the ordinary posterior.

Note that for some multi-fidelity models, the MF-posterior $p(\theta_t | r_t = t, \mathbf{D}^{(t)})$ may not have an analytic form even when $p(\theta_t | \mathbf{X}^{(t)})$ does. In this paper, we only consider models in the exponential family, since this restriction often allows for efficient online updates. However, our approach may also extend to conditionally conjugate models. (See Miller and Dunson [2018] for a discussion.) In such settings, we could apply online variational inference to approximate predictive distributions [Turner et al., 2013]. As in standard BOCD, computing this predictive distribution without conjugate priors requires numerical approximations.

### 3.2 Multi-Fidelity Run-Length Posterior Estimation

To accommodate multi-fidelity observations, we must modify the online posterior estimation procedure for the run lengths. We now condition the recursion on both the observations and data fidelities:

$$
p(r_t = t | \mathbf{D}_{1:t}) \propto p(r_t, \mathbf{X}_{1:t} | z_{1:t})
= \sum_{r_{t-1}} p(r_t | r_{t-1}) p(\mathbf{x}_t | r_{t}, \zeta_t, \mathbf{D}^{(t)})
$$

Similar to Equation (1), in the multi-fidelity case, the joint distribution of Equation (11) decomposes into a changepoint prior $p(r_t | r_{t-1})$, a predictive distribution, and the previous message. The latter two are now conditioned on fidelities. Thus, we can efficiently update the run length posterior in a recursive manner.

### 3.3 Examples

Before discussing how we choose fidelities, we demonstrate our approach with two examples of multi-fidelity models,
which we use in Section 4. To simplify notation, we ignore the run length in this section, since it only specifies which data need to be accounted for in the MF-posterior distribution. See Appendix A for more detailed derivations.

Multi-fidelity Gaussian. Consider a univariate Gaussian model with known variance \( \sigma_x^2 \),
\[
    x_i \overset{iid}{\sim} \mathcal{N}(\theta_i, \sigma_x^2), \quad \theta_i \sim \mathcal{N}(\mu_0, \sigma_\theta^2).
\] (12)
The multi-fidelity likelihood is
\[
    \prod_{i=1}^t p_{\theta_i}(x_i)^{\zeta_i} \propto \prod_{i=1}^t \exp \left\{-\frac{\zeta_i}{2\sigma_x^2} (x_i - \theta_i)^2 \right\},
\] (13)
and the MF-posterior is the product of \( t+1 \) independent Gaussian densities, which is again a Gaussian:
\[
    \pi(\theta_t \mid D_{1:t}) \propto N(\theta_t \mid \mu_0, \sigma_\theta^2) \prod_{i=1}^t N(x_i \mid \theta_i, \sigma_x^2/\zeta_i)
\] (14)
\[
    \propto N(\theta_t \mid \mu_t, \sigma_t^2),
\] (15)
where
\[
    \frac{1}{\sigma_t^2} = \frac{1}{\sigma_0^2} + \sum_{i=1}^t \frac{\zeta_i}{\sigma_x^2}, \quad \mu_t = \sigma_t^2 \left( \frac{\mu_0}{\sigma_0^2} + \sum_{i=1}^t \frac{\zeta_i x_i}{\sigma_x^2} \right).
\] (16)
The MF-posterior predictive distribution can be computed by integrating out \( \theta_t \). This is a convolution of two Gaussians—the posterior in Equation (15) and the prior \( \pi(\theta_t) \)—which is again Gaussian:
\[
    p(x_{t+1} \mid \zeta_{t+1}, D_{1:t}) = N\left(x_{t+1} \mid \mu_t, \sigma_t^2 + \sigma_x^2/\zeta_{t+1}\right).
\] (17)
In this example, the fidelity \( \zeta \) has the natural interpretation of increasing the posterior variance when \( \zeta < 1 \). In Equation (11), this has the effect that the multi-fidelity run length posterior is less concentrated. Any confidence in a change point is by definition lower.

Multi-fidelity Bernoulli. Consider a Bernoulli model,
\[
    x_i \overset{iid}{\sim} \text{Bernoulli}(\theta_i), \quad \theta_i \sim \text{Beta}(\alpha_0, \beta_0).
\] (18)
The MF-posterior is proportional to a beta distribution
\[
    \pi(\theta_t \mid D_{1:t}) = \text{Beta}(\alpha_t, \beta_t) \text{ with parameters}
\]
\[
    \alpha_t := \alpha_0 + \sum_{i=1}^t \zeta_i x_i, \quad \beta_t := \beta_0 + \sum_{i=1}^t \zeta_i (1 - x_i).
\] (19)
The multi-fidelity posterior predictive distribution is the same as for a standard beta-Bernoulli model with \( \alpha_t \) and \( \beta_t \) and additional re-weighting due to \( \zeta_{t+1} \):
\[
    p(x_{t+1} \mid \zeta_{t+1}, D_{1:t}) = \frac{B(\alpha_t + \alpha_{t+1}, \beta_t + \beta_{t+1})}{B(\alpha_t, \beta_t)},
\] (20)
where \( B(\cdot, \cdot) \) is the beta function. When \( \zeta_t < 1 \), the fidelity has the natural effect of discounting count observations.

3.4 Active Fidelity Selection
So far, we have only discussed modeling data with multiple fidelities. However, in our framing of the problem, we must actively decide the fidelity of our observation \( x_t \), i.e., we must pick \( \zeta_t \in \mathbb{Z} \). We propose an information-theoretic approach, similar to ideas in active data collection [MacKay, 1992], Bayesian optimization [Hernández-Lobato et al., 2014], preference learning [Houlsby et al., 2012], and Bayesian quadrature [Gessner et al., 2020].

We propose maximizing the weighted information rate of the multi-fidelity run length distribution. After observing \( D_{1:t-1} \) observations and fidelities, our current information about \( r_t \) is the Shannon entropy \( H[p(r_t \mid D_{1:t-1})] \). Since we must choose a fidelity without observing \( x_t \), we want to choose the one that minimizes the expected entropy with respect to the predictive distribution in Equation (4). Thus, we choose the fidelity that maximizes the information gain of the run length posterior. The utility of \( \zeta_t \) is therefore
\[
    U(\zeta_t) = H[r_t \mid D_{1:t-1}] - E_{x_t}[H[r_t \mid D_{1:t-1}, x_t, \zeta_t]].
\] (21)
At time \( t \), the left term in Equation (21) is easy to compute, since we have already computed the posterior distribution \( p(r_{t-1} \mid D_{1:t-1}) \). We simply roll our estimation forward in time according to the changepoint process and without conditioning on new data. Furthermore, this value is the same for all fidelities, and therefore an equivalent formulation is to minimize the expected run length entropy, the right term in Equation (21). This entropy term is easy to compute because it is with respect to a discrete distribution that we can estimate at time \( t \). The expectation is with respect to the predictive distribution (Equation (4)) and must be approximated in general.

However, we are not interested in the fidelity that just maximizes information gain regardless of cost. If this were the case, we would simply always use the highest fidelity. Let \( \lambda(\zeta) \) denote the cost of fidelity \( \zeta_t \). In general, \( \lambda(\cdot) \) could be a function of the input domain, but here we assume it is a scalar constant that is known, e.g., wall-time, energy usage, or floating point operations. Then the information rate of fidelity \( \zeta_t \) at time \( t \) is \( \alpha(\zeta_t) := U(\zeta_t)/\lambda(\zeta) \). However, given the interaction of fixed costs and estimated fidelities, it is possible that the maximum information rate is always achieved using the highest (or lowest) fidelity. In this case, we may still want some amount of low-fidelity (or high-fidelity) usage depending on dataset size and computational budget. To address this, consider arbitrary weights \( w(\zeta_t) \). Our decision rule is then: use fidelity \( \zeta_t^* \) that maximizes the weighted information rate:
\[
    \zeta_t^* := \arg \max_{\zeta_t \in \mathbb{Z}} w(\zeta_t) \alpha(\zeta_t).
\] (22)
Note that the weights can be tuned on held-out data to achieve a desired expected budget. Introducing weights is
We considered alternative decision rules to Equation (22). For example, in scenarios with just two fidelities (low and high), we explored a decision rule that picked the low-fidelity datum when the absolute difference in information gains was less than some margin hyperparameter. However, empirically, this resulted in frequent switching between fidelities since the two information gains were often quite close in value. We found that information rate was more stable because it requires a more significant change in information gain to induce a switch. See Appendix B for a discussion and additional results.

3.5 PRACTICAL CONSIDERATIONS

Analyzing costs. Since we are motivated by real-time decision-making, a sensible question is whether our decision-making algorithm is cheaper than using only high-fidelity observations. Here, we give a complete example of the cost for the beta-Bernoulli model. Since the predictive distribution is easy to work with, a useful reformulation of Equation (21) is

\[ U(\zeta_t) = E[\mathbf{x}_t | \mathbf{D}_{1:t-1}] - E_{r_t}[E[\mathbf{x}_t | \mathbf{D}_{1:t-1}, r_t, \zeta_t]], \]

(23)

which uses the symmetry of information gain. At time \( t \), the cost in floating point operations (flops) of computing Equation (23) is \( 32t + 1 \) flops. The cost grows linearly with time because computing information gain requires summing over the run length posterior \( p(r_t | \mathbf{D}_{1:t-1}) \), and the support of this distribution grows linearly with time. However, Fearnhead and Liu (2007) proposed an optimal resampling algorithm, similar to particle filtering, that enables efficient approximate inference. This allows for a fixed cost to compute information gain. For example, with 10,000 particles, computing the information gain for the Bernoulli model requires 0.32 million flops. For comparison, consider MobileNets, which are a class of efficient neural networks designed for mobile and embedded vision applications [Howard et al., 2017]. The smallest reported MobileNet requires 41 million multi-adds (82 million flops). Thus, computing the beta-Bernoulli information gain twice (when the cardinality of \( Z \) is 2) is 140 times cheaper than evaluating the smallest MobileNet, while still using 10,000 particles in the run length posterior estimation.

Estimating fidelity \( \zeta_t \). A second practical consideration is estimating \( \zeta_t \). In the Gaussian case with known variance \( \sigma^2_z \), we can estimate \( \zeta_t / \sigma^2_z \) using the sample variance of held-out data and then calculate the value for \( \zeta_t \). In the Bernoulli case, we use model accuracy as a proxy for \( \zeta_t \). For example, if a binary classifier has a true positive rate of 90%, we treat an observation of 1 as a 0.9 using \( \zeta_t = 0.9 \).

4 EXPERIMENTS

In this section, we empirically evaluate our algorithm on synthetic, video, and audio data, and compare performance of MF-BOCD against BOCD using only low- or high-fidelity data, as well as a randomized baseline. Please see Appendix C for didactic code and the repository for a complete implementation.  

To evaluate our framework, we define two metrics. Let \( \mathbf{X}_{1:T} := \{ \mathbf{x}_1, \ldots, \mathbf{x}_T \} \) denote the mean of the predictive distribution, Equation (4), of BOCD or MF-BOCD for all time points. Then the reported mean squared error (MSE) is between \( \mathbf{X}_{1:T} \) from the evaluated model and \( \mathbf{X}_{1:T} \) from BOCD using only high-fidelity data. Now let \( \mathbf{R}_{1:T} \) denote a lower triangular matrix denoting the run length posterior at all time points. The \( L_1 \) distance is between \( \mathbf{R}_{1:T} \) from the evaluated model and \( \mathbf{R}_{1:T} \) from BOCD using only high-fidelity data. In other words, we compare the evaluated model to the best it could have done in practice.

As a baseline, we compare MF-BOCD with a model that randomly switches between fidelities and which uses roughly the same percentage of high-fidelity data as MF-BOCD. For the random switching model, the decision to use low-fidelity data was based on the outcome of a Bernoulli random variable with bias equal to the percentage of low-fidelity data used by MF-BOCD, normalized to \([0, 1]\). This comparison isolates the question: is it when a multi-fidelity model uses high-fidelity data that improves performance or just the presence of high-fidelity data at all?

4.1 NUMERICAL EXPERIMENTS

The purpose of these experiments is to demonstrate that information rate is a useful decision rule and to build intuition about the model’s behavior in a controlled setting. Consider a synthetic univariate signal with two fidelities. We assume data are i.i.d. Gaussian within each partition, and we use the Gaussian multi-fidelity model described in Section 3.3. When a changepoint occurs, the parameter \( \theta_t \) is drawn from a prior \( \mathcal{N}(1, 3) \). The data is then drawn from a distribution \( x_t \sim \mathcal{N}(\theta_t, \zeta_t / \sigma^2_z) \) where \( \sigma^2_z = 1 \). Our fidelities are from the set \( Z = \{ \zeta_{HF}, \zeta_{LF} \} \). We set the higher fidelity to \( \zeta_{HF} = 1 \) and the lower fidelity to \( \zeta_{LF} = 1/2 \). Thus, low-fidelity data have twice the variance. Costs are arbitrary in this setting, and we set them to \( \lambda(\zeta_{HF}) = 2 \) and \( \lambda(\zeta_{LF}) = 1 \). We simulated the data using \( T = 500 \) observations with a changepoint prior with \( 1/\beta = 1/100 \).

This experiment illustrates information rate as a decision rule as described in Section 3.4. In regions in which the model is confident about the run length posterior, low-fidelity data are preferred because both fidelities provide sufficient information. However, when the model is uncertain.

\[https://github.com/princetonlips/mf-bocd\]
Figure 2: Comparison of two multi-fidelity models. Orange x marks and blue circles denote low- and high-fidelity data respectively. (Left two panels) A multi-fidelity model with random switching between fidelities. The probability of switching to low-fidelity data was chosen to be the fraction of low-fidelity observations used in the experiment in the right column. (Right two panels) MF-BOCD: a multi-fidelity model that actively selects the fidelity based on information rate.

about the run length posterior, the high-fidelity observations are preferred (Figure 2). In contrast to information-based switching, the multi-fidelity model with random switching has both higher MSE and \( L_1 \) metrics. This suggests that while just using some high-fidelity data is useful, choosing when to use that high-fidelity data can improve performance. While this result is illustrative, we also include two randomized ablation experiments in Appendix D.

4.2 CAMBRIDGE VIDEO DATA

The numerical experiments provide a useful illustration of the role of information gain in a controlled setting. However, the fidelities and costs are contrived. In this section, we present a complete example of MF-BOCD with observation models and associated costs for the purpose of real-time detection of changepoints in streaming video data.

The Cambridge-driving Labeled Video Database (CamVid) is a collection of over ten minutes of video footage with object class semantic labels from 32 classes [Brostow et al., 2009]. The videos have been manually labeled at 1 frame per second, for just over 700 images. Each frame is \( 320 \times 480 \) pixels. For observation models, we used pretrained V3 MobileNets [Howard et al., 2017, 2019]. The high-fidelity model is larger and more accurate (Table 4 in Appendix E).

The output of each observation model is a segmentation mask, which we converted to a binary signal depending on whether or not a given class is in the image. In particular, we used the “fence” signal because fences go in and out of the frame but typically remain in a sequence of frames for a brief period. We then fit the multi-fidelity Bernoulli model (Section 3.3) to the CamVid test set. We used the predictive version of information gain, Equation \( (23) \). We arbitrarily set the low-fidelity model’s cost to 1 and the high-fidelity model’s cost as function of that, \( 36.7/19.5 \approx 1.9 \), using the number of flops (in billions) as a proxy for cost (Table 4). The high-fidelity model used \( \zeta_{HF} = 1 \). The low-fidelity model’s fidelity is a function of the difference in mean intersection-over-union for each model, \( \zeta_{LF} = 1 - (0.723 - 0.674) \approx 0.95 \).

We found that the output of low- and high-capacity neural networks were a reasonable proxy for low- and high-fidelity data. Standard BOCD using only high-fidelity observations estimates a run-length posterior that captures more groundtruth changepoints and has a predictive mean with smaller MSE and \( L_1 \) distance than BOCD using only low-fidelity data. The multi-fidelity model’s decision rule weights were tuned to approximate total computational cost of 50% low-fidelity data using cross-validation data, and the randomized approach flips a fair coin to choose the data fidelity. On test data, MF-BOCD estimated a run length posterior that still closely matched the high-fidelity run-length posterior (Figure 3). The information-based approach results in a better predictive mean (MSE) and better run length posterior estimation (\( L_1 \) distance) than both the low-fidelity and randomized versions.

Finally, we estimated the computational cost of MF-BOCD relative to baselines. With roughly 50% low-fidelity data, the costs in billions of flops for MF-BOCD was 4827, for BOCD using just low-fidelity data was 3333, and for BOCD using just high-fidelity data was 6303. The cost of decision making was marginal, requiring 0.00046 billion flops (Appendix E). As this calculation demonstrates, making a decision between high- and low-capacity neural networks can be significantly cheaper than evaluating either model. So while random usage of low-fidelity data is a reasonable approach to lowering the computational budget, decision-making can improve inference and predictions with marginal added cost.
4.3 MIMII AUDIO DATA

Next, we evaluated MF-BOCD on the sound dataset for Malfunctioning Industrial Machine Investigation and Inspection [MIMII, Purohit et al., 2019]. The raw data are 10-second audio clips recorded from 4 different industrial machines (slide rails in this experiment) during either normal or anomalous operation. For example, anomalous conditions might involve rail damage, a loose belt, or no grease. The high-fidelity observation model is a depth-wise separable convolutional neural network [MicroNets, Banbury et al., 2020]. The low-fidelity observation model is a two-layer fully-connected neural network. Both models take frames of log-Mel spectrograms of audio signals as inputs and return an anomaly score as output. They were pretrained on audio clips of normal behavior. Then each 10-second test set clip was converted to 14 anomaly scores using these observation models. The anomaly score is a number between 0 and 1, with 0 indicating normal. We thresholded the anomaly scores to produce binary labels. We picked machine- and model-specific thresholds using ROC curves. (See Appendix E for details.)

To randomly generate audio files with changepoints, we sampled a sequence of Bernoulli random variables \( y_1:T \). Then for each \( y_t \), we chose a normal (anomalous) audio clip uniformly at random with replacement if \( y_t = 0 \) (\( y_t = 1 \)). We converted clips to low- (high-) fidelity data by evaluating the low- (high-) neural network and computing the median anomaly score for that clip. As in Section 4.2, we used a Bernoulli model with \( \alpha_{HF} = 1 \) and \( \alpha_{LF} \) set to the low-fidelity model’s true positive rate relative to the high-fidelity model. For each machine, we randomly generated 500 datasets with changepoints and computed the MSE and \( L_1 \) distances for low-fidelity BOCD and for MF-BOCD with both random and information-based switching. We found that the information-based approach to switching had lower MSE and \( L_1 \) distance than BOCD using just low-fidelity data and had better performance than randomized switching on the first three machines (Table 1). An interesting negative result is that MF-BOCD does not do significantly better than random on machine 4. We hypothesize that this is due to the poor quality of the low-fidelity observation model, which has an AUC \(< 0.5 \) (Figure 8, Appendix). With these data, MF-BOCD is making hard decisions (argmax) with bad information. And in general, a randomized approach can sometimes do well (Figure 4). An interesting direction for future work would be to soften the decision rule via sampling, perhaps controlled by a temperature.

As in the CamVid experiments, we found that the total cost of decision-making was marginal; the neural network costs dominated the calculations (Table 1). Thus, MF-BOCD offers a useful way to trade off detection accuracy for computational savings.
5 DISCUSSION

We have extended Bayesian online changepoint detection to the multi-fidelity setting in which observations have associated fidelities and costs. We found that choosing the data fidelity based on maximal information rate with respect to the run-length posterior yields interpretable policies that lower computational costs while still maintaining good performance in terms of parameter and run-length posterior estimation. In simple models, decision-making is cheap relative to the cost of evaluating even tiny neural networks designed for commodity microcontrollers. Flops savings translate to energy savings [Banbury et al., 2020], which is crucial for resource-constrained applications.

While we focus on the online and resource-constrained setting, this framework could be extended to scenarios in which observations take a long time to compute, such as changepoint detection in protein-folding [Fan et al., 2015] or engineering design [Robinson et al., 2008]. In such settings, expensive approximations of the posterior predictive distribution or information gain may be tolerable, as well as retrospective smoothing of the run-length distributions.

Alternative decision rules should also be explored, as these will induce different policies. Gessner et al. [2020] discuss how any monotonic transformation of Equation (22) gives rise to the same policy because the global maximum is the same even if the value at that maximum is not. However, this is not necessarily true after dividing the decision rules by costs. Furthermore, a probabilistic decision rule might be useful in scenarios where the difference between low- and high-fidelity observation models is marginal.

ACKNOWLEDGEMENTS

We thank Paul Whatmough and Igor Fedorov for helpful conversations on ML for resource-constrained devices. B.E. Engelhardt and G.W. Gundersen received support from a grant from the Helmsley Trust, a grant from the NIH HTAN Research Program, NIH NHLBI R01 HL133218, and NSF CAREER AWD1005627. D. Cai was supported in part by a Google Ph.D. Fellowship in Machine Learning. R.P. Adams was supported in part by NSF IIS-2007278.

Table 1: Comparison between low-fidelity BOCD (LF), random switching (RN), and MF-BOCD (IG). Mean and two standard errors were computed over 500 randomly generated MIMII datasets with changepoints, using the method described in the text. Cost is in millions of flops. Bold numbers indicate statistically significant using 95% confidence intervals. %LF is the percentage of low-fidelity data used by both multi-fidelity models, RN and IG. The reported %LF is the average across all datasets.

<table>
<thead>
<tr>
<th></th>
<th>Machine 1</th>
<th>Machine 2</th>
<th>Machine 3</th>
<th>Machine 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>0.0060 (0.0004)</td>
<td>0.0195 (0.0008)</td>
<td>0.0347 (0.0012)</td>
<td>0.1743 (0.0042)</td>
</tr>
<tr>
<td>LF</td>
<td>0.0026 (0.0002)</td>
<td>0.0063 (0.0004)</td>
<td>0.0126 (0.0006)</td>
<td>0.0411 (0.0028)</td>
</tr>
<tr>
<td>IG</td>
<td>0.0029 (0.0002)</td>
<td>0.0065 (0.0004)</td>
<td>0.0121 (0.0006)</td>
<td>0.0393 (0.0030)</td>
</tr>
<tr>
<td>(L_1)</td>
<td>101.87 (3.28)</td>
<td>187.73 (3.61)</td>
<td>192.49 (4.02)</td>
<td>242.85 (4.63)</td>
</tr>
<tr>
<td>LF</td>
<td>57.61 (3.14)</td>
<td>97.79 (3.66)</td>
<td>132.06 (3.63)</td>
<td>178.86 (4.97)</td>
</tr>
<tr>
<td>RN</td>
<td>61.79 (3.02)</td>
<td>92.99 (3.65)</td>
<td>130.17 (3.88)</td>
<td>173.27 (4.95)</td>
</tr>
<tr>
<td>IG</td>
<td>61.79 (3.02)</td>
<td>92.99 (3.65)</td>
<td>130.17 (3.88)</td>
<td>173.27 (4.95)</td>
</tr>
<tr>
<td>Ops</td>
<td>100</td>
<td>16109.38</td>
<td>12867.76</td>
<td>13557.11</td>
</tr>
<tr>
<td>LF</td>
<td>14447.58</td>
<td>12866.40</td>
<td>13557.34</td>
<td></td>
</tr>
<tr>
<td>RN</td>
<td>14447.58</td>
<td>12866.40</td>
<td>13557.34</td>
<td></td>
</tr>
<tr>
<td>IG</td>
<td>24940</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
</tr>
<tr>
<td>HF</td>
<td>24940</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
</tr>
<tr>
<td>%LF</td>
<td>42</td>
<td>36</td>
<td>48</td>
<td>46</td>
</tr>
</tbody>
</table>

References


Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by


