# Discriminative Bayesian Filtering for the Semi-supervised Augmentation of Sequential Observation Data

Michael C. Burkhart(✉)

Adobe Inc., San José, USA
mburkhar@adobe.com

**Abstract.** We aim to construct a probabilistic classifier to predict a latent, time-dependent boolean label given an observed vector of measurements. Our training data consists of sequences of observations paired with a label for precisely one of the observations in each sequence. As an initial approach, we learn a baseline supervised classifier by training on the labeled observations alone, ignoring the unlabeled observations in each sequence. We then leverage this first classifier and the sequential structure of our data to build a second training set as follows: (1) we apply the first classifier to each unlabeled observation and then (2) we filter the resulting estimates to incorporate information from the labeled observations and create a much larger training set. We describe a Bayesian filtering framework that can be used to perform step 2 and show how a second classifier built using the latter, filtered training set can outperform the initial classifier.

At Adobe, our motivating application entails predicting customer segment membership from readily available proprietary features. We administer surveys to collect label data for our subscribers and then generate feature data for these customers at regular intervals around the survey time. While we can train a supervised classifier using paired feature and label data from the survey time alone, the availability of nearby feature data and the relative expensive of polling drive this semi-supervised approach. We perform an ablation study comparing both a baseline classifier and a likelihood-based augmentation approach to our proposed method and show how our method best improves predictive performance for an in-house classifier.

**Keywords:** Bayesian filtering · Discriminative modeling · Data augmentation · Semi-supervised learning · Machine learning · Learning from survey data

## 1 Problem Description and Notation

We aim to predict a binary-valued label of interest $Z_t \in \{0, 1\}$ from a vector $X_t \in \mathbb{R}^m$ of measurable features. We are provided a supervised dataset

$$\mathcal{D}_0 = \{(x_\tau^1, z_\tau^1), (x_\tau^2, z_\tau^2), \ldots, (x_\tau^n, z_\tau^n)\}$$


M. Paszynski et al. (Eds.): ICCS 2021, LNCS 12743, pp. 271–283, 2021.
https://doi.org/10.1007/978-3-030-77964-1_22

of $n$ labeled training pairs and an unsupervised dataset

$$\mathcal{D}_1 = \{x^1_{1:\tau-1}, x^1_{\tau+1:T}; x^2_{1:\tau-1}, x^2_{\tau+1:T}; \ldots ; x^n_{1:\tau-1}, x^n_{\tau+1:T}\}$$

of time-indexed feature data for each training instance in a contiguous period $1 \leq t \leq T$ surrounding $\tau$. We adopt the notation $x^i_{1:T} = (x^i_1, x^i_2, \ldots, x^i_T)$ for indexed sequences of data and let $\tau \in \{1, 2, \ldots, T\}$ denote the time for which each sequence $x^i_{1:T}$ of features has an associated label $z^i_\tau$. Strictly speaking, we allow this time $\tau$ to be different for each sequence, i.e. $\tau$ may depend on $i$. We suppress the superscript $i$ when describing calculations for a single, generic instance.

If we have reason to believe that the relationship between the observed features and latent labels is stationary, i.e. that $p(z_t|x_t)$ does not depend on the time $t$, then a natural first approach to solving this problem entails training a supervised classifier on the dataset $\mathcal{D}_0$. *Can the unlabeled sequences of observations in $\mathcal{D}_1$ help us to build a better classifier?* That question, central to the field of semi-supervised learning, motivates our work. We intend to incorporate information from $\mathcal{D}_1$ through a process of data augmentation. In this paper, we develop and validate a novel method for estimating labels for the augmentation process. We develop a discriminative Bayesian filtering framework that provides a principled way to incorporate information from the unlabeled observations with information from the known label provided for a different observation in the same sequence.

Our work focuses on creating new training pairs $(x^i_t, \hat{z}^i_t)$ where $x^i_t$ belongs to one of the sequences in $\mathcal{D}_1$ and $\hat{z}^i_t$ denotes an estimated label for that observation at that time. We combine two sources of knowledge to form our estimate: (1) the snapshot $x^i_t$ of feature data, to which we may apply our original model for $p(z_t|x_t)$ and (2) the ground-truth label $z^i_\tau$ that fixes instance $i$'s label at a nearby point in time, to which we may iteratively apply a latent state model. For each point in $\mathcal{D}_1$, we calculate the posterior probability $p(z_t|x_{\tau+1}, \ldots, x_t, z_\tau)$ when $t > \tau$ and $p(z_t|x_t, \ldots, x_{\tau-1}, z_\tau)$ when $t < \tau$. We then take estimates for the posterior that are almost certain (very near to zero or one), threshold them, and use them to form an augmented training set, paired with their corresponding feature-values. We use this larger set to train a second classifier and argue that it tends to have better predictive ability than both the first classifier and a classifier trained using only the first source of knowledge. See Fig. 1 for a visual comparison of these approaches.

*Outline.* The paper is organized as follows. In the next section, we introduce our filtering framework and describe how to form filtered estimates for a given sequence. In Sect. 3, we show how to use these filtered estimates to create an augmented training dataset. Then in Sect. 4, we compare our classifier trained using filtered data to both the baseline classifier and to an augmented classifier trained using pseudo-labeling [20], a common self-learning approach that augments the training set using estimates for the likelihood alone. We survey related work in Sect. 5 before concluding in Sect. 6.

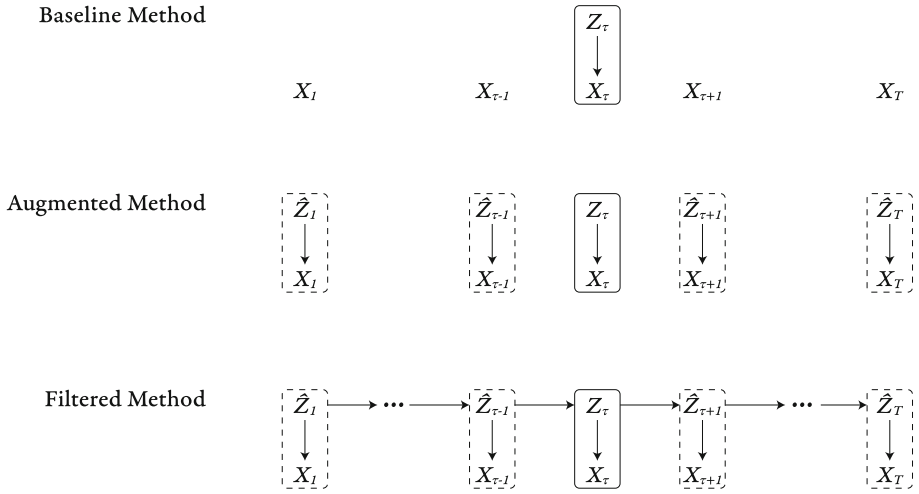


**Fig. 1.** *Schematic comparing the baseline, augmented, and filtered methods.* The baseline method uses the supervised set $\mathcal{D}_0$ alone, ignoring $\mathcal{D}_1$. The augmented method assigns a probabilistic estimate to each feature-point in $\mathcal{D}_1$ (a type of pseudo-labeling), increasing the available data by a factor of $T$ prior to thresholding. The filtered method additionally incorporates information from the ground truth label to improve its probabilistic estimates.

## 2 Filtering Methodology

In this section we focus on a single instance $i$ and describe a filtering process that produces predictions for each unlabeled member of the sequence $x_{1:T} = x^i_{1:T}$ of observations using a provided model $p(z_t|x_t)$ and the sequence's corresponding binary label $z_\tau = z^i_\tau$ for some time $\tau = \tau(i)$.

We view the labels and corresponding observations as belonging to a latent state space model. Letting $Z_{1:T} := Z_1, \ldots, Z_T$ denote random variables corresponding to the latent labels and $X_{1:T} := X_1, \ldots, X_T$ denote random variables corresponding to the observations, we model the relationship between these variables according to the Bayesian network:

$$\begin{array}{ccccccccc} Z_1 & \longrightarrow & \cdots & \longrightarrow & Z_{t-1} & \longrightarrow & Z_t & \longrightarrow & \cdots & \longrightarrow & Z_T \\ \downarrow & & & & \downarrow & & \downarrow & & & & \downarrow \\ X_1 & & & & X_{t-1} & & X_t & & & & X_T \end{array} \tag{1}$$

Using this framework, we aim to infer the predictive posterior distribution $p(z_t|x_{\tau+1:t}, z_\tau)$ for times $t > \tau$ and $p(z_t|x_{t:\tau-1}, z_\tau)$ for times $t < \tau$, where $Z_t$ is uncertain. To motivate this exercise, we hypothesize that augmented pairs $(x_t, \hat{z}_t)$ produced using the posterior will better assist in training than those produced using the likelihood, $p(z_t|x_t)$.

Traditional approaches to filtering specify a state model $p(z_t|z_{t-1})$ that relates the current state to the previous state and a measurement model $p(x_t|z_t)$ that relates the current observation to the current latent state. The posterior distribution of the hidden state given a sequence of measurements can then be calculated or approximated through a series recursive updates. See Chen [10] or Särkkä [30] for comprehensive surveys.

In the remainder of this section, we outline the details of our filtering framework and show how to calculate the posterior probability under our model. In the subsequent section, we outline how this filtering approach can be used for data augmentation.

### 2.1 Discriminative Measurement Model

As opposed to the distribution $p(x_t|z_t)$ that describes the outcomes of hundreds of measurements ($m \gg 1$) given a single boolean label, the distribution $p(z_t|x_t)$ that describes the likelihood of a label given a vector of measurements often proves much more tractable to learn effectively using off-the-shelf classifiers. *Consequently, we approximate the measurement model using* $p(z_t|x_t)$. We apply Bayes' rule and note that, up to a constant depending only on $x_t$, we may replace $p(x_t|z_t)$ with $p(z_t|x_t)/p(z_t)$. We further assume that this model is stationary, so that $p(z_t|x_t)$ is independent of $t$. This approach mirrors that of McCallum et al.'s Maximum Entropy Markov Model [22] and the more recent Discriminative Kalman Filter [5,6], as it relies on a discriminative approximation to the measurement model. (In particular, the model is no longer generative, following the terminology of Ng and Jordan [25].)

### 2.2 Reversible State Model

We specify the state model as as stationary Markov chain

$$\mathbb{P}(Z_t = 1|Z_{t-1} = 0) = \alpha_0, \tag{2a}$$
$$\mathbb{P}(Z_t = 1|Z_{t-1} = 1) = \alpha_1, \tag{2b}$$

for some $0 < \alpha_0, \alpha_1 < 1$ and $2 \leq t \leq T$. If we let

$$\beta = \alpha_0/(1 + \alpha_0 - \alpha_1), \tag{3}$$

it follows that if

$$\mathbb{P}(Z_{t_0} = 1) = \beta \tag{4}$$

at some time $t_0$, then $\mathbb{P}(Z_t = 1) = \beta$ for all $t$. Furthermore, this chain is reversible [11, sec. 6.5], and

$$\mathbb{P}(Z_t = 1|Z_{t+1} = 0) = (1 - \alpha_1)\beta/(1 - \beta) = \alpha_0, \tag{5a}$$
$$\mathbb{P}(Z_t = 1|Z_{t+1} = 1) = \alpha_1, \tag{5b}$$

for all $t$.

### 2.3 Filtering Forward in Time

We first describe how to filter forward in time. Starting with the ground-truth label at time $\tau$, we iteratively combine information from our estimate of the previous label (using the state model) with our estimate of the current label given the measurements at that time (using the measurement model). For $t > \tau$, we recursively calculate the posterior

$$\rho_{\tau:t} := \mathbb{P}(Z_t = 1 | X_{\tau+1:t} = x_{\tau+1:t}, Z_\tau = z_\tau) \tag{6}$$

in terms of the likelihood $p_t = \mathbb{P}(Z_t = 1 | X_t = x_t)$ and the previous posterior $\rho_{\tau:t-1}$ as

$$\begin{aligned}\rho_{\tau:t} &\propto \frac{\mathbb{P}(Z_t = 1 | X_t = x_t)}{\mathbb{P}(Z_t = 1)} \mathbb{P}(Z_t = 1 | X_{\tau+1:t-1} = x_{\tau+1:t-1}, Z_\tau = z_\tau) \\ &\propto \frac{p_t}{\mathbb{P}(Z_t = 1)} \big( \mathbb{P}(Z_t = 1 | Z_{t-1} = 1) \rho_{\tau:t-1} + \mathbb{P}(Z_t = 1 | Z_{t-1} = 0)(1 - \rho_{\tau:t-1}) \big)\end{aligned}$$

where we initialize $\rho_{\tau:\tau}$ as the observed point mass $p(Z_\tau = 1)$. It follows from our state model (2) and initialization (4) that

$$\rho_{\tau:t} \propto \frac{p_t}{\beta} \big( \alpha_1 \rho_{\tau:t-1} + \alpha_0 (1 - \rho_{\tau:t-1}) \big) \tag{7}$$

up to a constant depending on the observations alone. To relieve ourselves of that constant, we note that

$$\begin{aligned}1 - \rho_{\tau:t} &\propto \frac{\mathbb{P}(Z_t = 0 | X_t = x_t)}{\mathbb{P}(Z_t = 0)} \mathbb{P}(Z_t = 0 | X_{\tau+1:t-1} = x_{\tau+1:t-1}, Z_\tau = z_\tau) \\ &\propto \frac{1 - p_t}{\mathbb{P}(Z_t = 0)} \big( \mathbb{P}(Z_t = 0 | Z_{t-1} = 1) \rho_{\tau:t-1} + \mathbb{P}(Z_t = 0 | Z_{t-1} = 0)(1 - \rho_{\tau:t-1}) \big)\end{aligned}$$

which itself simplifies to

$$1 - \rho_{\tau:t} \propto \frac{1 - p_t}{1 - \beta} \big( (1 - \alpha_1) \rho_{\tau:t-1} + (1 - \alpha_0)(1 - \rho_{\tau:t-1}) \big). \tag{8}$$

Dividing the right-hand side of (7) by the sum of (7) and (8) cancels the constant of proportionality and yields $\rho_{\tau:t}$.

### 2.4 Filtering Backward in Time

Now, we describe how to filter backward in time. As before, we proceed recursively to calculate

$$\tilde{\rho}_{t:\tau} := \mathbb{P}(Z_t = 1 | X_{t:\tau-1} = x_{t:\tau-1}, Z_\tau = z_\tau) \tag{9}$$

for $t = \tau - 1, \tau - 2, \ldots, 1$ in terms of $\tilde{\rho}_{t+1:\tau}$ and the likelihood $p_t$:

$$\begin{aligned}\tilde{\rho}_{t:\tau} &\propto \frac{\mathbb{P}(Z_t = 1 | X_t = x_t)}{\mathbb{P}(Z_t = 1)} \mathbb{P}(Z_t = 1 | X_{t+1:\tau-1} = x_{t+1:\tau-1}, Z_\tau = z_\tau) \\ &\propto \frac{p_t}{\mathbb{P}(Z_t = 1)} \big( \mathbb{P}(Z_t = 1 | Z_{t+1} = 1) \tilde{\rho}_{t+1:\tau} + \mathbb{P}(Z_t = 1 | Z_{t+1} = 0)(1 - \tilde{\rho}_{t+1:\tau}) \big)\end{aligned}$$

where, analogously, $\tilde{\rho}_{\tau:\tau}$ is taken to be the observed point mass $p(Z_\tau = 1)$. Upon substituting the state model (2) and initialization (4), we have

$$\tilde{\rho}_{t:\tau} \propto \frac{p_t}{\beta}\big(\alpha_1 \tilde{\rho}_{t+1:\tau} + \alpha_0(1 - \tilde{\rho}_{t+1:\tau})\big). \tag{10}$$

To remove the constant of proportionality, we also calculate:

$$\begin{aligned} 1 - \tilde{\rho}_{t:\tau} &\propto \frac{\mathbb{P}(Z_t = 0 | X_t = x_t)}{\mathbb{P}(Z_t = 0)} \mathbb{P}(Z_t = 0 | X_{t+1:\tau-1} = x_{t+1:\tau-1}, Z_\tau = z_\tau) \\ &\propto \frac{1 - p_t}{\mathbb{P}(Z_t = 0)}\big(\mathbb{P}(Z_t = 0 | Z_{t+1} = 1)\tilde{\rho}_{t+1:\tau} + \mathbb{P}(Z_t = 0 | Z_{t+1} = 0)(1 - \tilde{\rho}_{t+1:\tau})\big) \end{aligned}$$

which simplifies to

$$1 - \tilde{\rho}_{t:\tau} \propto \frac{1 - p_t}{1 - \beta}\big((1 - \alpha_1)\tilde{\rho}_{t+1:\tau} + (1 - \alpha_0)(1 - \tilde{\rho}_{t+1:\tau})\big). \tag{11}$$

Dividing the right-hand side of (10) by the sum of (10) and (11) then yields our objective, $\tilde{\rho}_{t:\tau}$.

## 3 Augmentation Methodology

In this section, we describe our method for data augmentation, relying on the filtering framework developed in the previous section. Given $n$ sequences of measurements $x^i_{1:T}$ with corresponding labels $z^i_\tau \in \{0, 1\}$ for $\tau = \tau(i) \in \{1, \ldots, T\}$ and $1 \leq i \leq n$, we define the supervised dataset

$$\mathcal{D}_0 = \{(x^1_\tau, z^1_\tau), (x^2_\tau, z^2_\tau), \ldots, (x^n_\tau, z^n_\tau)\}$$

of features $x^i_\tau \in \mathbb{R}^m$ paired with their corresponding labels $z^i_\tau \in \{0, 1\}$, along with the unsupervised dataset containing the unlabeled portions of each sequence,

$$\mathcal{D}_1 = \{x^1_{1:\tau-1}, x^1_{\tau+1:T}; x^2_{1:\tau-1}, x^2_{\tau+1:T}; \ldots; x^n_{1:\tau-1}, x^n_{\tau+1:T}\}.$$

We augment our supervised dataset $\mathcal{D}_0$ with information from $\mathcal{D}_1$ as follows:

1. We first learn a supervised model $f : \mathbb{R}^m \to [0, 1]$ on $\mathcal{D}_0$ such that for inputs $x \in \mathbb{R}^m$,
$$f(x) \approx \mathbb{P}(Z_\tau = 1 | X_\tau = x). \tag{12}$$
We refer to this probabilistic classifier as the baseline model.
2. For each instance $i$, we apply the baseline model to each feature-point in $\mathcal{D}_1$ to form
$$\tilde{\mathcal{D}}_1 = \{(x^i_t, f(x^i_t))\}_{x^i_t \in \mathcal{D}_1},$$
as our stationarity assumption implies $f(x) \approx \mathbb{P}(Z_t = 1 | X_t = x)$ for all $x, t$.

3. For each $i$, we apply the filtering equations from the previous section, starting at the time point $\tau$, and filtering forward in time to calculate the posterior estimates $\rho^i_{\tau:t}$ for $t = \tau+1, \tau+2, \ldots, T$, and backward in time to determine $\tilde{\rho}^i_{t:\tau}$ for $t = \tau-1, \tau-2, \ldots, 1$. We form

$$\breve{\mathcal{D}}_1 = \{(x^i_t, \rho^i_{\tau:t})\}_{1\le i\le n, t\ge\tau} \cup \{(x^i_t, \tilde{\rho}^i_{t:\tau})\}_{1\le i\le n, t<\tau}$$

and threshold a subset of these points in the manner we will describe in Sect. 3.1.

We use a held-out validation dataset to select parameters $\alpha_0, \alpha_1$ from (2) for the underlying state model.* These control the propensity for an instance to maintain a label from one time step to the next. As $\mathbb{E}[Z_t] = \beta = \mathbb{E}[Z_\tau]$ can be approximated from the training data, once an optimal value for $\alpha_0$ has been selected, this value along with an empirical approximation for $\mathbb{E}[Z_\tau]$ can be used with (3) to select a good value for $\alpha_1$.

### 3.1 Thresholding to Create Binary Labels

As many classifiers (including the lightGBM and XGBoost models) expect binary (non-probabilistic) labels for training data, we threshold the filtered labels and form a binary-valued set $\breve{\mathcal{D}}'_1$ from the filtered set $\breve{\mathcal{D}}_1$ and lower and upper bounds $0 < b_L < b_U < 1$ as follows. For each point $(x, \rho) \in \breve{\mathcal{D}}_1$, if $\rho < b_L$, we add $(x, 0)$ to $\breve{\mathcal{D}}'_1$; if $b_L < \rho < b_U$, we discard the point; and if $b_U < \rho$, we add $(x, 1)$ to $\breve{\mathcal{D}}'_1$. This yields a thresholded, filtered training set $\breve{\mathcal{D}}'_1$ with binary-valued labels that contains $\mathcal{D}_0$ as a subset. The parameters $0 < b_L < b_U < 1$ can also be selected using validation (or cross-validation).

We summarize our approach using pseudo-code in Algorithm 1.

## 4 Ablation Study and Results

In this section, we describe how we applied these methods to real-life customer survey data at Adobe.

### 4.1 Data Provenance

We surveyed approximately ten thousand Adobe subscribers in October 2019 and a distinct set of approximately equal size in February 2020. Based on survey responses alone, we assigned classifications to each subscriber. For the purposes

* Given a set $\{\alpha^k_0\}_{k\in K}$ of candidate values for $\alpha_0$ and a set $\{\alpha^\ell_1\}_{\ell\in L}$ for $\alpha_1$, we select parameters via an exhaustive grid search as follows. For each $(k, \ell) \in K \times L$, we apply Algorithm 1 with $\alpha^k_0$ and $\alpha^\ell_1$ to the training set, train a classifier on the resulting filtered dataset, and then evaluate this classifier's predictive performance on the validation set (using AUC). Upon completion, we select the parameter values $\alpha^k_0$ and $\alpha^\ell_1$ that yield the most performant classifier.

**Data:** labeled dataset $\mathcal{D}_0$ and unlabeled dataset $\mathcal{D}_1$; parameters $0 < \alpha_0, \alpha_1 < 1$ and $0 < b_L < b_U < 1$ obtained from validation
**Result:** labeled binary-valued dataset $\breve{\mathcal{D}}_1'$ extending $\mathcal{D}_0$ that can be used for supervised learning
Train a supervised model $f : \mathbb{R}^m \to [0, 1]$ on $\mathcal{D}_0$ such that $f(x)$ approximates $\mathbb{P}(Z_t = 1 | X_t = x)$ for $x \in \mathbb{R}^m$;
Initialize $\breve{\mathcal{D}}_1' = \mathcal{D}_0$;
**for** $i = 1, \ldots, n$ **do** *#iterate over instances*
  *#filter forward from* $\tau$;
  let $\rho_{\tau:\tau} = z_\tau^i$ from $\mathcal{D}_0$;
  **for** $t = \tau + 1, \tau + 2, \ldots, T - 1, T$ **do**
    *#determine predictive posterior*;
    let $\rho_{\tau:t} = \pi_1/(\pi_0 + \pi_1)$ where
      $\pi_0 = (1 - f(x_t^i))\big((1 - \alpha_1)\rho_{\tau:t-1} + (1 - \alpha_0)(1 - \rho_{\tau:t-1})\big)/(1 - \beta)$ and
      $\pi_1 = f(x_t^i)\big(\alpha_1\rho_{\tau:t-1} + \alpha_0(1 - \rho_{\tau:t-1})\big)/\beta$;
    *#threshold*;
    **if** $\rho_{\tau:t} < b_L$ **then** add $(x_t^i, 0)$ to $\breve{\mathcal{D}}_1'$ ;
    **if** $\rho_{\tau:t} > b_U$ **then** add $(x_t^i, 1)$ to $\breve{\mathcal{D}}_1'$ ;
  **end**
  *#filter backward from* $\tau$;
  let $\tilde{\rho}_{\tau:\tau} = z_\tau^i$ from $\mathcal{D}_0$;
  **for** $t = \tau - 1, \tau - 2, \ldots, 2, 1$ **do**
    *#determine predictive posterior*;
    let $\tilde{\rho}_{t:\tau} = \pi_1/(\pi_0 + \pi_1)$ where
      $\pi_0 = (1 - f(x_t^i))\big((1 - \alpha_1)\tilde{\rho}_{t+1:\tau} + (1 - \alpha_0)(1 - \tilde{\rho}_{t+1:\tau})\big)/(1 - \beta)$ and
      $\pi_1 = f(x_t^i)\big(\alpha_1\tilde{\rho}_{t+1:\tau} + \alpha_0(1 - \tilde{\rho}_{t+1:\tau})\big)/\beta$;
    *#threshold*;
    **if** $\tilde{\rho}_{t:\tau} < b_L$ **then** add $(x_t^i, 0)$ to $\breve{\mathcal{D}}_1'$ ;
    **if** $\tilde{\rho}_{t:\tau} > b_U$ **then** add $(x_t^i, 1)$ to $\breve{\mathcal{D}}_1'$;
  **end**
**end**

**Algorithm 1:** Discriminative Bayesian Filtering for Data Augmentation

of testing this algorithm, we considered only a single survey category. We engineered hundreds of proprietary features for each user. We then trained a supervised classifier using the ground-truth survey data to predict segment membership given feature data. We generated features for the surveyed subscribers for each of the twelve months between April 2019 and March 2020, inclusive.

We split the surveyed subscribers randomly into seven partitions of approximately equal size. Each partition was further split into a training, validation, and test set at a 70%–15%–15% ratio, respectively. For each partition, we took the supervised training set $\mathcal{D}_0$ to be the training-set subscribers with paired features and ground-truth survey results at their respective survey times and the unsupervised set $\mathcal{D}_1$ to be the features of the training-set subscribers calculated during the year-long period surrounding their survey dates.

### 4.2 Ablation with Different Supervised Classification Methods

As our method works with any supervised classifier, we learn $f$ in (12) using a variety of different methods: a LightGBM classifier, an XGBoost classifier, and a feedforward neural network classifier. LightGBM [15] attempted to improve upon other gradient boosting machines [12] by performing gradient-based sampling for the estimation of information gain and by bundling infrequently co-occurring features. Our implementation used the default parameters, to avoid over-tuning. XGBoost [9], a predecessor to LightGBM, introduced a novel algorithm for handling sparse data and allowed for unprecedented scalability. Our implementation relied on the default parameters. The neural network we used comprised of a single hidden layer of 30 neurons with rectified linear activation [24] and $L^2$-regularization for network weights [17], trained with L-BFGS [21].

For each of the seven data partitions, we compare the predictive performance (on the held-out test set) of three different approaches to building a supervised classifier:

- For the *baseline* method, we take our classifier to be the original $f$ trained on the supervised dataset $\mathcal{D}_0$.
- For the *augmented* method, we apply the baseline model to $\mathcal{D}_1$ and threshold the results as described in Sect. 3.1 using thresholds selected for performance on the validation dataset; we then train a classifier on the subsequent dataset.
- For the *filtered* method, we train a classifier using the dataset obtained by applying Algorithm 1. Parameters $0 < \alpha_0, \alpha_1 < 1$ and $0 < b_L < b_U < 1$ were chosen to maximize predictive performance on the validation set.

We aim to isolate the effects of augmenting the training dataset with filtered, posterior-based estimates from the benefits obtained by simply employing likelihood-based estimates.

To illustrate this difference, consider the following example. Suppose we administer a survey in June to Alice, Bob, and ten thousand other subscribers. We train a classifier using the data from June and apply it to Alice and Bob's feature data for July to predict that Alice has a 90% chance of belonging to the segment of interest in July while Bob has a 1% chance. If our lower-bound threshold is $u_L = 2\%$ and our upper threshold is $u_B = 95\%$, then for the *augmented* training set, we would include only Bob's features for July, with a negative label. Suppose further that our ground truth labels in June indicate that Alice and Bob both belonged to the segment of interest that month. If subscribers have an 80% chance of maintaining a positive label from one month to the next ($\alpha_1 = 0.8$ in Eq. 2b), and the segment of interest includes 30% of the population ($\beta = 0.3$ in Eq. 3), then Alice has a 99% chance of belonging to the segment given both her June and July data, while Bob has a 9% chance (see eqs. (7) and (8)). Thus, with the same thresholds, our *filtered* training set would include only Alice's features for July, with a positive label.

We performed all numerical tests on a 15-in. 2018 MacBook Pro (2.6 GHz Intel Core i7 Processor; 16 GB 2400 MHz DDR4 Memory). We used Python 3.8.6 with the following packages (versioning in parentheses): lightgbm (3.0.0), numpy (1.18.5), pandas (1.1.4), scikit-learn (0.23.20), & xgboost (1.2.1).

### 4.3 Results

We measure model performance using AUC and report our results in Table 1. Mean performance increases with each successive approach (baseline to augmented to filtered methods) for each of the three types of classification methods used. For the LightGBM classifier, we have reason to prefer the filtered training method over the baseline (avg. +7.0% improvement in AUC; $p = 0.00787$; paired sample $t$-test with 6 degrees of freedom against a one-sided alternative) and over the augmented training method (avg. +5.7% improvement in AUC; $p = 0.0333$; same test).

**Table 1.** *Classification AUC (Area Under the receiver operating characteristic Curve) for each of the methods tested.* The mean column reports the average AUC over the 7 independent trials for each method.

| Classifier | Method | Trial 1 | Trial 2 | Trial 3 | Trial 4 | Trial 5 | Trial 6 | Trial 7 | Mean |
|---|---|---|---|---|---|---|---|---|---|
| LightGBM | Baseline | 0.611 | 0.564 | 0.698 | 0.564 | 0.645 | 0.636 | 0.588 | 0.615 |
| | Augmented | 0.611 | 0.536 | 0.689 | 0.525 | 0.720 | 0.611 | 0.687 | 0.626 |
| | Filtered | 0.630 | 0.582 | 0.704 | 0.620 | 0.705 | 0.668 | 0.690 | 0.657 |
| XGBoost | Baseline | 0.568 | 0.553 | 0.669 | 0.590 | 0.657 | 0.688 | 0.621 | 0.621 |
| | Augmented | 0.555 | 0.564 | 0.677 | 0.601 | 0.654 | 0.663 | 0.652 | 0.624 |
| | Filtered | 0.664 | 0.580 | 0.619 | 0.600 | 0.696 | 0.662 | 0.645 | 0.638 |
| Feedforward NN | Baseline | 0.556 | 0.520 | 0.517 | 0.485 | 0.621 | 0.581 | 0.514 | 0.542 |
| | Augmented | 0.591 | 0.551 | 0.511 | 0.458 | 0.634 | 0.558 | 0.570 | 0.553 |
| | Filtered | 0.589 | 0.554 | 0.539 | 0.460 | 0.640 | 0.565 | 0.595 | 0.563 |

## 5 Related Work

We now describe how our proposed algorithm for data augmentation relates to both filtering and semi-supervised learning.

### 5.1 Discriminative Bayesian Filtering

State-space models having the graphical form (1) relate an unobserved Markovian sequence $(Z_1, Z_2, \dots)$ of interest to a series of observed measurements $(X_1, X_2, \dots)$, where each measurement depends only on the current hidden state. Also known as hidden Markov models, they have been extensively studied due to their wide range of applications. Discriminative variants, including Maximum Entropy Markov Models [22] and Conditional Random Fields [18], allow one to use $p(z_t|x_t)$ instead of $p(x_t|z_t)$ for inference. Moving from a generative to a discriminative approach allows one to more directly consider the distribution of states given observations, at the cost of no longer learning the joint distribution of the hidden and observed variables. (In particular, after training such a model, one cannot sample observations.) Applications include human motion tracking [16,33] and neural modeling [2–4].

### 5.2 Data Augmentation and Semi-supervised Learning

Given a small set of labeled training data and an additional set of unlabeled points, semi-supervised methods attempt to leverage the location of the unlabeled training points to learn a better classifier than could be obtained from the labeled training set alone [8,38]. For example, graph-based approaches use pairwise similarities between labeled and unlabeled feature-points to construct a graph, and then let labeled points pass their labels to their unlabeled neighbors. In the sense that our filtering process passes information along the Bayesian graph (1), our method is similar in spirit to graph-based methods—like Label Propagation [39] and Label Spreading [37]—though our graph derives from the natural temporal relationship of our measurements and not from the locations of the feature-points.

Self-learning refers to the practice of using the predictions from one classifier in order to train a second classifier [31,34,35]. For example, Pseudo-labeling [20] assigns predicted labels to unlabeled features and adds them to a supervised training set, in an approach equivalent to minimum entropy regularization [13]. Recent work in deep learning elaborates heavily on this idea, where latent representations gained from intermediate network layers play a crucial role [7,14,19,28,29,32,40]. A common failure mode for self-learners in general entails inadvertently misclassifying points and then propagating these erroneous labels to their unlabeled neighbors. Recent proposals to reduce this so-called confirmation bias mostly focus on deep neural networks [1,26,36].

## 6 Conclusions and Directions for Future Research

In this paper, we considered a semi-supervised learning problem involving sequential observation data and showed how filtering could be employed for data augmentation. We described a method that leverages the predictive posterior to augment the training dataset and compared it to a standard pseudo-labeling approach that performs augmentation using likelihood-based estimates. We then showed how classifiers trained on the filtered dataset can outperform those trained using pseudo-labeling.

The particular problem we considered consisted of time-delineated sequences of features, each coupled with a single time-dependent label. This problem corresponds to a relatively basic case, where each subscriber is surveyed at a single point in time. One can imagine having ground truth labels for a subscriber or instance at multiple points in time. In such a case, Bayesian smoothing may be applied to determine the predictive posterior for the points in time between the two labels. For more complex relationships between observations, belief propagation [27] or expectation propagation [23] may be worth exploring.

**Acknowledgements.** The author would like to thank his manager Binjie Lai, her manager Xiang Wu, and his coworkers at Adobe, especially Eunyee Koh for performing an internal review. The author is also grateful to the anonymous reviewers for their thoughtful feedback and to his former advisor Matthew T. Harrison for inspiring this discriminative filtering approach.

## References


1. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: International Joint Conference on Neural Networks, vol. 3, pp. 189–194 (2020)
2. Batty, E., et al.: Behavenet: nonlinear embedding and Bayesian neural decoding of behavioral videos. In: Advances in Neural Information Processing Systems, pp. 15706–15717 (2019)
3. Brandman, D.M., et al.: Rapid calibration of an intracortical brain-computer interface for people with tetraplegia. J. Neural Eng. **15**(2), 026007 (2018)
4. Brandman, D.M., Burkhart, M.C., Kelemen, J., Franco, B., Harrison, M.T., Hochberg, L.R.: Robust closed-loop control of a cursor in a person with tetraplegia using Gaussian process regression. Neural Comput. **30**(11), 2986–3008 (2018)
5. Burkhart, M.C.: A discriminative approach to bayesian filtering with applications to human neural decoding. Ph.D. thesis, Brown University, Division of Applied Mathematics, Providence, U.S.A. (2019)
6. Burkhart, M.C., Brandman, D.M., Franco, B., Hochberg, L.R., Harrison, M.T.: The discriminative Kalman filter for Bayesian filtering with nonlinear and non-gaussian observation models. Neural Comput. **32**(5), 969–1017 (2020)
7. Burkhart, M.C., Shan, K.: Deep low-density separation for semi-supervised classification. In: Krzhizhanovskaya, V.V., et al. (eds.) ICCS 2020. LNCS, vol. 12139, pp. 297–311. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50420-5_22
8. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press, Cambridge (2006)
9. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
10. Chen, Z.: Bayesian filtering: from Kalman filters to particle filters, and beyond. Technical report, McMaster U (2003)
11. Durrett, R.: Probability: Theory and Examples. Cambridge University Press, Cambridge (2010)
12. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Statist. **29**(5), 1189–1232 (2001)
13. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in Neural Information Processing Systems, pp. 529–536 (2004)
14. Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: Conference on Computer Vision and Pattern Recognition (2019)
15. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, pp. 3146–3154 (2017)
16. Kim, M., Pavlovic, V.: Discriminative learning for dynamic state prediction. IEEE Trans. Pattern Anal. Mach. Intell. **31**(10), 1847–1861 (2009)
17. Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. In: Advances in Neural Information Processing Systems, pp. 950–957 (1991)
18. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning (2001)
19. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: International Conference on Learning Representations (2017)
20. Lee, D.H.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: ICML Workshop on Challenges in Representation Learning (2013)

21. Liu, D.C., Nocedal, J.: On the limited memory method for large scale optimization. Math. Program. **45**(3), 503–528 (1989)
22. McCallum, A., Freitag, D., Pereira, F.: Maximum entropy Markov models for information extraction and segmentation. In: International Conference on Machine Learning, pp. 591–598 (2000)
23. Minka, T.P.: Expectation propagation for approximate Bayesian inference. In: Uncertainty in Artificial Intelligence (2001)
24. Nair, V., Hinton, G.: Rectified linear units improve restricted Boltzmann machines (2010)
25. Ng, A., Jordan, M.: On discriminative vs. generative classifiers: a comparison of logistic regression and Naive Bayes. In: Advances in Neural Information Processing Systems, vol. 14, pp. 841–848 (2002)
26. Oliver, A., Odena, A., Raffel, C.A., Cubuk, E.D., Goodfellow, I.: Realistic evaluation of deep semi-supervised learning algorithms. In: Advances in Neural Information Processing Systems, pp. 3235–3246 (2018)
27. Pearl, J.: Reverend Bayes on inference engines: a distributed hierarchical approach. In: Proceedings of Association for the Advancement of Artificial Intelligence, pp. 133–136 (1982)
28. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems, pp. 3546–3554 (2015)
29. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: Advances in Neural Information Processing Systems, pp. 1163–1171 (2016)
30. Särkkä, S.: Bayesian Filtering and Smoothing. Cambridge University Press, Cambridge (2013)
31. Scudder III, H.J.: Probability of error for some adaptive pattern-recognition machines. IEEE Trans. Inf. Theory **11**(3), 363–371 (1965)
32. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems, pp. 1195–1204 (2017)
33. Taycher, L., Shakhnarovich, G., Demirdjian, D., Darrell, T.: Conditional random people: tracking humans with CRFs and grid filters. In: Computer Vision and Pattern Recognition (2006)
34. Whitney, M., Sarkar, A.: Bootstrapping via graph propagation. In: Proceedings of Association for Computational Linguistics, vol. 1, pp. 620–628 (2012)
35. Yarkowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of Association for Computational Linguistics, pp. 189–196 (1995)
36. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization. In: International Conference on Learning Representations (2018)
37. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems, pp. 321–328 (2004)
38. Zhu, X.: Semi-supervised learning literature survey. Technical report, TR 1530, U. Wisconsin-Madison (2005)
39. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107, Carnegie Mellon University (2002)
40. Zhuang, C., Ding, X., Murli, D., Yamins, D.: Local label propagation for large-scale semi-supervised learning (2019). arXiv:1905.11581