ASAC: Active Sensing using Actor-Critic models

Jinsung Yoon
Electrical and Computer Engineering Department
University of California, Los Angeles, California, United States
jsyoon0823@g.ucla.edu

James Jordon
Department of Engineering Science, University of Oxford, Oxford, United Kingdom
james.jordon@wolfson.ox.ac.uk

Mihaela van der Schaar
University of Cambridge, Cambridge, United Kingdom
University of California, Los Angeles, California, United States
mihaela@ee.ucla.edu

Abstract

Deciding *what* and *when* to observe is critical when making observations is costly. In a medical setting where observations can be made *sequentially*, making these observations (or not) should be an active choice. We refer to this as the active sensing problem. In this paper, we propose a novel deep learning framework, which we call ASAC (Active Sensing using Actor-Critic models) to address this problem. ASAC consists of two networks: a selector network and a predictor network. The selector network uses previously selected observations to determine what should be observed in the future. The predictor network uses the observations selected by the selector network to predict a label, providing feedback to the selector network (well-selected variables should be predictive of the label). The goal of the selector network is then to select variables that balance the cost of observing the selected variables with their predictive power; we wish to preserve the conditional label distribution. During training, we use the actor-critic models to allow the loss of the selector to be "back-propagated" through the sampling process. The selector network "acts" by selecting future observations to make. The predictor network acts as a "critic" by feeding predictive errors for the selected variables back to the selector network. In our experiments, we show that ASAC significantly outperforms state-of-the-arts in two real-world medical datasets.

1. Introduction

In many medical settings, making observations is costly Weinstein et al. (1996). For example, performing lab tests on a patient incurs a cost, both financially as well as causing fatigue to the patient Koch et al. (2009); Kumwilaisak et al. (2008). In such settings, the decision to observe is important. This decision involves a trade-off between the value of the information obtained from the observation and the cost of making the observation. This problem presents itself when the data can be observed sequentially, so that we can observe a particular measurement before deciding which other measurements to observe. This problem

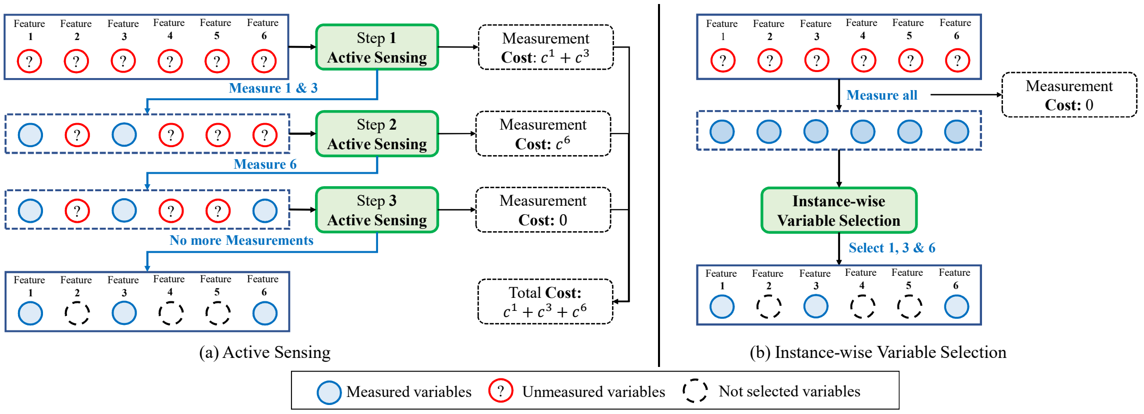

Figure 1: Comparison of Active Sensing and Instance-wise Variable selection in the static setting.

presents itself in both static and in time-series settings, with the key difference being that in the time-series setting, the values for a given stream[1] will change over time and thus we may wish to re-measure this, whereas in the static setting we know that once we observe a stream, we know its (fixed) value.

Genetic tests, for example, will have the same outcome whether we perform them now or later. As such, it may be advantageous to perform some tests, observe the results, and then decide on further tests to perform based on the results of the first Billings et al. (1992). Because the outcome of the tests will not change over time (we are in a *static setting*), there is no need to perform the tests we have already performed again and also no "worry" that we might miss something by not measuring it now (we can always go back and measure it later).

On the other hand, in an Intensive Care Unit (ICU) setting Ezzie et al. (2007) where important lab tests are being repeated and the results are always changing, we can no longer ignore a stream once it has been measured (its value may have changed since the last time we measured it) and moreover if we decide *not* to measure something, then we have missed our chance to measure it in that particular instant (we cannot go back and measure its value in the past). We can, however, still use past observations in determining what to measure next.

We refer to the problem of deciding what to observe in the future based on the measurements observed so far as *active sensing* Yu et al. (2009); Alaa and van der Schaar (2016). This problem presents itself in many healthcare applications Alaa and van der Schaar (2016); Schroeder et al. (2010). We formalize the problem of active sensing as a sequential decision making process in which, at each step, we select variables to measure based on all previously selected variables. When selecting variables, we wish to select those which are most predictive of the label, while also minimising cost.

This formulation of active sensing is related to instance-wise variable selection frameworks such as Yoon et al. (2019); Chen et al. (2018). In instance-wise variable selection, the goal is

[1] We use the term stream to refer to both the sequential values of a time-series variable and the single value of a static variable interchangeably.

to find a minimal subset of variables such that the conditional label distribution is preserved. However, in instance-wise variable selection, all of the variables should be measured *before* making the decision of which to select. In such settings, the goal is to efficiently *summarize* the information present in the entire feature vector in a lower dimensional feature vector. This is typically because the costly part there is not in observing the value of a variable but rather in presenting the value of the variable. In the active sensing framework, the cost has been shifted from presenting the information to measuring it and as such features that are not selected are not measured. Moreover, in the static setting of instance-wise variable selection, only a single selection is made, whereas for active sensing, both in the static and time-series settings, a sequence of selections is made. Fig. 1 illustrates these differences between active sensing and instance-wise variable selection. In the Supplementary Materials, we also illustrate these differences in the time-series setting.

Technical Significance In this paper, we propose ASAC (Active Sensing using Actor-Critic models), an algorithm capable of addressing active sensing in both static and time-series settings. ASAC consists of two networks: a selector network and a predictor network. The selector network uses previously selected features to determine which streams to observe next. The predictor network uses the selected features to predict a label. The networks are trained to minimize a Kullback-Leibler divergence between the conditional label distribution given all features and the conditional label distribution given only the selected features (thus ensuring that the selected features are as predictive of the label as all the features). Cost is introduced by adding a penalty term to the loss. We draw on actor-critic methodology Konda and Tsitsiklis (2000) to allow "back-propagation" through the sampling process of the selector network. We model each network using LSTMs Hochreiter and Schmidhuber (1997) to deal with sequential inputs and outputs, though any sequential model (e.g. temporal convolutions Van Den Oord et al. (2016)) could be used.

In our experiments, we demonstrate the efficacy of ASAC in a variety of scenarios using synthetic data. Then, using two real-world medical datasets (ADNI Petersen et al. (2010) and MIMIC-III Johnson et al. (2016)) we show that ASAC significantly outperforms the existing state-of-the-art methods.

Clinical Relevance: In medicine, making observations is usually costly and there is a clear trade-off between cost of measurements (e.g. MRI) and the value of observations (e.g. understanding the patient states based on the observation) Heywang-Köbrunner et al. (1997). Therefore, ASAC can do a critical role in clinical decision support that provides advice which observation should be measured and when. The proposed active sensing framework (ASAC) can be widely applied in various medical settings.

For instance, in an Intensive Care Unit (ICU) setting, there are more than 100 possible measurements on vital signs and lab tests Ezzie et al. (2007) and of course not all the possible tests are necessary for the entire patients Roberts et al. (1993) [2]. Among these extensive combinations of the measurements (and their timings), the proposed methods can provide advice which lab tests and vital signs should be measured and when based on the patient states estimated by the previously measured patients' observations. In breast cancer screening setting, there are various methods to screen the breast cancer such as mammogram, ultrasound, MRI and biopsy Saslow et al. (2007). Of course, not all the

[2] The missing (unmeasured) data can be imputed by imputation methods such as GAIN Yoon et al. (2018a)

patients need to be screened by all of the above methods Force et al. (2009). Most patients only need mammogram to screen the breast cancer and only the subset of the patient needs further screening tools Gøtzsche and Jørgensen (2013). The proposed model can provide the advice in this case as well that which patient needs additional screening examination based on the previous screening results (e.g. mammogram results).

The proposed methods try to minimize the observation costs without minimum information loss by not observing some measurements. We verify the proposed model in Intensive Care Unit (ICU) setting using MIMIC-III dataset and chronic disease setting using ADNI dataset.

1.1. Related Works

This paper draws motivation from existing instance-wise variable selection frameworks such as L2X Chen et al. (2018), LIME Ribeiro et al. (2016), Shapley Lundberg and Lee (2017), DeepLIFT Shrikumar et al. (2017) and in particular, Instance-wise variable selection (INVASE) Yoon et al. (2019). As noted above, a key difference between instance-wise variable selection and active sensing is in *what* is measured *before* making the selection. In addition, each of these works formalize the problem only in the static setting (where there are no temporal features). The applications for this problem are restricted to model interpretation and the models cannot be extended to the active sensing framework.

Deep Sensing Yoon et al. (2018b) is the work most closely related to ours. Like us, they attempt to solve the active sensing problem using deep learning, especially RNNs. The Deep Sensing framework involves learning 3 different networks: an interpolation network, a prediction network and an error estimation network. Each network is separately optimized for its own objective and then combined together after training to be used for active sensing. On the other hand, ASAC jointly optimizes the selector and predictor networks, *both* for the objective of active sensing, doing so by leveraging ideas from actor-critic methods Konda and Tsitsiklis (2000). Furthermore, Deep Sensing treats each feature independently, deciding what to measure by looking at the affect of a single feature on the label *in isolation*. ASAC, on the other hand, jointly estimates the effect of multiple features on the label prediction. This is critical when the features are highly correlated and also when the cost of measuring one feature differs significantly from measuring another noisier correlated feature. In the experiments, we show that our framework significantly outperforms the state-of-the-art in all settings. Due to space limitations, further details of other related works Bahdanau et al. (2014); Vaswani et al. (2017); Xu et al. (2015); Yu et al. (2009); Alaa and van der Schaar (2016) can be found in the Supplementary Materials.

2. Problem Formulation

In this section, we first describe the active sensing problem in the static setting, and then explain the differences in the time-series setting.

2.1. Static Setting

Let $\mathcal{X} = \mathcal{X}^1 \times ... \times \mathcal{X}^d$ be a d-dimensional feature space and $\mathcal{Y}$ be a label space (either $\mathbb{R}$ for regression problems or $\{1, 2, ..., C\}$ for multi-class classification problems with C classes).

We consider random variables $\mathbf{X} \in \mathcal{X}$, $Y \in \mathcal{Y}$ with some joint distribution p (and marginal distributions p_X and p_Y). For each feature, we assume that there is some cost, c^i, where $i = 1, ..., d$, associated with measuring the i-th feature. The cost vector is denoted as $\mathbf{c} = (c^1, ..., c^d)$.

A sensing decision is a vector $\mathbf{s} = (s^1, ..., s^d) \in \{0, 1\}^d$ where $s^i = 1$ corresponds to observing i-th feature. Let $*$ be any point not in $\mathcal{X}^1, ..., \mathcal{X}^d$. For any sensing vector $\mathbf{s}$ and any feature vector $\mathbf{x} = (x^1, ..., x^d) \in \mathcal{X}$ let $\mathbf{x}(\mathbf{s})$ be the vector obtained by

$$x(\mathbf{s})^i = \begin{cases} x^i & \text{if } s^i = 1 \\ * & \text{if } s^i = 0 \end{cases} \tag{1}$$

We refer to $\mathbf{x}(\mathbf{s})$ as the observed feature vector. In the static setting, we define a sensing decision sequence as $(\mathbf{s}_1, ..., \mathbf{s}_m)$ where each $\mathbf{s}_j$ sensing decision and we require that if $s^i_{j-1} = 1$, then $s^i_j = 1$ so that the sensing decisions form a nested sequence (this is simply so that $\mathbf{s}^j$ describes fully which features have already been measured at step j) and each $\mathbf{s}_j = \mathbf{s}_j(\mathbf{x}(\mathbf{s}_{j-1}))$ is allowed to depend on $\mathbf{x}(\mathbf{s}_{j-1})$.

Our goal, then, is to find a sensing decision sequence $(\mathbf{s}_1, ..., \mathbf{s}_m)$ that minimizes the total cost of measuring the chosen variables (i.e. $\mathbf{c}^T \mathbf{s}_m = \sum_{i=1}^d c^i \times s^i_m$) subject to the conditional distribution of Y given $\mathbf{X}$ being equal to the conditional distribution of Y given $\mathbf{X}(\mathbf{s}_m)$. That is, we wish to select variables that still allow us to predict Y as well as if we had measured everything, and among the sets of variables that do this, we wish to find the set with minimal measuring cost.

2.2. Time-series setting

In the time-series setting, a few modifications need to be made to the problem formulation given in Section 2.1. Instead of considering simple random variables, we now consider an indexed family (or sequence) of these random variables $\mathbf{X} = (\mathbf{X}_t)_{t \in \mathcal{T}}$ and $(Y_t)_{t \in \mathcal{T}}$ where t is an index in some time indexing set $\mathcal{T}$ with $\mathcal{T}$ being some bounded subset of either $\mathbb{R}$ or $\mathbb{N}$. In our case we focus on the discrete setting where $\mathcal{T} = \{1, ..., T\} \subset \mathbb{N}$ where T is some *random* stopping time (whose distribution we absorb into p), with our random processes assumed to be regularly sampled.

In contrast to the static setting, a sensing decision sequence now no longer requires that if $s^i_{t-1} = 1$, then $s^i_t = 1$, since now the values for each component of the process may vary between decisions and so will need to be remeasured if selected again (thus incurring a new cost). In addition, each sensing decision is allowed to depend on *all* observations made so far, that is $\mathbf{s}_t = \mathbf{s}_t(\mathbf{x}(\mathbf{s}_1), ..., \mathbf{x}(\mathbf{s}_{t-1}))$.[3] We denote $\mathbf{s}_{\leq t} = (\mathbf{s}_1, ..., \mathbf{s}_t)$, $\mathbf{x}_{\leq t} = (\mathbf{x}_1, ..., \mathbf{x}_t)$ and $\mathbf{x}(\mathbf{s}_{\leq t}) = (\mathbf{x}_1(\mathbf{s}_1), ..., \mathbf{x}_t(\mathbf{s}_t))$ to simplify notation.

In addition, we can extend this formulation further by allowing measurement delays to be included. Now that we have incorporated a time element, it also becomes natural that some features will take more or less time to measure than others (for example blood cultures can take up to one week to perform). To incorporate this into our formulation, we define a measurement time vector $\tau = (\tau_1, ..., \tau_d) \in \mathcal{T}^d$ which indicates the length of time it takes

[3] This is actually no different to the static setting where $\mathbf{x}(\mathbf{s}_j)$ contains all information found in $\mathbf{x}(\mathbf{s}_1), ..., \mathbf{x}(\mathbf{s}_{j-1})$.

to measure each feature. Then in this setting, our "current" feature vector, $\mathbf{x}_t(\mathbf{s}_{\leq t})$, now depends on all[4] selections made in the past (i.e. on $\mathbf{s}_{\leq t}$ rather than just $\mathbf{s}_t$) and is defined by

$$x_t(\mathbf{s}_{\leq t})^i = \begin{cases} x^i_{t-\tau_i} & \text{if } s^i_{t-\tau_i} = 1 \\ * & \text{if } s^i_{t-\tau_i} = 0 \end{cases} \tag{2}$$

so that if feature i was selected τ_i steps ago, then its value appears now in the current set of measured values. In this setting, we also write $\mathbf{x}(\mathbf{s}_{\leq t}) = (\mathbf{x}_1(\mathbf{s}_{\leq 1}), ..., \mathbf{x}_t(\mathbf{s}_{\leq t}))$.

The goal here is as in the static setting, where the total cost is now $\sum_{t=1}^T \mathbf{c}^T\mathbf{s}_t = \sum_{t=1}^T \sum_{i=1}^d c^i \times s^i_t$ and the conditional distribution constraint requires that Y_t given $\mathbf{X}_{\leq t}$ has the same distribution as Y_t given $\mathbf{X}(\mathbf{s}_{\leq t})$ for all $t \in \{1, ..., T\}$.

A time-series dataset, which we denote by $\mathcal{D}$, consists of N patient observations, assumed i.i.d. according to p so that $\mathcal{D} = \{(\mathbf{x}_{t,i}, y_{t,i})_{t=1}^{T_i}\}_{i=1}^N$ where $(\mathbf{x}_{t,i}, y_{t,i})_{t=1}^{T_i}$ is the stream corresponding to patient i of (random) length T_i.

In the remainder of the paper, the more general time-series setting will be used by default. When reading the rest of the paper, keep in mind that the discussion also applies to the static setting.

2.3. Optimization problem

Based on the above problem formulations, the optimization problem can be determined as follows.

$$\begin{aligned} \min_{\mathbf{s}_1,...,\mathbf{s}_T} \quad & \sum_{t=1}^T \mathbb{E}_{\mathbf{x}\sim p_X}\Big[\mathbf{c}^T\mathbf{s}_t\Big] \\ \text{s.t.} \quad & (Y_t|\mathbf{X}_{\leq t} = \mathbf{x}_{\leq t}) \overset{d}{=} (Y_t|\mathbf{X}(\mathbf{s}_{\leq t}) = \mathbf{x}(\mathbf{s}_{\leq t})) \text{ for all } t \in \{1, 2, ..., T\} \end{aligned} \tag{3}$$

In order to find a suitable (tractable) sensing decision sequence, we transform the distributional constraint into a soft constraint using the Kullback-Leibler (KL) divergence. To do this, we consider the problem of minimizing the KL divergence between the two conditional distributions with an added cost penalty term. The objective function we aim to minimize (with respect to the sensing decision sequence) is then

$$\sum_{t=1}^T \mathbb{E}_{\mathbf{x}\sim p_X}\Big[\big[KL((Y_t|\mathbf{X}_{\leq t} = \mathbf{x}_{\leq t})||(Y_t|\mathbf{X}(\mathbf{s}_{\leq t}) = \mathbf{x}(\mathbf{s}_{\leq t})))\big] + \lambda\mathbf{c}^T\mathbf{s}_t\Big] \tag{4}$$

where $\lambda \geq 0$ is a hyper-parameter that trades-off between the constraint (KL term) and the objective (cost term).

We can rewrite the KL divergence term as

$$\begin{aligned} & KL((Y_t|\mathbf{X}_{\leq t} = \mathbf{x}_{\leq t})||(Y_t|\mathbf{X}(\mathbf{s}_{\leq t}) = \mathbf{x}(\mathbf{s}_{\leq t}))) \\ = & \int_{\mathcal{Y}} p_Y(y|\mathbf{x}_{\leq t})\Big[\log(p_Y(y|\mathbf{x}_{\leq t})) - \log(p_Y(y|\mathbf{x}(\mathbf{s}_{\leq t})))\Big]dy \end{aligned}$$

[4] In fact, it depends only on $\mathbf{s}_u$ for $u \in \{t - \tau_i : i = 1, ..., d\}$, i.e. the times in the past in which measurements were "started" and whose results would be reported now.

and we note that $\log(p_Y(y|\mathbf{x}_{\leq t}))$ is independent of the sensing decision sequence $\mathbf{s}_{\leq t}$. We can therefore define an equivalent loss, $l(\mathbf{x}_{\leq t}, \mathbf{s}_{\leq t})$, as follows

$$l(\mathbf{x}_{\leq t}, \mathbf{s}_{\leq t}) = \int_{\mathcal{Y}} p_Y(y|\mathbf{x}_{\leq t})\Big[-\log(p_Y(y|\mathbf{x}(\mathbf{s}_{\leq t})))\Big]dy. \tag{5}$$

Then, the new optimization problem is defined as

$$\min_{\mathbf{s}_1,\dots,\mathbf{s}_T} \quad \sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}\sim p_X}\Big[l(\mathbf{x}_{\leq t}, \mathbf{s}_{\leq t}) + \lambda \mathbf{c}^T \mathbf{s}_t\Big]. \tag{6}$$

3. Proposed model

In order to solve the optimization problem given in equation (6), we first need to estimate the unknown density function: $p_Y(\cdot|\mathbf{x}(\mathbf{s}_{\leq t}))$. To do this, we introduce a predictor function $f_\phi : \prod_{i=1}^{t}(\mathcal{X} \times \{0,1\}^d) \to \mathcal{Y}$ parameterized by ϕ which will be trained to predict y given all (selected) observations up until time t (i.e. $\mathbf{x}(\mathbf{s}_{\leq t})$ and $\mathbf{s}_{\leq t}$).

In order to perform sensing decisions (which are binary), we introduce a selector function $f_\theta : \prod_{i=1}^{t}(\mathcal{X} \times \{0,1\}^d) \to [0,1]^d$ parameterized by θ that will output continuous values in $[0,1]^d$ which will be treated as probabilities to then be sampled from to create an output in $\{0,1\}^d$. The selection mechanism is therefore *probabilistic* in nature, and as such our optimization problem in (6) now needs to include an expectation over the sensing decision sequence $\mathbf{s}_{\leq T}$. This selector function f_θ will take measurements up until time t as input and then output probabilities from which the decision sequence for time $t+1$ will be sampled. In order to "back-propagate" through the sampling process, we draw on actor-critic models Konda and Tsitsiklis (2000) to derive the gradient of our selector function loss in Section 3.2.

These two networks will be trained iteratively. This is important because both functions influence each other. The predictor function directly determines the loss of the selector function and thus has a direct impact on the training of the selector function. The selector function, on the other hand, has the more subtle effect of changing the distribution over which the predictor function needs to perform well. As the selector function is updated, the input distribution for the predictor network changes, and it is important that the predictor function performs well on the new distribution. As such, the predictor network needs to be updated after each selector function update (and vice-versa).

3.1. Predictor function

The predictor function is trained to minimize a prediction loss

$$\mathcal{L}(\phi) = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}\sim p_X}[l_t(\phi)] \tag{7}$$

where for C-class classification we have the standard cross-entropy loss given by

$$l_t(\phi) = -\sum_{i=1}^{C} y_t^i \log(f_\phi^i(\mathbf{x}(\mathbf{s}_{\leq t}), \mathbf{s}_{\leq t})) \tag{8}$$

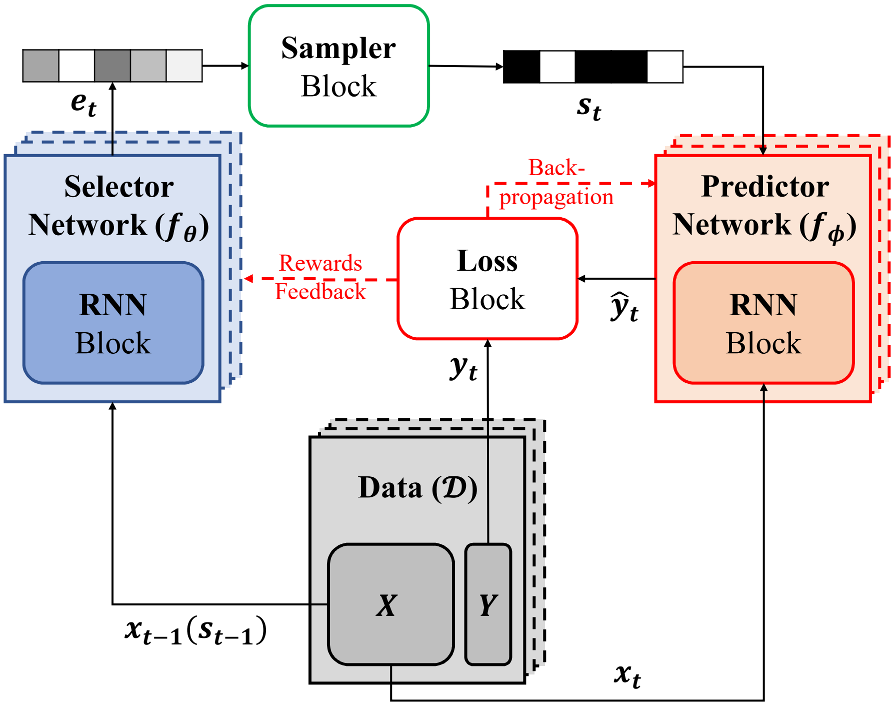

Figure 2: Block diagram of ASAC.

and for regression we have the standard mean-squared error loss given by

$$l_t(\phi) = (y_t - f_\phi(\mathbf{x}(\mathbf{s}_{\leq t}), \mathbf{s}_{\leq t}))^2. \tag{9}$$

We then use $l_t(\phi)$ as our estimate for $l(\mathbf{x}_{\leq t}, \mathbf{s}_{\leq t})$.

f_ϕ can be implemented using any function approximator capable of dealing with time-series inputs (though in the static setting it needs only to be able to deal with static inputs). In this paper, we model f_ϕ as a Recurrent Neural Network (RNN) (in particular as an LSTM Hochreiter and Schmidhuber (1997)).

We explicitly model the predictor function f_ϕ using the RNN structure as follows. At time stamp t, we first define the hidden state H_t by

$$H_t = f_1(H_{t-1}, \mathbf{s}_t, \mathbf{x}(\mathbf{s}_t))$$

where f_1 is some function parameterized as a fully connected network (the same network is used for each time point). The output of the predictor network is then given by

$$f_\phi(\mathbf{x}(\mathbf{s}_{\leq t}), \mathbf{s}_{\leq t}) = f_2(H_t) = f_2(f_1(H_{t-1}, \mathbf{s}_t, \mathbf{x}(\mathbf{s}_t)))$$

for another function f_2 parameterized as a (different) fully connected network.

Note that H_t depends on $H_{t-1}, \mathbf{s}_t$ and $\mathbf{x}(\mathbf{s}_t)$. Iterating this dependency we get that H_t depends on $\mathbf{s}_{\leq t}$ and $\mathbf{x}(\mathbf{s}_{\leq t})$.

3.2. Selector function

The selector function, $f_\theta : \prod_{i=1}^{t}(\mathcal{X} \times \{0,1\}^d) \to [0,1]^d$, outputs probabilities from which we sample independently to obtain a sensing decision. The probability of a given sensing decision, $\mathbf{s} = (s^1, ..., s^d)$ given the observations and selections made until time t is given by

$$\pi_\theta(\mathbf{s}|\mathbf{x}(\mathbf{s}_{\leq t}), \mathbf{s}_{\leq t}) = \prod_{i=1}^{d} f_\theta(\mathbf{x}(\mathbf{s}_{\leq t}), \mathbf{s}_{\leq t})^{s^i}(1 - f_\theta(\mathbf{x}(\mathbf{s}_{\leq t}), \mathbf{s}_{\leq t}))^{1-s^i}$$

Using a slight abuse of notation, we will write $\mathbf{s} \sim \theta$ and $\mathbf{s}_t \sim \theta|\mathbf{s}_{\leq t-1}$ to denote the marginal and conditional distribution of the sensing decision induced by the selector network (note that both of these are conditional on $\mathbf{x}_{\leq t-1}$). Using this, the objective function in equation (6) can be rewritten as follows (we omit the outer expectation ($\mathbb{E}_{\mathbf{x}\sim p_X}$) due to space limitation and replace $l(\mathbf{x}_{\leq t}, \mathbf{s}_{\leq t})$ with $l_t(\phi)$):

$$\begin{aligned}
\mathcal{L}(\theta) &= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{s}\sim\theta}\big[l_t(\phi) + \lambda \mathbf{c}^T \mathbf{s}_t\big] \qquad (10)\\
&= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{s}_1\sim\theta}\Big[\cdots \mathbb{E}_{\mathbf{s}_t\sim\theta|\mathbf{s}_{\leq t-1}}\big[l_t(\phi) + \lambda \mathbf{c}^T \mathbf{s}_t\big]\Big]\\
&= \sum_{t=1}^{T} \sum_{\mathbf{s}_1\in\{0,1\}^d} \pi_\theta(\mathbf{s}_1)\Big[\sum_{\mathbf{s}_t\in\{0,1\}^d} \pi_\theta(\mathbf{s}_t|\mathbf{s}_{\leq t-1}) \times \big[l_t(\phi) + \lambda \mathbf{c}^T \mathbf{s}_t\big]\Big]\\
&= \sum_{t=1}^{T} \sum_{\mathbf{s}_{\leq t}\in\{0,1\}^{d\times t}} [\prod_{\tau=1}^{t} \pi_\theta(\mathbf{s}_\tau|\mathbf{s}_{\leq\tau-1})]\big[l_t(\phi) + \lambda \mathbf{c}^T \mathbf{s}_t\big]
\end{aligned}$$

Using ideas from actor-critic models Konda and Tsitsiklis (2000) (details can be found in the Supplementary Materials), the gradient of this loss $\nabla_\theta \mathcal{L}(\theta)$ can be shown to be

$$\nabla_\theta \mathcal{L}(\theta) = \sum_{t=1}^{T}\sum_{j=1}^{t} \mathbb{E}_{\mathbf{s}\sim\theta}\big[[l_t(\phi) + \lambda \mathbf{c}^T \mathbf{s}_t]\nabla_\theta \log \pi_\theta(\mathbf{s}_j|\mathbf{s}_{\leq j-1})\big] \qquad (11)$$

where $\nabla_\theta \log \pi_\theta(\mathbf{s}_j|\mathbf{s}_{\leq j-1})$ is

$$\sum_{i=1}^{d}\Big[s_j^i \nabla_\theta \log f_\theta^i(\mathbf{x}(\mathbf{s}_{\leq j-1}), \mathbf{s}_{\leq j-1}) - (1 - s_j^i)\nabla_\theta \log f_\theta^i(\mathbf{x}(\mathbf{s}_{\leq j-1}), \mathbf{s}_{\leq j-1})\Big]. \qquad (12)$$

which can be directly deduced from Equation (10).

We explicitly model the selector function f_θ using the RNN structure as follows. At time stamp t, we first define the hidden state h_t by

$$h_t = f_3(h_{t-1}, \mathbf{s}_t, \mathbf{x}(\mathbf{s}_t))$$

where f_3 is some function parameterized as a fully connected network (the same network is used for each time point). The output of the selector network is then given by

$$f_\theta(\mathbf{x}(\mathbf{s}_{\leq t}), \mathbf{s}_{\leq t}) = f_4(h_t) = f_4(f_3(h_{t-1}, \mathbf{s}_t, \mathbf{x}(\mathbf{s}_t)))$$

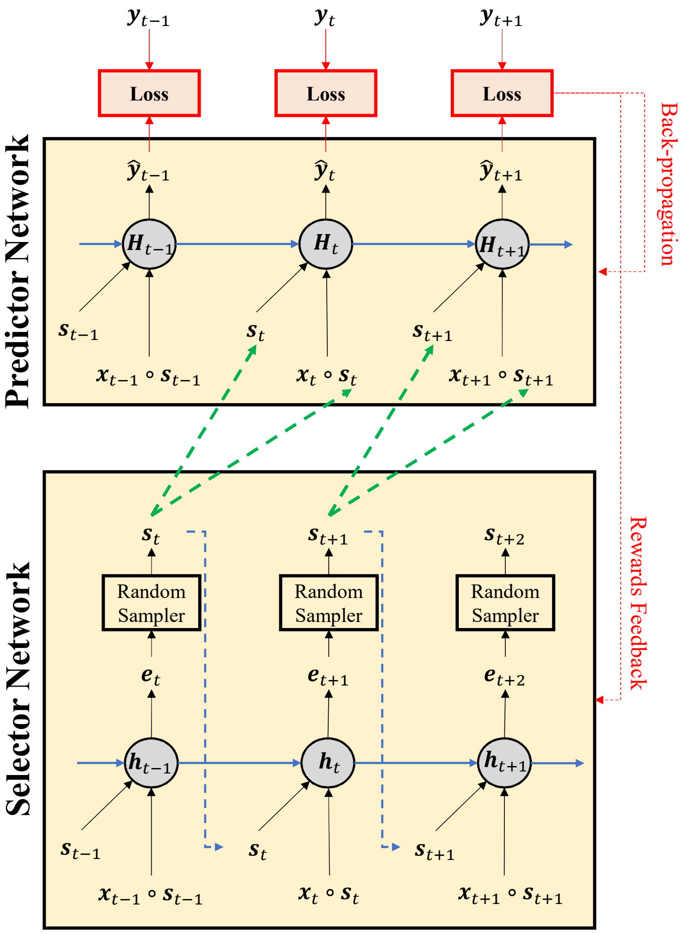

Figure 3: Block diagram of ASAC in a time-series setting.

for another function f_4 parameterized as a (different) fully connected network.

Note that h_t depends on $h_{t-1}, \mathbf{s}_t$ and $\mathbf{x}(\mathbf{s}_t)$. Iterating this dependency we get that h_t depends on $\mathbf{s}_{\leq t}$ and $\mathbf{x}(\mathbf{s}_{\leq t})$.

Fig. 2 illustrates the entire structure of ASAC. Fig. 3 illustrates ASAC in the time-series setting. The lower part of Fig. 3 depicts the selector network (e_t represents the output of f_θ at time stamp t) and the upper part of the Fig. 3 depicts the predictor network.

3.3. Training the networks

The selector and predictor networks are jointly and iteratively trained. First, the predictor network (f_ϕ) is trained to minimize the predictor loss $\mathcal{L}(\phi)$ given the sensing decisions made by the selector network (f_θ). We investigated the effect of sampling multiple sensing decisions for the same time-step and sample but found that this had very little effect on the performance. As such, when we create a mini-batch to train the predictor network with, we sample only 1 sensing decision for each sample in the mini-batch.

The parameters of the predictor network are updated according to

$$\phi \leftarrow \phi - \beta \frac{1}{n_{mb}} \sum_{i=1}^{n_{mb}} \sum_{t=1}^{T_i} (y_{t,i} - f_\phi(\mathbf{x}(\mathbf{s}_{\leq t,i}), \mathbf{s}_{\leq t,i}))^2$$

where n_{mb} is the size of the mini-batch and $\beta > 0$ is the learning rate (specific to the predictor network). Then, given a fixed predictor network, the selector network parameters are updated according to

$$\theta \leftarrow \theta - \alpha \frac{1}{n_{mb}} \sum_{i=1}^{n_{mb}} \sum_{t=1}^{T_i} \sum_{\tau=1}^{t} \Big[\big[l_{t,i}(\phi) + \lambda \mathbf{c}^T \mathbf{s}_{t,i} \big] \times \nabla_\theta \log f_\theta(\mathbf{x}(\mathbf{s}_{\leq \tau,i}), \mathbf{s}_{\leq \tau,i}) \Big]$$

where $\alpha > 0$ the learning rate (specific to the selector network). Pseudo-code can be found in the Supplementary Materials.

3.3.1. Missing Data during Training

The loss we have derived lends itself naturally to missing data in the training set. By inspecting Equations (11) and (12), we see that the gradient is made up of a sum over each feature. During training, when "back-propagating" to the selector network, for features that were selected by the network but were missing (and so their measurement can't be given), we do not back-propagate their loss. The selector network only back-propagates for both not-selected features and selected-and-not-missing features.

4. Experiments

4.1. Data Description

We use two real-world medical datasets to evaluate the performance of ASAC against Deep Sensing Yoon et al. (2018b) for various cost constraints.

ADNI dataset: The Alzheimers Disease Neuro-imaging Initiative (ADNI) study data is a longitudinal survival dataset of per-visit measurements for 1,737 patients Petersen et al. (2010). The data tracks disease progression through clinical measurements at 1/2-year intervals, including quantitative biomarkers, cognitive tests, demographics, and risk factors. For this dataset, the adverse event we predict is unstable state occurrence.

MIMIC-III dataset: The MIMIC-III dataset Johnson et al. (2016) has de-identified electronic health records (EHR) from Beth Israel Deaconess Medical Center from 2001 to 2012. It was collected from two information systems (Philips CareVue Clinical and iMDsoft MetaVision ICU) that have very different data structures. We only use data collected by MetaVision (after 2008) for consistency. We extract 40 physiological data streams from lab tests (20) and vital signs (20) that have the lowest missing rates (including heart rate, respiratory rate, blood pressures). The number of patients is 23,153 and there are 5,143 sequences of length larger than 100 time steps with the longest being 1,487 time steps. For this dataset, the adverse event we predict is death.

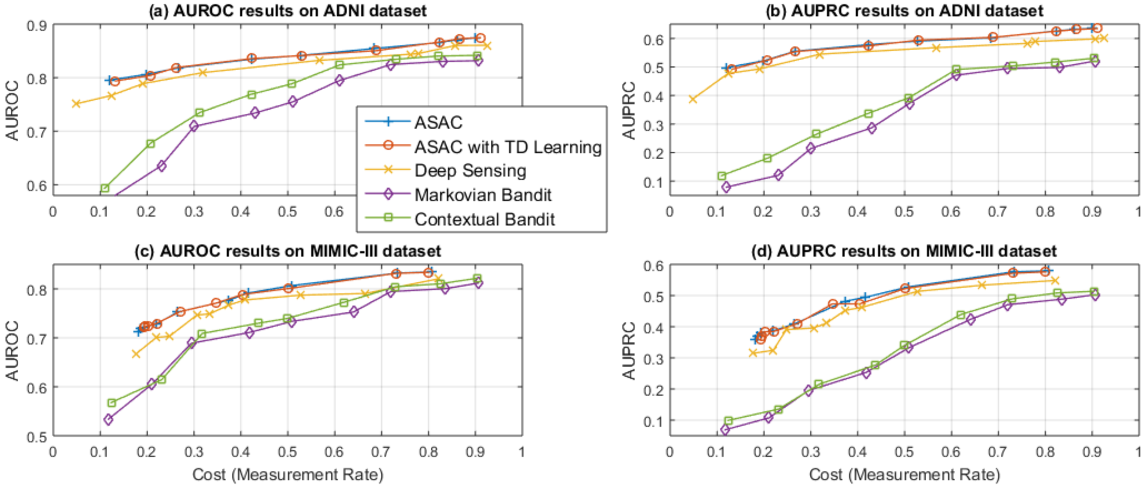

Figure 4: Results on risk predictions on both ADNI and MIMIC-III dataset with various cost constraints in terms of AUROC and AUPRC. X-axis is cost constraints (rate of selected measurements). Y-axis is predictive performance.

4.2. Experimental Results

We evaluate the performance of ASAC against 3 benchmarks: (1) Deep Sensing Yoon et al. (2018b), (2) Contextual Bandit Li et al. (2010); Agarwal et al. (2014), (3) Markovian Bandit Gai et al. (2011). Furthermore, we also evaluate our model when replacing the actor-critic methodology with TD learning Sutton (1988) and refer to this model as ASAC with TD learning. We randomly divided the dataset into mutually exclusive training (80%) and testing (20%) sets. We conducted 10 independent experiments with different training/testing sets in each and we report the mean and standard deviation of the performance in the 10 experiments.

In Fig. 4, we plot AUROC and AUPRC against the average measurement rate of all features (corresponding to all features being assigned the same cost). In MIMIC-III, we ignore the cost when a missing feature is selected.

As can be seen in Fig. 4, ASAC (and ASAC with TD learning) consistently outperforms all 3 benchmarks, achieving higher predictive power for the same cost across all costs. Variance analysis can be found in the Supplementary Materials showing that ASAC achieves statistically significant improvements over all 3 benchmarks. We see from Fig. 4(c)(d) that ASAC is robust to missing data, where we note that around 40% of the data is missing in the MIMIC-III dataset. ASAC and ASAC with TD learning achieve similar performances indicating that the ASAC framework can be robustly combined with various Reinforcement Learning frameworks to address the active sensing problem.

We can see a trade-off between accuracy and observational costs. In the ASAC framework, we can either maximize the accuracy given constraints on observational costs or minimize the cost given the desired accuracy constraint. As can be seen in the grid line in Figure 4; the horizontal line represents fixing the accuracy, vertical line represents fixing the cost. We illustrate these trade-off curves for ASAC in Figure 4, which shows that ASAC outperforms state-of-the-art under both types of constraints.

4.3. Analysis on ASAC with Synthetic datasets

We perform 3 synthetic experiments that we believe capture key attributes of an active sensing method. In each simulation, the feature distribution is a 10 dimensional auto-regressive Gaussian model over 10 time steps, i.e.

$$\mathbf{X}_t = \phi \odot \mathbf{X}_{t-1} + (1 - \phi) \odot \mathbf{Z}_t \tag{13}$$

where $\odot$ denotes element-wise multiplication, $\phi \in [0, 1]^{10}$ is a vector that determines the dependency of each feature on the past (a higher ϕ corresponds to a larger dependency on the past) and $\mathbf{Z}_t$ is an independent Gaussian noise vector $\mathbf{Z}_t \sim \mathcal{N}(0, \mathbf{I}^{10})$.

4.3.1. Time dependency vs Measurement rate

In our first experiment, we investigate the effect of time dependency on measurement rate of a variable. If we fix the cost and label-dependency of all variables to be the same, then we would expect a variable with a large ϕ to be measured less frequently by a good active sensing method (due to being more easily predicted from previous values).

To do this, we set the label, Y_t according to

$$Y_t = \exp(-0.1 \times |\sum_{i=1}^{10} X_t^i|) + \epsilon \tag{14}$$

where $\epsilon \sim \mathcal{N}(0, 0.1)$. We set the cost for each variable to be the same, which we vary from 1 to 5. We set $\phi = (0, 0.1, ..., 0.9)$. The measurement rate (the selection probability) of each variable is reported in Table 1, along with the overall RMSE for each experiment.

ϕ^i/Cost	1	2	3	4	5
0 ($\mathbf{X}_t^1$)	1.00	1.00	1.00	0.46	0.38
0.1 ($\mathbf{X}_t^2$)	1.00	1.00	1.00	0.44	0.36
0.2 ($\mathbf{X}_t^3$)	1.00	1.00	1.00	0.30	0.26
0.3 ($\mathbf{X}_t^4$)	1.00	1.00	1.00	0.25	0.12
0.4 ($\mathbf{X}_t^5$)	1.00	1.00	1.00	0.22	0.10
0.5 ($\mathbf{X}_t^6$)	1.00	0.98	0.23	0.21	0.07
0.6 ($\mathbf{X}_t^7$)	1.00	0.94	0.13	0.10	0.05
0.7 ($\mathbf{X}_t^8$)	1.00	0.93	0.07	0.03	0.01
0.8 ($\mathbf{X}_t^9$)	0.92	0.41	0.03	0.02	0.0
0.9 ($\mathbf{X}_t^{10}$)	0.45	0.11	0.01	0.01	0.0
RMSE	0.106	0.110	0.126	0.138	0.146

Table 1: Measurement rate of each feature when each feature has a different auto-regressive coefficient.

As can be seen in Table 1, ASAC meets our expectations. Features with a low ϕ, are regularly re-measured since past values are not as predictive of the present value, whereas features with a high ϕ are measured less frequently. As cost increases, we also see a monotonic decrease in the measurement rate of all variables.

4.3.2. Cheaper but noisier features

In our second synthetic experiment, we investigate the effect of having cheaper, noisier versions of our original 10 features. In this experiment we are interested in understanding how well ASAC can trade-off between the cost and noise level of the noisy versions. This setting has several real-world parallels; in medicine, cheap at-home tests (such as blood pressure tests and home pregnancy tests) exist, but are less reliable (noisier) than the more expensive state-of-the-art procedures that would be used in, say, a hospital setting.

To model this, we introduce 10 new noisy features

$$\hat{\mathbf{X}}_t = \mathbf{X}_t + \delta \tag{15}$$

where $\delta \sim \mathcal{N}(0, \gamma)$ with $\gamma > 0$ controlling the "noisiness". In this experiment, we set the label according to

$$Y_t = \exp(-|0.1X_t^1 + 0.2X_t^2 + 0.3X_t^3 + 0.4X_t^4|) + \epsilon \tag{16}$$

where now we have set different magnitudes for the coefficients of the first 4 variables (and the last 6 variables are now just there as pure noise). We would expect that as we increase the cost of the true variables (or equivalently decrease the cost of the noisy variables), the variables with lower importance (X^1 and X^2) will be the first ones to be "replaced" with their noisy version, whereas it will take a higher cost for X^4 to be replaced with $\hat{X}^4$.

We fix the cost of the original features to be 1, and investigate noise levels $\gamma \in \{0.2, 0.4, 0.6\}$ and vary the cost of a noisy feature to be $\hat{c} \in \{0.1.0.2.0.5\}$. We set $\phi^i = 0.5$ for all i. In Table 2 we report the measurement rate of each of the first 4 variables and their noisy versions.

γ	Cost	1	0.1	1	0.2	1	0.5
	Features	$\mathbf{X}_t$	$\hat{\mathbf{X}}_t$	$\mathbf{X}_t$	$\hat{\mathbf{X}}_t$	$\mathbf{X}_t$	$\hat{\mathbf{X}}_t$
0.2	X^1	0.00	1.00	0.00	1.00	0.00	1.00
	X^2	0.00	1.00	0.00	1.00	0.29	0.70
	X^3	0.00	1.00	0.00	1.00	1.00	0.00
	X^4	0.00	1.00	0.00	1.00	1.00	0.00
0.4	X^1	0.00	1.00	0.00	0.94	1.00	0.00
	X^2	0.00	1.00	0.30	0.65	1.00	0.00
	X^3	1.00	0.00	1.00	0.00	1.00	0.00
	X^4	1.00	0.00	1.00	0.00	1.00	0.00
0.6	X^1	0.00	1.00	0.51	0.33	1.00	0.00
	X^2	0.75	0.25	1.00	0.00	1.00	0.00
	X^3	1.00	0.00	1.00	0.00	1.00	0.00
	X^4	1.00	0.00	1.00	0.00	1.00	0.00

Table 2: Measurement rate based on different cost and noise parameter γ for original feature ($\mathbf{X}_t$) and noisy feature ($\hat{\mathbf{X}}_t$).

As can be seen in Table 2, ASAC meets our expectations. As we move right and down in the table (corresponding to an increasing cost for the noisy feature and increasing noise, respectively), we see that true features are selected more frequently but that the noisy versions for the less predictive features (X^1 and X^2) are sometimes selected even at higher costs and noise levels. In particular, at $(\gamma, \hat{c}) = (0.2, 0.2)$, only the noisy features are selected. When γ is increased to 0.4, ASAC starts to select X^3 and X^4 all of the time, and X^2 some of the time, while the noisy version of X^1 is always preferred. When we increase γ to 0.6, the true version of X^2 is the only version selected by ASAC and the true version of X^1 finally becomes desirable enough to measure (sometimes).

4.3.3. Y dependent cost

η	0.1		0.3		0.5	
Features	$\mathbf{X}_t$	$\hat{\mathbf{X}}_t$	$\mathbf{X}_t$	$\hat{\mathbf{X}}_t$	$\mathbf{X}_t$	$\hat{\mathbf{X}}_t$
$Y_t = 1$	0.89	0.10	0.63	0.21	0.25	0.69
$Y_t = 0$	0.13	0.81	0.14	0.80	0.12	0.78

Table 3: Measurement rate when the cost is different for $Y_t = 1$ and $Y_t = 0$.

In our final synthetic experiment, we allow for a cost that depends on Y. In our medical example, this could correspond to the fact that when a patient is sick, it is more important to be sure about it, than when a patient is well. In the presence of the cheaper-but-noisier features from 4.3.2, we expect a worsening condition to create a switch in selections. While a patient is healthy, we are happy to monitor the patient using the at-home tests, but when a patients condition appears to be deteriorating, it becomes more important that accurate measurements are made than cost being kept low.

We model this by incorporating the patients condition into the cost, setting the cost when the patient is sick ($Y_t = 1$) to be $\eta \in [0, 1]$ times the cost when the patient is healthy ($Y_t = 0$)[5]. We investigate $\eta \in \{0.1, 0.3, 0.5\}$.

We generate the true features as before, now with $\phi^i = 0.9$ for all i, and generate noisy features as in 4.3.2 with $\gamma = 0.4$. We set the label to be binary according to

$$Y_t = \begin{cases} 1, \text{ w.p } \exp(-0.1 \times |\sum_{i=1}^{10} X_t^i + \epsilon - 2|) \\ 0, \text{ w.p } 1 - \exp(-0.1 \times |\sum_{i=1}^{10} X_t^i + \epsilon - 2|) \end{cases}$$

where “w.p” means “with probability”.

We see from Table 3 that ASAC is able to correctly identify that measuring the true features is more important when $Y_t = 1$, with measurement frequencies while $Y_t = 1$ for the true features being higher for all 3 values of η. When $\eta = 0.5$, which corresponds to the cost being half as important while the patient is sick, we see that true features are measured twice as frequently. As η decreases, and so accurate predictions become more important, we see that ASAC selects true features more frequently. When $\eta = 0.1$, ASAC

[5] By reducing the measurement cost, we are equivalently up-weighting the importance of accurately predicting.

selects true feature nearly 7 times more frequently while the patient is sick compared to when they are not. ASAC can therefore be used to handle settings where the trade-off between measurement cost and prediction accuracy varies according to the label (which is often the case in medicine).

5. Conclusion

We propose a novel active sensing framework, called Active Sensing using Actor-Critic models (ASAC), to address the important question of *what* and *when* to observe. This is critical when observations are costly. We demonstrated through real-world and synthetic experiments that the ASAC framework can significantly reduce the cost of observation with only a small loss in predictive power. Using the MIMIC-III dataset we also demonstrated that ASAC is robust to missing data.

We believe ASAC has wide-ranging applications, both in cost reduction but also for things such as planning, in which patients can be told when they might expect to need their next check-up and for what (i.e. personalized screening).

Acknowledgments

The authors would like to thank the reviewers for their helpful comments. The research presented in this paper was supported by the Office of Naval Research (ONR) and the NSF (Grant number: ECCS1462245, ECCS1533983, and ECCS1407712).

References

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

Ahmed M Alaa and Mihaela van der Schaar. Balancing suspense and surprise: Timely decision making with endogenous information acquisition. In *Advances in Neural Information Processing Systems*, pages 2910–2918, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Paul R Billings, Mel A Kohn, Margaret De Cuevas, Jonathan Beckwith, Joseph S Alper, and Marvin R Natowicz. Discrimination as a consequence of genetic testing. *American journal of human genetics*, 50(3):476, 1992.

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. Learning to explain: An information-theoretic perspective on model interpretation. *arXiv preprint arXiv:1802.07814*, 2018.

Michael E Ezzie, Scott K Aberegg, and James M O'Brien Jr. Laboratory testing in the intensive care unit. *Critical care clinics*, 23(3):435–465, 2007.

US Preventive Services Task Force et al. Screening for breast cancer: Us preventive services task force recommendation statement. *Annals of internal medicine*, 151(10):716, 2009.

Yi Gai, Bhaskar Krishnamachari, and Mingyan Liu. On the combinatorial multi-armed bandit problem with markovian rewards. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pages 1–6. IEEE, 2011.

Peter C Gøtzsche and Karsten Juhl Jørgensen. Screening for breast cancer with mammography. *Cochrane database of systematic reviews*, (6), 2013.

SH Heywang-Köbrunner, P Viehweg, A Heinig, and CH Küchler. Contrast-enhanced mri of the breast: accuracy, value, controversies, solutions. *European journal of radiology*, 24(2): 94–108, 1997.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Hèlen Koch, Marloes A van Bokhoven, Gerben ter Riet, JM Tineke van Alphen-Jager, Trudy van der Weijden, Geert-Jan Dinant, and Patrick JE Bindels. Ordering blood tests for patients with unexplained fatigue in general practice: what does it yield? results of the vampire trial. *Br J Gen Pract*, 59(561):e93–e100, 2009.

Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

Kanya Kumwilaisak, Alberto Noto, Ulrich H Schmidt, Clare I Beck, Claudia Crimi, Kent Lewandrowski, and Luca M Bigatello. Effect of laboratory testing guidelines on the utilization of tests and order entries in a surgical intensive care unit. *Critical care medicine*, 36(11):2993–2999, 2008.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

Ronald Carl Petersen, PS Aisen, LA Beckett, MC Donohue, AC Gamst, DJ Harvey, CR Jack, WJ Jagust, LM Shaw, AW Toga, et al. Alzheimer's disease neuroimaging initiative (adni): clinical characterization. *Neurology*, 74(3):201–209, 2010.

Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

DANIEL E Roberts, DEAN D Bell, TRISH Ostryzniuk, KAREN Dobson, LUIS Oppenheimer, DOUGLAS Martens, NICHOLAS Honcharik, HARVEY Cramp, ELFREIDE Loewen, and SHEILA Bodnar. Eliminating needless testing in intensive care–an information-based team management approach. *Critical care medicine*, 21(10):1452–1458, 1993.

Debbie Saslow, Carla Boetes, Wylie Burke, Steven Harms, Martin O Leach, Constance D Lehman, Elizabeth Morris, Etta Pisano, Mitchell Schnall, Stephen Sener, et al. American cancer society guidelines for breast screening with mri as an adjunct to mammography. *CA: a cancer journal for clinicians*, 57(2):75–89, 2007.

Charles E Schroeder, Donald A Wilson, Thomas Radman, Helen Scharfman, and Peter Lakatos. Dynamics of active sensing and perceptual selection. *Current opinion in neurobiology*, 20(2):172–176, 2010.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, page 125, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

Milton C Weinstein, Louise B Russell, Marthe R Gold, Joanna E Siegel, et al. *Cost-effectiveness in health and medicine*. Oxford university press, 1996.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920*, 2018a.

Jinsung Yoon, William R. Zame, and Mihaela van der Schaar. Deep sensing: Active sensing using multi-directional recurrent neural networks. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=r1SnX5xCb.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. INVASE: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJg_roAcK7.

Shipeng Yu, Balaji Krishnapuram, Romer Rosales, and R. Bharat Rao. Active sensing. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, pages 639–646, 2009.

Derivation of the gradient of $\mathcal{L}(\theta)$

$$\begin{aligned}
\nabla_\theta \mathcal{L}(\theta) =& \sum_{t=1}^{T} \sum_{\mathbf{s}_{\leq t}\in\{0,1\}^{d\times t}} \nabla_\theta \Big[\prod_{\tau=1}^{t} \pi_\theta(\mathbf{s}_\tau|\mathbf{s}_{\leq\tau-1})\Big]\big[l_t(\phi)+\lambda\mathbf{c}^T\mathbf{s}_t\big]\\
=& \sum_{t=1}^{T} \sum_{\mathbf{s}_{\leq t}\in\{0,1\}^{d\times t}} \sum_{j=1}^{t} \frac{\nabla_\theta \pi_\theta(\mathbf{s}_j|\mathbf{s}_{\leq j-1})}{\pi_\theta(\mathbf{s}_j|\mathbf{s}_{\leq j-1})} \times \Big[\prod_{\tau=1}^{t} \pi_\theta(\mathbf{s}_\tau|\mathbf{s}_{\leq\tau-1})\Big]\big[l_t(\phi)+\lambda\mathbf{c}^T\mathbf{s}_t\big]\\
=& \sum_{t=1}^{T} \sum_{\mathbf{s}_{\leq t}\in\{0,1\}^{d\times t}} \sum_{j=1}^{t} \nabla_\theta \log \pi_\theta(\mathbf{s}_j|\mathbf{s}_{\leq j-1}) \times \Big[\prod_{\tau=1}^{t} \pi_\theta(\mathbf{s}_\tau|\mathbf{s}_{\leq\tau-1})\Big]\big[l_t(\phi)+\lambda\mathbf{c}^T\mathbf{s}_t\big]\\
=& \sum_{t=1}^{T} \sum_{j=1}^{t} \mathbb{E}_{\mathbf{s}\sim\theta}\big[\nabla_\theta \log \pi_\theta(\mathbf{s}_j|\mathbf{s}_{\leq j-1}) \times [l_t(\phi)+\lambda\mathbf{c}^T\mathbf{s}_t]\big]
\end{aligned}$$

Further Related Works

Parallels can be drawn between active sensing and attention mechanisms Bahdanau et al. (2014); Vaswani et al. (2017), though like instance-wise variable selection, attention observes the entire set of measurements and then decides which time points to “focus on”. In contrast to instance-wise variable selection, attention is typically applied over time. Attention was first introduced and has been more thoroughly explored as “soft” attention Bahdanau et al. (2014); Vaswani et al. (2017) in which different time points are weighted (and *not* hard-selected) according to their importance. Active sensing, on the other hand, is only meaningful in a hard-selection setting (since weighting measurements still requires them to be measured and therefore the cost is still incurred). Hard attention mechanisms do exist Xu et al. (2015), but they share the characteristic found in both instance-wise variable selection and soft attention in that all values must be observed before a selection is made.

In Yu et al. (2009), they propose a solution for active sensing using a Bayesian approach with Gaussian processes. Data stream are modelled as Gaussian processes and therefore, the complexity of the algorithm increases quadratically in the dimensionality of the data and estimation accuracy decreases quickly with the number of dimensions. Alaa and van der Schaar (2016) discuss the active sensing problem for a single data stream observed over time, reducing the problem from *what* and *when* to observe to just *when* to observe. In Alaa and van der Schaar (2016), they explicitly model the stream as a given stochastic process and use the characteristics of the assumed process to learn optimal sampling times. This work cannot be applied in the multi-stream setting we investigate in this paper.

The information bottleneck Tishby et al. (2000) attempts to find a representation $\tilde{X}$ that is a function of X trading off between maximizing the mutual information between $\tilde{X}$ and Y and minimizing the mutual information between $\tilde{X}$ and X. The first contrast with the active sensing framework is that when constructing $\tilde{X}$, the entire features are used, whereas in active sensing, the decision of what to select is only based on previously selected features. Moreover, $\tilde{X}$ does not necessarily correspond to a subset of the features, but can lie in an entirely different representation space. In ASAC, the selected features are necessarily a subset of the features, and cannot just be any arbitrary mapping of them. We also aim to minimize the cost of the selection, which is not the same as minimizing the mutual information between $\tilde{X}$ and X.

	Active sensing	Time-series	Multi-variate	(Non-) Causal	Optimization
ASAC	✓	✓	✓	Causal	Joint
Yoon et al. (2018b)	✓	✓	✓	Causal	Individual
Qin et al. (2017)		✓	✓	Non-causal	Joint
Bahdanau et al. (2014)		✓	✓	Non-causal	Joint
Vaswani et al. (2017)		✓	✓	Non-causal	Joint
Yoon et al. (2019)			✓	Non-causal	Joint
Alaa and van der Schaar (2016)	✓	✓		Causal	Joint

Table 4: Comparison of related works. Causal refers to whether or not a selection depends on future selections or not.

Additional Real-world Experiments: Variance Analysis

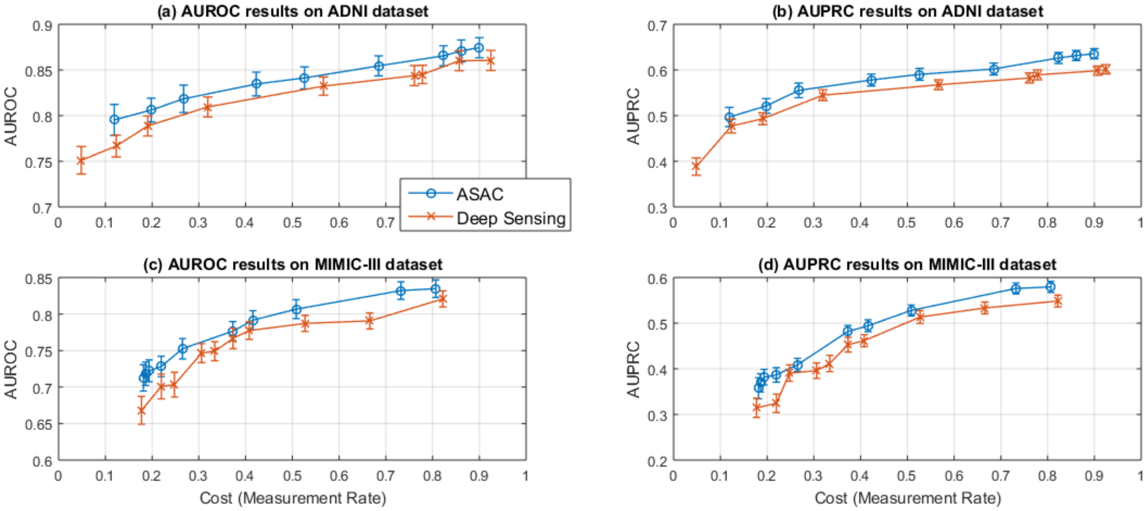

Figure 5: Risk predictions on ADNI and MIMIC-III datasets. X-axis is cost constraints (rate of selected measurements). Y-axis is predictive performance. Error bar represents the confidence bounds of the performance.

Additional Figures

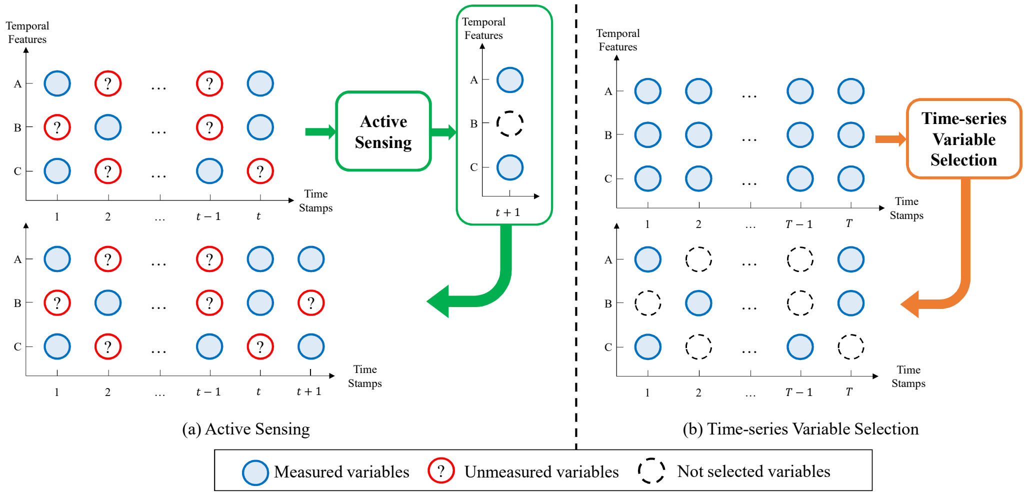

Figure 6: Comparison of Active Sensing and Instance-wise Variable selection in the time-series setting.

Pseudo-code

Algorithm 1 Pseudo-code of ASAC

```
**Inputs:** learning rates $\alpha, \beta > 0$, mini-batch size $n_{mb} > 0$, dataset $\mathcal{D}$, hyper-parameter $\lambda \geq 0$
**Initialize** parameters $\theta, \phi$
**while** training loss has not converged **do**
  Sample a mini-batch $(\mathbf{x}_{\leq T_i,i}, \mathbf{y}_{\leq T_i,i})_{i=1}^{n_{mb}} \sim \mathcal{D}$
  **for** $i = 1, ..., n_{mb}$ **do**
    **for** $t = 1, ..., T_i$ **do**
      Calculate selection probability vector
          $\mathbf{e}_{t,i} \leftarrow f_\theta(\mathbf{x}(\mathbf{s}_{\leq t,i}), \mathbf{s}_{\leq t,i})$
      Sample selection vector from $\mathbf{e}_{t,i}$
          $\mathbf{s}_{t,i} \sim Ber(\mathbf{e}_{t,i})$
      Calculate loss $l_{t,i}(\phi)$
          $l_{t,i}(\phi) \leftarrow (y_{t,i} - f_\phi(\mathbf{x}(\mathbf{s}_{\leq t,i}), \mathbf{s}_{\leq t,i}))^2$
    **end for**
  **end for**
  Update the predictor network parameters $\phi$
      $\phi \leftarrow \phi - \beta \frac{1}{n_{mb}} \sum_{i=1}^{n_{mb}} \sum_{t=1}^{T_i} (y_{t,i} - f_\phi(\mathbf{x}(\mathbf{s}_{\leq t,i}), \mathbf{s}_{\leq t,i}))^2$
  Update the selector network parameters $\theta$
      $\theta \leftarrow \theta - \alpha \frac{1}{n_{mb}} \sum_{i=1}^{n_{mb}} \sum_{t=1}^{T_i} \sum_{\tau=1}^{t} \Big[ \big[ l_{t,i}(\phi) + \lambda \mathbf{c}^T \mathbf{s}_{t,i} \big]$
          $\times \nabla_\theta \log f_\theta(\mathbf{x}(\mathbf{s}_{\leq \tau,i}), \mathbf{s}_{\leq \tau,i}) \Big]$
**end while**
```