Optimizing Early Warning Classifiers to Control False Alarms via a Minimum Precision Constraint

Preetish Rath **Michael C. Hughes**
Dept. of Computer Science, Tufts University

Abstract

Early warning prediction systems can suffer from high false alarm rates that limit utility, especially in settings with high class imbalance such as healthcare. Despite the widespread need to control false alarms, the dominant classifier training paradigm remains minimizing cross entropy, a loss function which does not treat false alarms differently than other types of mistakes. While existing efforts often try to reduce false alarms by post-hoc threshold selection after training, we suggest a comprehensive solution by changing the loss function used to train the classifier. Our proposed objective maximizes recall while enforcing a constraint requiring precision to exceed a specified value. We make our objective tractable for gradient-based optimization by developing tight sigmoidal bounds on the counts needed to compute precision and recall. Our objective is applicable to any classifier trainable via gradient descent, including linear models and neural networks. When predicting mortality risk across two large hospital datasets, we show how our method satisfies a desired constraint on false alarms while achieving better recall than alternatives.

1 INTRODUCTION

Machine learning has led to state-of-the-art early warning alert systems for many high-stakes applications, from public health to finance to earthquake safety (Escobar et al., 2020; Mousavi et al., 2019; Al Banna et al., 2020). In this work, we are motivated by applications in healthcare, especially recent early warning alert systems for critical care hospital settings (Hyland et al., 2020; Sendak et al., 2020; Wellner et al., 2017). In such settings, alerts are intended to trigger extra attention from clinical staff on specific patients predicted to be at risk of adverse outcomes. Clinicians have limited time that could have many productive uses; it is critical that automated alerts identify patients who truly need help and do not suggest those who do not need attention.

An unneeded alert, also known as a *false alarm* or a *false positive*, has two detrimental consequences. In the moment, it pulls resources away from where they could be better used. In the long term, too many false alarms can cause clinical staff to distrust the alerts all together, a phenomenon known as *alarm fatigue* (Cvach, 2012; Deb & Claudio, 2015; Sendelbach & Funk, 2013). Some false alarms are inevitable, especially in clinical tasks where the adverse outcome to predict is quite rare. In the mortality risk task we study later, only about 8-10% of patients die in the hospital, and most survive many hours before death. If a system is able to issue alerts throughout a patient's stay, it is critical that the model is designed to avoid alarm fatigue and deliver overall net benefit. If the tool fails to limit false alarms to an acceptable rate, alerts may be ignored completely.

The key challenge of designing alert systems is thus to *balance* the false alarm rate (related to *precision*) with the true positive rate (known as *recall*) (Romero-Brufau et al., 2015). In the clinic, recall measures the fraction of truly at-risk patients correctly identified by an alert. Unfortunately, standard objectives for training early-warning classifiers, such as binary cross entropy (BCE), are not designed specifically to address false alarms. Many early warning systems for clinical settings (Hyland et al., 2020; Futoma et al., 2017) are trained using such objectives and only balance false alarm concerns in a secondary threshold selection or early stopping stage *after* training. Such post-hoc adjustment is limited and may not identify the ideal tradeoff between recall and precision. Our experiments[1] show the deficiency of binary cross entropy even with post-hoc adjustment or per-class weighting.

To overcome this challenge, in this paper we show that

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

[1] Code URL: github.com/tufts-ml/false-alarm-control

any binary classifier trainable via stochastic gradient descent (SGD) can, by changing its loss function, be steered toward solutions that offer better control of false alarms. Our proposed objective seeks to maximize the number of truly at-risk patients who are helped by alerts (maximize recall), subject to meeting a guarantee on the fraction of all alarms that will be false.

This paper makes two key contributions:

1. New tractable bounds for maximizing recall subject to a minimum precision constraint. While it is easy to compute a model's precision on a concrete dataset, it is not tractable to compute gradients with respect to precision, making optimization difficult. Previous work by Eban et al. (2017) suggested surrogate bounds based on the hinge loss that are amenable to SGD. However, we find these bounds are too loose and lead to suboptimal performance. We thus derive a family of bounds based on the logistic sigmoid function that can be made arbitrarily tight. We find our new bounds achieve far better performance.

2. Demonstration of empirical success on a realistic clinical task. While our constrained optimization objective has been suggested before for general-purpose settings (Eban et al., 2017), we find that for clinical early warning systems it is particularly suitable yet not thoroughly studied. Recent methods for false alarm control (Fathony & Kolter, 2020; Tsoi et al., 2021) focus evaluation on small, repurposed tabular datasets, not realistic scenarios for early warning system deployment. This leaves open the critical question of performance in real-world clinical settings. Our experiments on two large electronic health records datasets suggest that our proposed objective can lead to absolute gains of 0.01-0.15 in recall while satisfying the desired precision constraint(see Sec. 5.3 and 5.4).

We view these contributions as critical steps to achieving effective early warning systems in clinical settings. We emphasize that none of our technical innovations are specific to healthcare; early warning systems in any domain can benefit from our approach.

2 BACKGROUND

When developing an early warning alarm system, our goal is to estimate a prediction model with parameters θ that can consume a feature vector $x \in \mathbb{R}^D$ and produce a real-valued score $f_\theta(x)$ indicating confidence that some (rare) outcome will occur. Large negative scores indicate certainty the outcome will not occur and large positive scores indicate certainty it will occur. Using a scalar threshold b, we can translate this score into a binary *decision* $\hat{y}(x)$, which equals 1 if $f_\theta(x) > b$ and 0 otherwise. In the intended use case, a positive decision causes the early warning system to trigger an *alarm*.

To train this model, we'll assume we have a dataset X, Y of N labeled examples: $X = \{x_n\}_{n=1}^N, Y = \{y_n\}_{n=1}^N$, with known binary label $y_n \in \{0, 1\}$ indicating which outcome happened to the example at index n with features $x_n \in \mathbb{R}^D$. In many cases, the outcome of interest (such as mortality) is rare. We'll refer to the rare label of interest as "positive" or 1, and the common label as "negative" or 0. Let N_+ denote the total count of positive true labels in a dataset: $N_+ = \sum_{n=1}^N y_n$. Similarly, N_- gives the count of negative true labels.

2.1 Evaluation metrics for binary classifiers

Many performance metrics exist for binary classifiers, each appropriate for different goals (Romero-Brufau et al., 2015). We review relevant metrics here.

Binary cross entropy (BCE) is defined as

$$\text{BCE}(\theta, X, Y) = \sum_{n=1}^{N} \log \left(\sigma(f_\theta(x_n))^{y_n} (\sigma(-f_\theta(x_n)))^{1-y_n} \right).$$

Here, $\sigma(f) = \frac{1}{1+e^{-f}}$ is the logistic sigmoid function, which maps real-valued inputs $f \in \mathbb{R}$ to the unit interval $0 \leq \sigma(f) \leq 1$. BCE is the most common loss used to train binary classifiers, motivated as a smooth upper bound on error rate (after scaling by $\frac{1}{N}$) as well as via maximum likelihood arguments. However, for problems where the positive class is rare but critical, neither error rate nor BCE can capture the key applied questions as all types of mistakes are treated equally.

True positive count. A "true positive" is an example whose true label is positive and predicted label is also positive. We define a dataset's true positive count as:

$$\text{tpc}(\theta, X, Y) = \sum_{n:y_n=1} \hat{y}_\theta(x_n) = \sum_{n:y_n=1} z(f_\theta(x_n) - b).$$

Here, $z(\cdot)$ is the zero-one function, which is one if its argument has positive sign and zero otherwise. This count is bounded: $0 \leq \text{tpc} \leq N_+$.

False positive count. A "false positive" is an example whose true label is negative but whose predicted label is positive. False positives mean the early warning system produces a false alarm. We define a dataset's false positive count as:

$$\text{fpc}(\theta, X, Y) = \sum_{n:y_n=0} \hat{y}_\theta(x_n) = \sum_{n:y_n=0} z(f_\theta(x_n) - b).$$

This count is also bounded: $0 \leq \text{fpc} \leq N_-$.

Recall. Recall is the fraction of all truly positive examples that are also called positive by the classifier:

$$\text{recall}(\theta, X, Y) = \frac{\text{tpc}(\theta, X, Y)}{N_+(Y)}, \tag{1}$$

where $N_+(Y)$ counts the positive labels in dataset Y.

Precision (aka True Alarm Rate). Precision is defined as the fraction of all positive alerts produced by the classifier that are truly positive,

$$\mathrm{prec}(\theta, X, Y) = \frac{\mathrm{tpc}(\theta, X, Y)}{\mathrm{tpc}(\theta, X, Y) + \mathrm{fpc}(\theta, X, Y)},$$

where the denominator counts all alerts (positive calls), which must either be true positive or false positive. Another name for precision is the "true alarm rate". Precision is equal to one minus the false alarm rate.

2.2 Suggested Optimization Objective

Consider a clinical prediction task trying to identify patients at risk of a rare adverse outcome, so that additional interventions can be prioritized for these patients. Interventions are a limited resource with a cost (otherwise they could be applied to all patients). At minimum, hospital staff time is limited, and thus an alarm raised for a patient who does not need extra care means that other (perhaps more needy) patients do not get attention. Furthermore, if staff become habituated to viewing alarms as rarely related to the intended outcome and thus unhelpful, they may be inclined to ignore them. Similar concerns about false alarms shape early warning systems in many other domains.

It is thus critical to view designing an effective early warning system as satisfying two goals. First, guaranteeing the false alarm rate is below some critical value established in conversation with stakeholders to ensure utility and avoid alarm fatigue. Second, achieving alerts that would successfully alarm (and hopefully help mitigate) the most possible adverse events.

These goals can be naturally formalized as the following optimization problem:

$$\max_{\theta} \ \mathrm{recall}(\theta, X, Y), \ \text{ s.t. } \mathrm{prec}(\theta, X, Y) \geq \alpha, \qquad (2)$$

where α is set to the minimum desired precision (equivalently, $1 - \alpha$ is the maximum false alarm rate). While this objective has been suggested before (Eban et al., 2017), in practice most ML early warning systems are not trained to satisfy some target minimum precision.

One could ask why we intend on constraining precision rather than recall. To answer this, we argue that our proposed precision constraint reflects the limited resources of the system (staff time and trust), while constraining recall does not. If we constrained recall to at least 85%, the system could only alarm for 85% of needy patients when 90% or 95% could be saved with little degradation of precision.

2.3 Baseline: Post-Hoc Threshold Search

After training a binary classifier to minimize cross entropy (or some other suitable objective), we can always perform a post-hoc search procedure to better identify a decision-making threshold b that meets desired performance criteria. In our applications, if we have a maximum allowable false alarm rate in mind, we can fix the parameterized score function $f_\theta(\cdot)$ obtained via training and simply select among a grid of candidate threshold values the one that best satisfies the original objective in Eq. (2). Graphically, for a linear classifier this is akin to trying all possible decision boundary hyperplanes parallel to the one produced via original training. As a one-dimensional grid search, precision and recall at each threshold can be easily computed, without any gradients or bounds needed.

This approach is helpful but far from optimal, as shown in Fig. 2, where we consider a synthetic dataset with $D = 2$ features where we wish to achieve a minimum precision of $\alpha = 0.9$. Even with this post-hoc search, the optimal linear classifier trained via BCE cannot exceed a precision of 0.68, well below the desired precision of 0.9. Later experiments show that post-hoc search also delivers sub-par results in real clinical tasks.

3 RELATED WORK

Optimization methods. Many previous efforts have focused on optimization objectives that target classifier performance metrics to try to balance precision and recall in some fashion (Rakotomamonjy, 2004; Burges et al., 2006; Yue et al., 2007; Lipton et al., 2014). As we review below, most of these methods either pursue different objectives (less appropriate for our use case than Eq. (2)), have limited scalability to large datasets, or have deficiencies in approximation quality.

Optimizing area under the precision-recall curve (AUPRC) has been pursued by Metzler & Croft (2005) and Caruana & Niculescu-Mizil (2006), as well as recently by Ramzi et al. (2021) and Qi et al. (2021), who both propose tractable surrogate losses for gradient-based optimization. However, optimizing AUPRC cannot maximize the quality of recall at a specific precision value, as our later method can. Figure B.4 in the supplement provides a real-data example of where one method dominates in AUPRC, but another (ours) achieves a better recall at the desired false alarm rate. Similar concerns exist for approaches that optimize the F-measure, such as the sigmoid approximations of Tsoi et al. (2021) or the pseudo-linear methods of Narasimhan et al. (2015) and Puthiya Parambath et al. (2014).

Fathony & Kolter (2020)'s adversarial prediction (AP) framework allows gradient-based training for a variety of non-decomposable objectives (such as our Eq. (2)). However, their AP method suffers from scalability issues, with runtime *cubic* in the number of examples in each minibatch. Our work is conceptually simpler and keeps runtime *linear* in the number of examples. Scalability is also a concern for the methods of Joachims (2005), which pursue optimizing precision at fixed recall at cost quadratic in the number of examples.

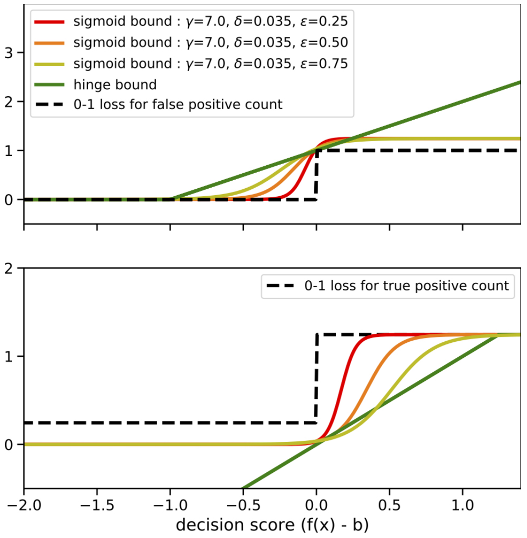

Figure 1: Illustration of the zero-one functions used to count false positives (*top*) and true positives (*bottom*), as well as bounds that make gradient-based learning feasible. In both plots, our family of sigmoid bounds is noticeably tighter than the hinge bound of Eban et al. (2017). *Bottom panel:* We bound a 0-1 loss that is *vertically-shifted* by $+\gamma\delta$, so our bound is non-negative for all inputs (unlike the hinge) and thus maintains the validity of our precision constraint as discussed around Eq. (4).

We build upon the work of Eban et al. (2017), which developed a framework for gradient-based learning of non-decomposable losses like our objective in Eq. (2). Eban et al. accomplish this via tractable hinge-loss bounds of the counts of true and false positives (as detailed in Sec. 4.1). However, we find in practice these hinge bounds are so loose (see Fig. 1) that they cannot solve the problem: solutions often do not satisfy the desired precision constraint (see Fig. 2). We will show that our tighter bounds and more careful treatment of the optimization problem lead to noticeably better solutions. Our surrogate bound ideas originally appeared in a preliminary workshop paper (Rath & Hughes, 2021). This present paper offers more rigorous conceptual justification as well as substantially expanded empirical evaluation (comparing to more alternative methods as well as more realistic "continual monitoring" tasks).

Clinical methods. Some false alarm control methods have been directly developed for use in a hospital (Chambrin (2001), Antink et al. (2016), Eerikäinen et al. (2016)). Au-Yeung et al. (2019) reduce false alarms in the ICU via post-hoc feature selection, but do not encode limits on false alarms into any training objective. Hever et al. (2019) reduce false alarms for arrhythmia by training random forests to match expert rules. However, such rules are not easily available for all tasks. Our work allows direct control of the desired precision level and applies to many models and tasks.

4 METHODS

Our ideal optimization objective given a training dataset $X, Y = \{x_n, y_n\}_{n=1}^N$ is to maximize recall subject to a minimum precision constraint, as in Eq. (2). While we can easily evaluate this objective for any candidate parameters θ, *training* our model – that is, finding good values for parameters θ given a dataset – is difficult. The key barrier is that the functions involved are based on sums of the flat zero-one function (shown in Fig. 1) and thus have gradient values equal to zero at almost all θ values. We could use gradient-free methods, but these are inefficient when θ is high-dimensional as it will be in most real settings involving many input variables. We need to transform this problem so that modern gradient-based learning will be effective.

Eban et al. (2017) suggest a promising route that leads to an unconstrained objective with non-zero gradients, which we review here and build upon. First, recognizing that maximizing recall is equivalent to maximizing the true positive count, can rewrite our problem in terms of true and false positive counts:

$$\max_\theta \operatorname{tpc}(\theta), \quad \text{subj. to: } \frac{\operatorname{tpc}(\theta)}{\operatorname{tpc}(\theta)+\operatorname{fpc}(\theta)} \geq \alpha. \tag{3}$$

Here for simplicity we write the counts as a function of parameters θ alone. While these counts do depend on the observed training data X and Y, we assume this data is known and fixed throughout training.

Next, we can rewrite our problem in an equivalent "standardized" form by framing the constraint as a function that must be less than or equal to zero, and minimizing the negative instead of maximizing:

$$\begin{aligned} \min_\theta \quad & -\operatorname{tpc}(\theta), & (4) \\ \text{subj. to:} \quad & \underbrace{-\operatorname{tpc}(\theta) + \frac{\alpha}{1-\alpha}\operatorname{fpc}(\theta)}_{g(\theta)} \leq 0. \end{aligned}$$

We emphasize that this transformation is only valid when the sum of tpc + fpc is strictly positive, as we need to multiply both sides of the constraint in Eq. (3) by this sum while preserving the intended inequality.

Using well-established optimization theory (Chong & Żak, 2013), we can transform this to an equivalent unconstrained optimization problem via the penalty method with Lagrange multiplier $\lambda > 0$:

$$\min_\theta -\operatorname{tpc}(\theta) + \lambda \max(0, g(\theta)) \tag{5}$$

Here, $g(\theta)$ is the penalty function defined in Eq. (4). If this penalty function is zero or less than zero, this means the desired constraint is *satisfied*: the precision is at least α.

This unconstrained formulation avoids the need to handle a difficult non-linear constraint, but still faces the key problem that the objective as a function of θ is *flat*. That is, infinitesimal changes in θ are unlikely to move any example's prediction from one side of the decision boundary to the other, and thus both tpc and fpc counts will remain the same for almost all small changes to θ. This flatness is a problem because it means any attempt at gradient-based training will not move the parameters θ from their original (poor performing) values.

4.1 Previous Bounds using Hinge Loss

Eban et al. (2017) obtain tractability - informative non-zero gradients at almost all possible θ parameter values - by defining non-flat *bounds* on the counts using the hinge loss function $h(y, a) = \max(0, 1 - s(y)a)$, where $a \in \mathbb{R}$ is a real value and $s(y)$ is the sign function, which returns $+1$ if $y = 1$ and -1 if $y = 0$.

First, the false positive count is upper bounded by

$$\text{fpc}^h(\theta) = \textstyle\sum_{n:y_n=0} h(y_n, f_\theta(x_n)). \tag{6}$$

Similarly, the true positive count is lower bounded by

$$\text{tpc}^h(\theta) = \textstyle\sum_{n:y_n=1} 1 - h(y_n, f_\theta(x_n)). \tag{7}$$

These bounds can replace the tpc and fpc terms in Eq. (5), which is then minimized using a gradient-based solver. These hinge bounds are visualized in Fig. 1, where we can observe two key problems. First, they are *loose* bounds: for just one term in the sum each bound can differ by 1 or more from the ideal value. Aggregated across all N_- or N_+ terms, the total error can be substantial. The looseness of the hinge bounds of Eban et al. (2017) can result in a fundamentally different optimal decision boundary than our proposed tighter sigmoid bounds (introduced later in Sec. 4.2), as illustrated in Fig 2. Second, the sum $\text{fpc}^h + \text{tpc}^h$ could be less than zero, as tpc^h may be negative. This condition would spoil the interpretation of the objective in Eq. (5) as equivalent to guaranteeing a specific minimum precision α, since we derived it assuming the denominator in Eq. (3) was strictly positive.

4.2 New Tighter Sigmoid Bounds

We now derive two *families* of functions that bound the zero-one function, one family of upper bounds (for the false positive count) and one family of lower bounds (for the true postive count). These are illustrated in Fig. 1. Both bounds are formed by shifting and scaling the logistic sigmoid function $\sigma(a) = \frac{1}{1+e^{-a}}$. Concrete bounds can be obtained by fixing tolerance hyperparameters to specific values. By varying tolerances, a user can make the bounds tighter (more accurate but with flatter gradients) or looser (more tractable). Reasonable settings deliver tighter bounds than the hinge bounds of Eban et al. (2017). Furthermore, we'll show that under all conditions our bounds meet the positivity condition required by the denominator in Eq. (3).

Upper bound on fpc. We first seek a non-flat *upper bound* on the zero-one function, denoted $u(a)$. We want this smooth function to meet the following conditions necessary for a tight upper bound:

$$\begin{aligned} u(-\infty) &\to 0 & \qquad u(-\epsilon) &\approx \delta \\ u(+\infty) &\to 1 + \gamma\delta & \qquad u(0) &\approx 1 + \delta \end{aligned} \tag{8}$$

where tolerance factors $\epsilon > 0, \delta > 0$ and scaling factor $\gamma \geq 1$ are specified by the user. Large values make the function smoother; small values make the bound tighter. In the limit as $\gamma \to 1, \epsilon \to 0, \delta \to 0$, our function converges to the zero-one function.

A natural choice for $u(a)$ is to make it a sigmoid whose amplitude is $1 + \gamma\delta$ with learnable slope and shift,

$$u_{m,b}(a) = (1 + \gamma\delta)\sigma(ma + b), \tag{9}$$

where the two parameters are slope $m \in \mathbb{R}$ and intercept $b \in \mathbb{R}$. By definition, this class of function satisfies the two limit conditions in Eq. (8). To meet the two approximation constraints, we can numerically solve for the optimal parameters that minimize squared error:

$$\hat{m}, \hat{b} = \underset{m \in \mathbb{R}, b \in \mathbb{R}}{\arg\min} (\delta - u_{m,b}(-\epsilon))^2 + (1 + \delta - u_{m,b}(0))^2.$$

Thus, for user-specified tolerance parameters γ, δ, ϵ, we can obtain a tight non-flat differentiable upper bound on the false positive count:

$$\text{fpc}^\sigma(\theta) = \textstyle\sum_{n:y_n=0} (1 + \gamma\delta)\sigma(\hat{m} f_\theta(x_n) + \hat{b}). \tag{10}$$

If we concretely select a modest setting of our tolerance parameters – $\gamma = 7.00$, $\delta = 0.035$, $\epsilon = 0.75$ – we use BFGS to solve the minimization problem and yield slope $\hat{m} = 6.85$ and shift $\hat{b} = 1.59$.

Vertically shifting tpc to enable valid lower bounds without changing the optimal parameter. When defining a lower bound for true positive counts, the bound should be strictly positive for all inputs, so that the denominator positivity condition in Eq. (3) is satisfied. First, we pick a small positive constant $\gamma\delta$ (where $\gamma > 0, \delta > 0$ are not necessarily the same values used to define fpc above). Now, we redefine the true positive count function by adding this small constant to each zero-one function:

$$\text{tpc}_{\gamma,\delta}(\theta) = \textstyle\sum_{n:y_n=1} \gamma\delta + z(f_\theta(x_n)) \tag{11}$$

This vertical shift is visualized in Fig. 1.

Instead of the original unconstrained problem in Eq. 5, we then solve a revised optimization problem:

$$\min_\theta - \text{tpc}_{\gamma,\delta}(\theta) + \lambda \max(0, g_{\gamma,\delta}(\theta)), \tag{12}$$

$$g_{\gamma,\delta}(\theta) = -\text{tpc}_{\gamma,\delta}(\theta) + \tfrac{\alpha}{1-\alpha}\text{fpc}(\theta) + \gamma\delta N_+.$$

Here the constraint function $g_{\gamma,\delta}$ differs from the original g by substituting $\text{tpc}_{\gamma,\delta}$ and adding constant $\gamma\delta N_+$.

Lemma 4.1. *Vertically shifting the true positive count does not alter the optimal parameter. Given fixed values of scalars $\lambda, \alpha, \gamma, \delta$, both Eq (5) and Eq. (12) have the* same *optimal parameter θ^*.*

Proof. For all possible θ, we have $g(\theta) = g_{\gamma,\delta}(\theta)$, because inside $g_{\gamma,\theta}$ the constant added in the $-\text{tpc}_{\gamma,\delta}$ term is exactly canceled by the extra additive constant term $\gamma\delta N_+$. Thus, the penalty terms in the two objectives will always yield the same values. The minimization objectives thus differ only by an additive constant (tpc vs. $\text{tpc}_{\gamma,\delta}$) and will have the same local optima. □

Lower bound for the shifted tpc. We now seek a lower bound ℓ of our vertically-shifted zero-one function, meeting the conditions:

$$\begin{aligned} &\ell(-\infty) \to 0 && \ell(0) \approx \delta \\ &\ell(+\infty) \to 1 + \gamma\delta && \ell(+\tilde{\epsilon}) \approx 1 + \delta \end{aligned} \quad (13)$$

Again tolerance factors $\epsilon > 0, \delta > 0, \gamma \geq 1$ are specified by the user, with similar interpretation as before. We emphasize that this lower bound ℓ is a distinct function from the upper bound u, as shown in Fig. 1. Even if we set the tolerance parameters the same for both bounds (which is not necessary but done for simplicity), the optimization desiderata in Eqs. (8) and (13) are different, yielding distinct functions ℓ and u.

To obtain a function ℓ meeting the goals of Eq. (13), we define a sigmoid function with amplitude $1 + \gamma\delta$ using the same functional form as Eq. (9). Given tolerances γ, δ, ϵ, our lower bound on the vertically-shifted true positive count becomes

$$\text{tpc}^{\sigma}_{\gamma\delta}(\theta) = \textstyle\sum_{n:y_n=1} (1 + \gamma\delta)\sigma(\tilde{m} f_\theta(x_n) + \tilde{b}). \quad (14)$$

Again, there are two parameters: slope $\tilde{m} \in \mathbb{R}$ and intercept $\tilde{b} \in \mathbb{R}$. Using the same optimization strategies as for the upper bound above, we solve for the values of $\tilde{m}, \tilde{b}$ parameters that best match the desiderata in Eq. (13). Concretely fixing tolerances to $\gamma = 7.00$, $\delta = 0.035$, $\epsilon = 0.75$, we obtain $\tilde{m} = 6.85$ and $\tilde{b} = -3.54$.

By plugging our lower bound for tpc and our upper bound for fpc into Eq. (12), we achieve a tractable objective amenable to gradient-based training. A strong advantage of using strict *bounds* as we do is the guarantee that if the optimal parameter θ^* found by optimizing Eq. (12) satisfies the constraint penalty (the g term evaluates to less than zero), then the precision will be at least α. Simply using differentiable approximations of the zero-one function instead of strict bounds would not provide such a guarantee.

4.3 Practical Implementation

Our proposed bounds allow tractable gradient-based optimization of our unconstrained objective in Eq. (12). Here we discuss practical engineering efforts that make these bounds usable on large real-world datasets.

Avoid local optima with many initializations. Linear-boundary classifiers like logistic regression (minimizing BCE) or support vector machines (minimizing hinge loss), have *convex* training problems. However, due to our sigmoid bounds even when our classifier f_θ is a generalized linear model, our objective is *non-convex* and thus sensitive to initialization. To avoid poor local optima, we take the best of many random initializations (in terms of validation-set performance), where each run is initialized via a "Glorot" procedure (Glorot & Bengio, 2010). While keeping the best of many increases runtime if runs are done sequentially, separate initializations are easily parallelized across many computers so that overall runtime is no worse than the longest individual run. Even without full parallelization, we argue our approach is *worthwhile* for the substantial gains in improved recall at the desired precision, especially because training happens much less often than prediction in a deployed alert system.

Minibatch gradient descent. If datasets are small enough, we can optimize our unconstrained objective in Eq. (12) using gradients computed from *all* examples. To scale to larger datasets we use stochastic estimates of the gradient from a minibatch and update parameters via the Adam optimizer (Kingma & Ba, 2014), trying learning rates from 0.0005 to 0.005 and selecting the best performing one on the validation set.

Using minibatches, however, means that stochastic estimates of the gradient of the $\max(0, g(\theta))$ term in Eq. (12) are not unbiased, because the maximum is performed *outside* the sum over all training examples. To mitigate this possible bias, we simply select large minibatches (500 or more examples). While not formally unbiased, in practice we find this choice is effective. We can reach high quality solutions that meet the intended precision when assessed on the entire training set. To avoid any bias, we could incrementally update per-batch gradients as well as their sum over iterations for a predefined set of batches, an approach known as Stochastic Average Gradient (Schmidt et al., 2017).

Batch size and penalty hyperparameters. In the supplement, we study the sensitivity of performance to batch size. Batches of 512 or 1024 examples are preferred over larger batches. We found that trying two λ values, 1000 and 10000, was sufficient to meet the desired false alarm constraint ($g(\theta) \leq 0$ is satisfied after training). We use these λ values in all experiments.

Runtime cost analysis. Our method's training cost will scale *linearly* in the number of examples in each batch. Computing our unconstrained objective in Eq. (12) at a specific θ can be done by evaluating each of the terms fpc^{σ} and $\text{tpc}^{\sigma}_{\gamma,\delta}$ once, which requires just one sigmoidal bound function evaluation ($\ell(\cdot)$ or $u(\cdot)$) for each example (see (10) and (14)). Computing the *gradient* has the same linear complexity as the loss itself, using well-known complexity results for back-propagation.

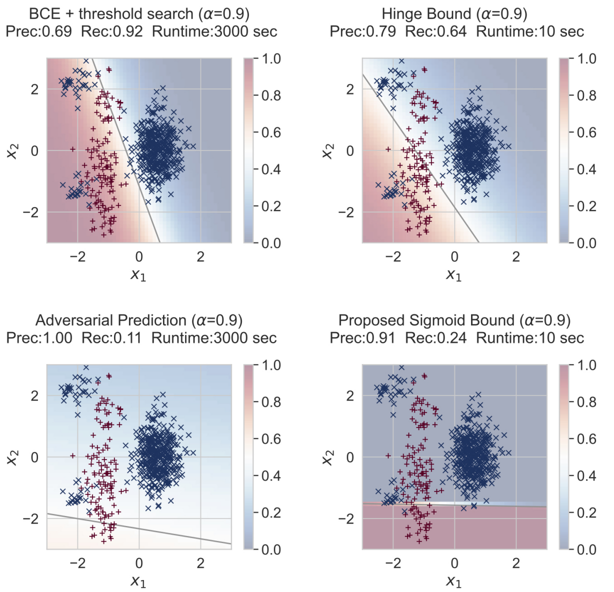

Figure 2: Linear decision boundaries (black lines) found by various methods on toy data when stated goal is to maximizes recall subject to a minimum precision of $\alpha = 0.9$. Subtitles report precision, recall, and total runtime for each method's solution. Red and blue markers represent positive and negative examples, respectively, for the toy dataset used for training (Sec. 5.1). *Top Left*: Even with post-hoc search, binary cross entropy (BCE) cannot exceed a precision of 0.684. *Top Right*: Despite including the desired $\alpha = 0.9$ precision constraint in its objective, Eban et al. (2017)'s hinge bound cannot satisfy the constraint. *Bottom*: Both Fathony & Kolter (2020)'s AP method and our proposed sigmoid bound satisfy the precision constraint; ours delivers 2x the recall of the AP solution in $1/300^{th}$ of the time.

Chosen objective in practice. In pursuit of our ideal but intractable objective in Eq. (2), optimizing our tractable unconstrained objective in Eq. (12) is guaranteed to produce a valid solution to the original objective if the penalty function $g_{\gamma,\delta}$ is less than or equal to zero. In practice, we find that often optimizing Eq. (12) is too strict and does not yield a satisfactory result. However, another strategy, dropping the additive constant $\gamma\delta N_+$ term in $g_{\gamma,\delta}$ and then post-hoc verifying the desired precision constraint is satisfied, works well. We recommend trying both ways and using validation-set performance to select the best.

5 EXPERIMENTAL EVALUATION

5.1 Results on Synthetic Data

To gain insight, we designed a toy classification task with 2 features where the solutions favored by different objectives can be easily visualized. Our goal is to train a logistic regression (LR) classifier whose decisions maximize recall subject to a minimum precision constraint of $\alpha = 0.9$. We illustrate the advantage of our logistic sigmoid bound empirically by comparing it with 5 other objectives. Fig. 2 shows all training examples ($N_+ = 120, N_- = 450$) together with the learned linear decision boundaries for four possible optimization strategies: BCE with post-hoc threshold selection, Eban et al. (2017)'s hinge-bound approach, Fathony & Kolter (2020)'s adversarial prediction (AP) method, and our proposed sigmoid bound approach. We further compare 2 other methods, class-weighted cross entropy and a log loss surrogate bound in Sec.B.2 in the appendix.

Although the BCE solution might look reasonable, recall that our specific goal is to maximize recall subject to a precision of at least 0.9. The BCE solution gets only 0.69 precision even with post-hoc threshold search, and thus remains quite far from the desired precision of 0.9. Similarly, the hinge bound achieves only 0.79 precision, also falling short of our goal. Both the AP method and our approach can satisfy the desired precision constraint. However, our proposed sigmoid bound achieves noticeably better recall of 0.24, which is over double the 0.11 recall found by AP.

To investigate bound quality, we verified that even after initializing at the sigmoid bound's better solution, optimizing Eq. (5) using the hinge bound prefers to wander away, suggesting the inadequacy of this loose bound (see our code repository for a reproducible example).

5.2 Results on Semi-Synthetic Data

Real-world prediction tasks often include many features, most of which are only weakly relevant to the outcome. To assess our method's robustness to such features, we designed a *semi-synthetic* task. Adapting the toy generation process that produced the 2-dim. feature vectors x and binary labels y in Fig. 2, we append to each x vector a 98-dimensional feature vector representing vitals, labs, and demographics for one randomly-chosen patient-stay in the MIMIC-III dataset (Johnson et al., 2016), generating 684 train, 513 valid, and 512 test examples. Our goal is to enforce a minimum precision of $\alpha = 0.8$. We deliberately pursue a slightly lower α here than in Sec. 5.1 to illustrate the flexibility of our methods to meet any desired precision constraint. Naturally, lowering the required minimum precision leads to better recall compared to the results in Fig. 2. To avoid local optima, we run our method many times (25 random seeds, 2 initialization scale factors, and 2 λ values), selecting the run that maximizes validation recall while meeting the precision constraint.

Table 1 shows that in this higher-dimensional problem (100 total features), only our method can meet the desired 0.8 precision even on the training set (others range from 0.63-0.73). Our method also exceeds others on test-set recall by at least 0.05. In terms of runtime, most methods are fast (15 seconds), but the AP method (Fathony & Kolter, 2020) shows poor scalability, requiring 2.5 hours due to its cubic scaling. This

	Precision			Recall			Runtime
Method	Train	Valid	Test	Train	Valid	Test	in sec.
BCE + threshold search	0.70	0.69	0.69	0.61	0.72	0.61	15
Hinge Bound	0.63	0.61	0.63	0.72	0.75	0.71	15
Adversarial Prediction (AP)	0.73	0.67	0.75	0.55	0.66	0.60	9000
Sigmoid Bound (ours)	0.80	0.72	0.74	0.67	0.77	0.76	15

Table 1: Precision-recall performance on the semi-synthetic task (Sec. 5.2), where the goal is to maximize recall subject to a minimum precision of $\alpha = 0.8$. Only our proposed sigmoid bound satisfies this goal on the training set. Our method also delivers better recall on the test set than all alternatives by at least 0.05 while keeping runtime fast.

poor scalability prevented our use of the AP method on the larger real-world tasks below.

5.3 Per-Sequence Mortality Prediction

We now assess our method's precision-recall performance at mortality risk predictions on two large open-access datasets of critical care electronic health records (EHR): MIMIC-III and eICU. Our goal is to predict in-hospital mortality after observing vitals and labs over the entire sequence of a patient's stay, up to just before discharge or death. We enforce a minimum precision value of $\alpha = 0.9$ on the training set. We found $\alpha = 0.9$ challenging to meet on heldout data even when satisfied at training, so to select the best runs we enforce a minimum validation precision $\alpha' = 0.8$.

MIMIC-III data. The MIMIC-III dataset (Johnson et al., 2016) contains deidentified open-access EHR data from a Boston hospital from 2002-2012. We extract 2 demographics, 10 vitals, and 94 lab measurements discretized to hourly bins for 34472 patient stays using the MIMIC-Extract pipeline (Wang et al., 2020b). Our train/valid./test splits have 20682/6895/6895 patient-stays, with $\sim$ 9.5% patient stays resulting in death. Each patient-stay's multivariate time-series is transformed into a feature vector as follows. For each raw feature, we apply 7 summarization functions (missing indicator; time since last non-missing; plus min, max, median, slope, and variance of non-missing values) to 4 possible time windows (0-100%, 50%-100%, last 16 hr, and last 24 hr). The resulting feature vector has size 106x7x4=2968.

eICU data. The eICU Collaborative Research Database (Pollard et al., 2018) contains data from 59 critical care units throughout the U.S. We extract 3 demographics, 8 vitals, and 6 lab measurements discretized to hourly bins using eICU Extract (Wang et al., 2020a) for 72670 patient stays. Our train/valid/test splits have 43642/14509/14518 patient-stays, with $\sim$ 8.2% resulting in death. We featurize each patient-stay the same way as MIMIC, yielding feature vectors of size 17x7x4=417.

Models. On both datasets, we tried two models: logistic regression (LR) and a multi-layer perceptron (MLP) with 1 hidden layer (32 hidden units; RELU activation). We search 25 random seeds, 2 initialization scale factors, 2 λ values, and 5 weight decays. Details for reproducibility are in the appendix.

Mortality Prediction Results. The recall performance of each method on both datasets can be found in Table 2. Across 20 independent randomly-chosen train-test partitions of both datasets, we find that all methods can satisfy the desired precision constraint (all methods exceed 0.9 on training and 0.8 on validation). However, our proposed sigmoid bound method consistently achieves better test-set recall values at the selected operating threshold (chosen on validation data). Compared to the BCE baseline, using a linear model we see our method boost the test-set recall (median across partitions) from 0.540 to 0.684 on MIMIC and from 0.087 to 0.195 on eICU. As expected, gains from the MLP are smaller (as a predictor gets more flexible, it can get closer to the optimal non-linear boundary). However, the gains are still noticeable: recall improves from 0.689 to 0.717 on MIMIC and from 0.280 to 0.298 on eICU. Eban et al. (2017)'s hinge bound method is *worse* than the BCE baseline in terms of test-set recall on three of the four dataset-model combinations tested.

Across Table 2, recall values in on eICU are noticeably lower than MIMIC (BCE achieves test recall of 0.5-0.7 on MIMIC but only 0.08-0.28 for eICU). This occurs despite the desired minimum precision value remaining the same ($\alpha = 0.9$) and being satisfied in all cases. The drop in recall performance likely occurs because we have more available informative clinical measurements for MIMIC than eICU (106 vs. 17). The eICU task is also slightly more difficult: only 8% of eICU sequences have positive labels compared to 9.5% for MIMIC. In this challenging setting, enforcing high precision means some modest tradeoffs with recall. We emphasize that without enforcing sufficiently high precision, alarm fatigue could render the whole system ineffective.

To better understand the gains of our method in terms of common evaluation criteria, we present ROC and PR curves for all methods in the appendix. Our method shows consistent gains in both curves.

5.4 Deployment Scenario for Mortality Alerts

In order to realistically assess our proposed method as an early warning alert for patients at risk of deterioration, we consider now evaluating predictions made *every 12 hours* throughout a patient's stay, with the outcome set to in-hospital mortality *within the next 24 hours*. For a patient stay lasting just over two days, we would create and evaluate four feature vectors, representing predictions made after 12, 24, 36, and 48 hours. This realistic setting exaggerates class imbalance (only 3% of examples are positive). To compensate, we lower

		Recall on MIMIC		Recall on eICU	
	Method	Train	Test	Train	Test
Logistic Regression	BCE + threshold search	.572 (.566, .580)	.540 (.532, .551)	.100 (.096, .104)	.087 (.082, .092)
	Hinge Bound (Eban et al.)	.696 (.691, .704)	.625 (.612, .635)	.018 (.015, .019)	.016 (.012, .018)
	Sigmoid Bound (ours)	.763 (.755, .766)	.684 (.675, .696)	.211 (.206, .218)	.195 (.188, .206)
1-layer MLP	BCE + threshold search	.766 (.757, .772)	.689 (.676, .708)	.325 (.321, .330)	.280 (.271, .290)
	Hinge Bound (Eban et al.)	.741 (.734, .750)	.640 (.620, .654)	.359 (.355, .363)	.235 (.229, .246)
	Sigmoid Bound (ours)	.821 (.817, .827)	.717 (.705, .729)	.385 (.379, .392)	.298 (.288, .305)

Table 2: Precision-recall performance for per-sequence in-hospital mortality prediction on MIMIC-III and eICU (Sec. 5.3), using target precision $\alpha = 0.9$ on train and $\alpha' = 0.8$ on validation. We show the $50^{th}(5^{th}, 95^{th})$ percentiles from 20 randomly-drawn train/validation/test partitions. While all methods meet the desired precision constraint after threshold search, our sigmoid bound method achieves better recall on test data, especially for logistic regression models.

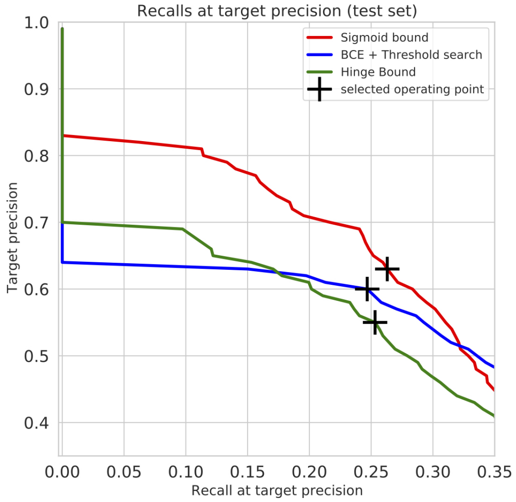

Figure 3: Predicting in-hospital mortality on MIMIC in a deployable scenario (making a prediction every 12 hours), using target precisions of $\alpha = 0.7$ on train and $\alpha' = 0.6$ on valid. Our proposed sigmoid bound improves precision by 0.026 (100's of fewer false alarms across the 6895 patient-stays in the test set), and improves recall by 0.01 (100's more truly at-risk patients identified).

the desired minimum precision to $\alpha = 0.7$ on train and $\alpha' = 0.6$ on validation. For simplicity, we featurize by applying the same summary functions used above on 2 windows (0-100% and 90-100%).

Fig. 3 illustrates the precision-recall tradeoff on test data for LR models in this scenario. At the chosen operating threshold (marked as a cross), our proposed sigmoid bound achieves better recall and precision than alternatives. We show a similar improvement in recall and precision in the deployment scenario for eICU in Fig. B.5 in the appendix.

6 DISCUSSION AND CONCLUSION

We have developed new bounds based on the sigmoid function that allow tractable gradient-based optimization of any differentiable classifier to meet a desired precision constraint. Our experiments evaluating in-hospital mortality prediction on two large hospital datasets suggest the feasibility of our approach in real early warning applications.

Limitations. A key drawback of our method is vulnerability to local optima due to non-convexity of the sigmoid bound. Given reasonable compute power at training time, we find we can mitigate local optima by taking the best of many runs.

For a deployment, setting α requires input from stakeholders (at what false alarm rate would alerts be ignored or not add value?). As in most real-world ML, we expect a modest but noticeable drop between train and heldout performance. As a rule of thumb, if the desired heldout precision is 0.5, we recommend setting α a bit higher (say $\alpha = 0.6$) for training then enforce $\alpha' = 0.5$ on a validation set to be sure desired performance is achieved. Our method is sensitive to other hyperparameters, including the penalty $\lambda > 0$ and the tolerances γ, δ, ϵ that govern the tractability-tightness tradeoff for our bound. While we found reasonable values for our tasks, future applications may need to tune values for better performance.

Finally, although our objective can control false alarms on the training set, this does not guarantee this constraint will be satisfied on heldout data. Guaranteeing generalization is an open research question.

Advantages. The primary advantage of our method is the ability to steer models to satisfy a maximum desired false alarm rate, in order to avoid alarm fatigue, while maintaining linear runtime. As demonstrated throughout our experiments, our objective clearly outperforms alternatives like post-hoc threshold search at maximizing recall at a desired minimum precision.

Future work could use our bounds on more flexible classifiers that are trainable via SGD, such as recurrent NNs, convolutional NNs, or graph NNs. For improved interpretability, one could explore integrating our loss with L0 and L1 norm weight sparsity penalties (Tibshirani, 1996; Ustun & Rudin, 2016).

Acknowledgements

This research was sponsored in part by the U.S. Army DEVCOM Soldier Center, and was accomplished under Cooperative Agreement Number W911QY-19-2-0003. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army DEVCOM Soldier Center, or the U.S. Government. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

We also acknowledge support from the U.S. National Science Foundation under award HDR-1934553 for the Tufts T-TRIPODS Institute. Michael C. Hughes is supported in part by NSF IIS-1908617.

References

Al Banna, M. H., Taher, K. A., Kaiser, M. S., Mahmud, M., Rahman, M. S., Hosen, A. S., and Cho, G. H. Application of artificial intelligence in predicting earthquakes: state-of-the-art and future challenges. *IEEE Access*, 8:192880–192923, 2020.

Antink, C. H., Leonhardt, S., and Walter, M. Reducing false alarms in the icu by quantifying self-similarity of multimodal biosignals. *Physiological measurement*, 37(8):1233, 2016.

Au-Yeung, W.-T. M., Sahani, A. K., Isselbacher, E. M., and Armoundas, A. A. Reduction of false alarms in the intensive care unit using an optimized machine learning based approach. *NPJ digital medicine*, 2(1): 1–5, 2019.

Burges, C., Ragno, R., and Le, Q. Learning to rank with nonsmooth cost functions. *Advances in neural information processing systems*, 19:193–200, 2006.

Caruana, R. and Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 161–168, 2006.

Chambrin, M.-C. Alarms in the intensive care unit: how can the number of false alarms be reduced? *Critical Care*, 5(4):1–5, 2001.

Chong, E. K. P. and Żak, S. H. Chapter 23: Algorithms for Constrained Optimization. In *An Introduction to Optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, New York, fourth edition edition, 2013.

Cvach, M. Monitor alarm fatigue: an integrative review. *Biomedical instrumentation & technology*, 46(4):268–277, 2012.

Deb, S. and Claudio, D. Alarm fatigue and its influence on staff performance. *IIE Transactions on Healthcare Systems Engineering*, 5(3):183–196, 2015.

Eban, E., Schain, M., Mackey, A., Gordon, A., Rifkin, R., and Elidan, G. Scalable Learning of Non-Decomposable Objectives. In *Artificial Intelligence and Statistics*, pp. 832–840, 2017. URL http://proceedings.mlr.press/v54/eban17a.html.

Eerikäinen, L. M., Vanschoren, J., Rooijakkers, M. J., Vullings, R., and Aarts, R. M. Reduction of false arrhythmia alarms using signal selection and machine learning. *Physiological measurement*, 37(8): 1204, 2016.

Escobar, G. J., Liu, V. X., Schuler, A., Lawson, B., Greene, J. D., and Kipnis, P. Automated identification of adults at risk for in-hospital clinical deterioration. *New England Journal of Medicine*, 383(20): 1951–1960, 2020.

Fathony, R. and Kolter, Z. AP-perf: Incorporating generic performance metrics in differentiable learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 4130–4140. PMLR, 2020.

Futoma, J., Hariharan, S., and Heller, K. Learning to Detect Sepsis with a Multitask Gaussian Process RNN Classifier. In *International Conference on Machine Learning*, 2017. URL http://arxiv.org/abs/1706.04152.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Hever, G., Cohen, L., O'Connor, M. F., Matot, I., Lerner, B., and Bitan, Y. Machine learning applied to multi-sensor information to reduce false alarm rate in the icu. *Journal of clinical monitoring and computing*, pp. 1–14, 2019.

Hyland, S. L., Faltys, M., Hüser, M., Lyu, X., Gumbsch, T., Esteban, C., Bock, C., Horn, M., Moor, M., et al. Early prediction of circulatory failure in the intensive care unit using machine learning. *Nature Medicine*, 26(3):364–373, 2020.

Joachims, T. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 377–384, 2005.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., et al. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3, 2016.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, 2014. URL http://arxiv.org/abs/1412.6980.

Lipton, Z. C., Elkan, C., and Naryanaswamy, B. Optimal thresholding of classifiers to maximize F1 measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 225–239. Springer, 2014.

Metzler, D. and Croft, W. B. A markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 472–479, 2005.

Mousavi, S. M., Zhu, W., Sheng, Y., and Beroza, G. C. Cred: A deep residual network of convolutional and recurrent units for earthquake signal detection. *Scientific reports*, 9(1):1–14, 2019.

Narasimhan, H., Kar, P., and Jain, P. Optimizing non-decomposable performance measures: A tale of two classes. In *International Conference on Machine Learning*, pp. 199–208. PMLR, 2015.

Pollard, T. J., Johnson, A. E., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5 (1):1–13, 2018.

Puthiya Parambath, S., Usunier, N., and Grandvalet, Y. Optimizing f-measures by cost-sensitive classification. *Advances in Neural Information Processing Systems*, 27:2123–2131, 2014.

Qi, Q., Luo, Y., Xu, Z., Ji, S., and Yang, T. Stochastic optimization of areas under precision-recall curves with provable convergence. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=Q_64PF6XNut.

Rakotomamonjy, A. Optimizing area under roc curve with SVMs. In *ROCAI*, pp. 71–80, 2004.

Ramzi, E., THOME, N., Rambour, C., Audebert, N., and Bitot, X. Robust and decomposable average precision for image retrieval. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=VjQw3v3FpJx.

Rath, P. and Hughes, M. C. Optimizing clinical early warning models to meet false alarm constraints. In *1st Workshop on Interpretable Machine Learning in Healthcare (IMLH)*, 2021.

Romero-Brufau, S., Huddleston, J. M., Escobar, G. J., and Liebow, M. Why the C-statistic is not informative to evaluate early warning scores and what metrics to use. *Critical Care*, 19(1), 2015.

Schmidt, M., Le Roux, N., and Bach, F. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Sendak, M. P., Ratliff, W., Sarro, D., Alderton, E., Futoma, J., Gao, M., Nichols, M., Revoir, M., Yashar, F., et al. Real-World Integration of a Sepsis Deep Learning Technology Into Routine Clinical Care: Implementation Study. *JMIR Medical Informatics*, 8 (7), 2020.

Sendelbach, S. and Funk, M. Alarm fatigue: a patient safety concern. *AACN advanced critical care*, 24(4): 378–386, 2013.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Tsoi, N., Milkessa, Y., and Vázquez, M. A heaviside function approximation for neural network binary classification. *CoRR*, abs/2009.01367, 2021. URL https://arxiv.org/abs/2009.01367.

Ustun, B. and Rudin, C. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

Wang, S., McDermott, M. B., Chauhan, G., Ghassemi, M., Hughes, M. C., and Naumann, T. Code for eicu extract. https://github.com/MLforHealth/MIMIC_Extract/tree/eICU_Extract, 2020a.

Wang, S., McDermott, M. B. A., Chauhan, G., Ghassemi, M., Hughes, M. C., and Naumann, T. MIMIC-Extract: A data extraction, preprocessing, and representation pipeline for MIMIC-III. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, CHIL '20, pp. 222–235, New York, NY, USA, 2020b. Association for Computing Machinery.

Wellner, B., Grand, J., Canzone, E., Coarr, M., Brady, P. W., Simmons, J., Kirkendall, E., Dean, N., Kleinman, M., et al. Predicting Unplanned Transfers to the Intensive Care Unit: A Machine Learning Approach Leveraging Diverse Clinical Elements. *JMIR Medical Informatics*, 5(4), 2017.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. A support vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 271–278, 2007.

Supplementary Material: Optimizing Early Warning Classifiers to Control False Alarms via a Minimum Precision Constraint

Abstract

This supplement includes additional experimental results and information for understanding and reproducing experiments. Section B provides additional results, including further toy example results, ROC curves and precision-recall curves for all tested methods on the hospital mortality risk datasets (Sec. B.3), results examining sensitivity to hyperparameters(Sec. B.5), and a demonstration of precision improving over time (Sec. B.6). Details about the real-world datasets used in our experiments are in Sec. C, and reproducible details about experimental protocols are in Sec. D.

Contents

A CODE AVAILABILITY

Our open-source code is available for browsing here: `https://github.com/tufts-ml/false-alarm-control`.

This current release includes a Jupyter notebook for exploring our proposed method (and alternatives) on the toy dataset from the main paper.

B ADDITIONAL RESULTS

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

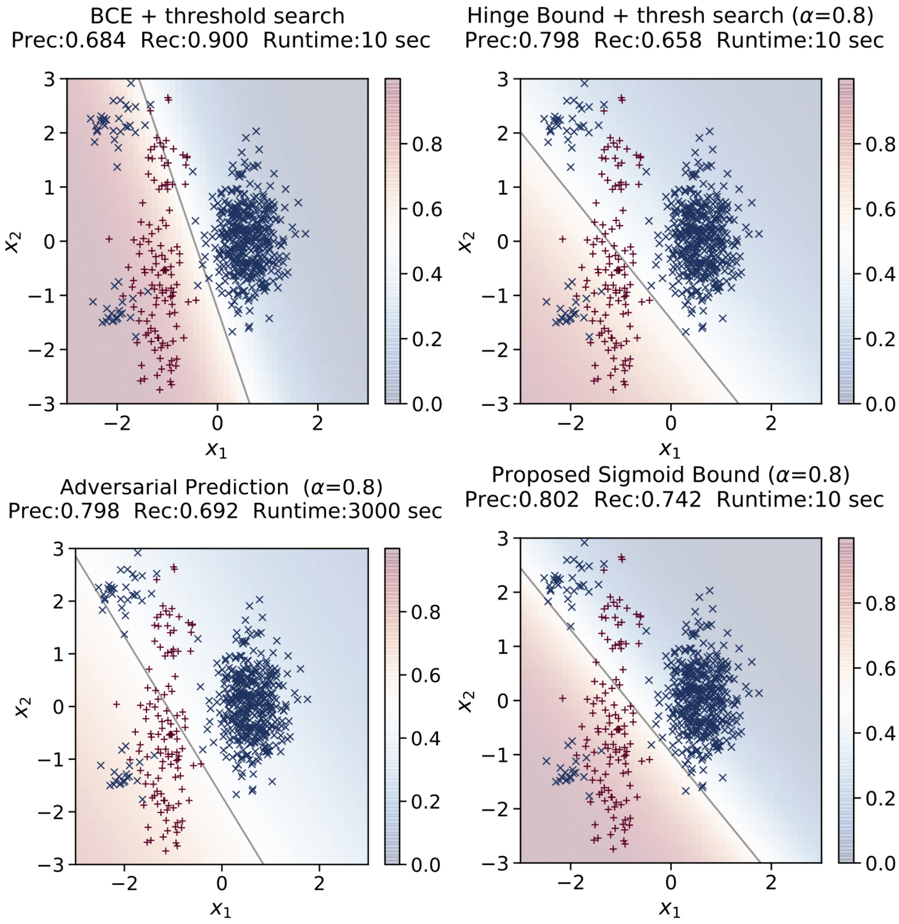

Figure B.1: **Further results on the synthetic task from our paper's Sec. 5.1. Here we set the desired precision value to** $\alpha = 0.8$ (Fig. 2 in the main paper used $\alpha = 0.9$). Each plot shows the linear decision boundary (black line) found using a specific objective, with red and blue points represent positive and negative examples respectively. *Top*: BCE and Eban et al. (2017)'s hinge bound with post-hoc selected threshold. BCE + threshold search is unable to exceed a precision of 0.684, while hinge bound comes just short of the desired precision of 0.8, reaching 0.798 instead. *Bottom Left*: Optimizing the adversarial objective proposed by Fathony & Kolter (2020) almost meets the desired precision of 0.8 but has a substantially longer training time (30x) than our proposed sigmoid bound. *Bottom Right*: Our proposed sigmoid bound is the only one that satisfies the desired minimum precision requirement of 0.8, while simultaneously achieving better recall than the other bounds.

B.1 Further Results on Toy Example

In Sec. 5.1 of the main paper, we presented a toy dataset where our method achieved better recall at a desired precision of at least $\alpha = 0.9$ than other methods. We now wish to show the same strong results hold in other settings; there is nothing special about our choice of $\alpha = 0.9$.

Here, using the same dataset, we include study a different setting, when the desired minimum precision constraint of $\alpha = 0.8$. Fig. B.1 shows results from our proposed sigmoid bound method and the same alternative methods: minimizing BCE with post-hoc threshold selection, Eban et al. (2017)'s hinge-bound approach, Fathony & Kolter (2020)'s adversarial prediction (AP) method).

In this additional $\alpha = 0.8$ setting, again even with post-hoc search, BCE cannot achieve the desired precision. The hinge bound also comes just short of the desired precision of 0.8. Optimizing the adversarial objective proposed by Fathony & Kolter (2020) almost meets the desired precision of 0.8 but has a substantially longer training time (30x) than our proposed sigmoid bound. In terms of recall, our proposed sigmoid bound achieves noticeably better recall of 0.74, compared to AP's 0.69 and the Hinge bound's 0.65.

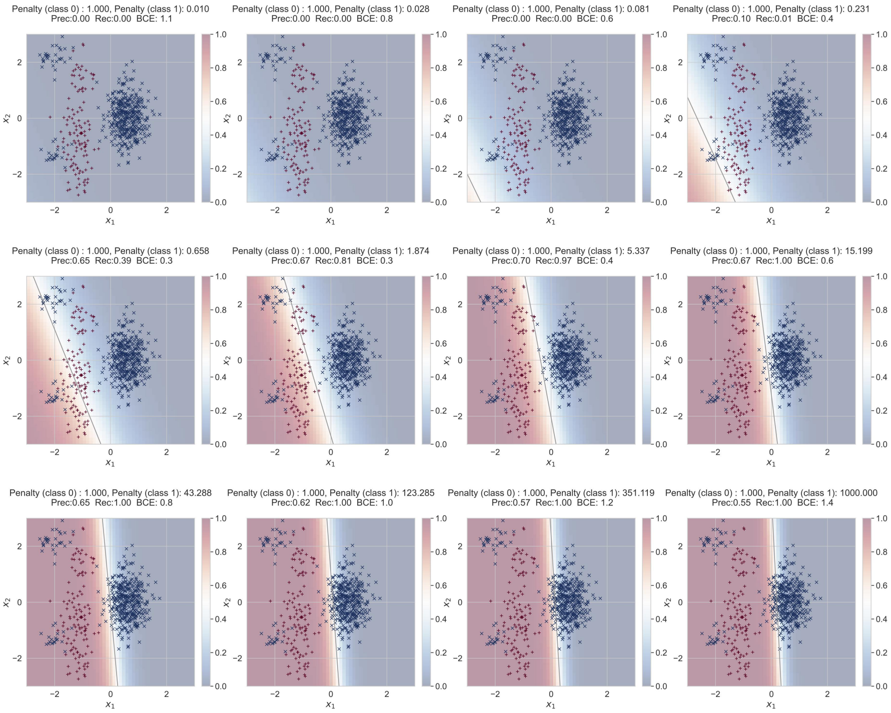

Figure B.2: We tried a **weighted cross entropy** classifier, varying the cost of positive examples (R7) across 12 log-spaced values from 0.01 to 1000. None of these met the desired precision constraint of 0.9 (the best of these got precision 0.7)

B.2 Comparison to other possible bounds

We tried 2 alternative methods on our toy 2D example from Sec. 5.1. First, we tried a **weighted cross entropy** classifier, varying the cost of positive examples (R7) across 12 log-spaced values from 0.01 to 1000 (See figure B.2). None of these met the desired precision constraint of 0.9 (the best of these got precision 0.7). Second, we tried a **log loss** surrogate bound (R5) (instead of the hinge in Eban et al's approach). Again, this did not meet our desired 0.9 precision (it got precision=0.72, recall=0.56).

B.3 ROC and PR curves for Sequence Level Predictions of Mortality Risk

In Fig. B.3,and Fig. B.4, we provide the full receiver operating curve and precision-recall curve for all possible objectives for training logistic regression on MIMIC and eICU. We see that our proposed method delivers noticeable gains especially in the precision-recall curve, which we argue is more suitable to early warning systems.

B.4 Deployment Scenario for Mortality Alerts (eICU)

In order to assess our proposed method realistically as an early warning alert for patients at risk of deterioration, we consider now evaluating predictions made *every 12 hours* throughout a patient's stay, with the outcome set to in-hospital mortality *within the next 24 hours*. For a patient stay lasting just over two days, we would create and evaluate four feature vectors, representing predictions made after 12, 24, 36, and 48 hours. This realistic setting exaggerates class imbalance (only 2.3% of examples are positive). To compensate, we lower the desired minimum precision to $\alpha = 0.7$ on train and $\alpha' = 0.6$ on validation. For simplicity, we featurize by applying the same summary functions used above on 2 windows (0-100% and 90-100%).

Fig. B.5 illustrates the precision-recall tradeoff on test data for LR models in this scenario. At the chosen operating point (marked as a cross), our proposed sigmoid bound achieves better recall and precision than alternatives.

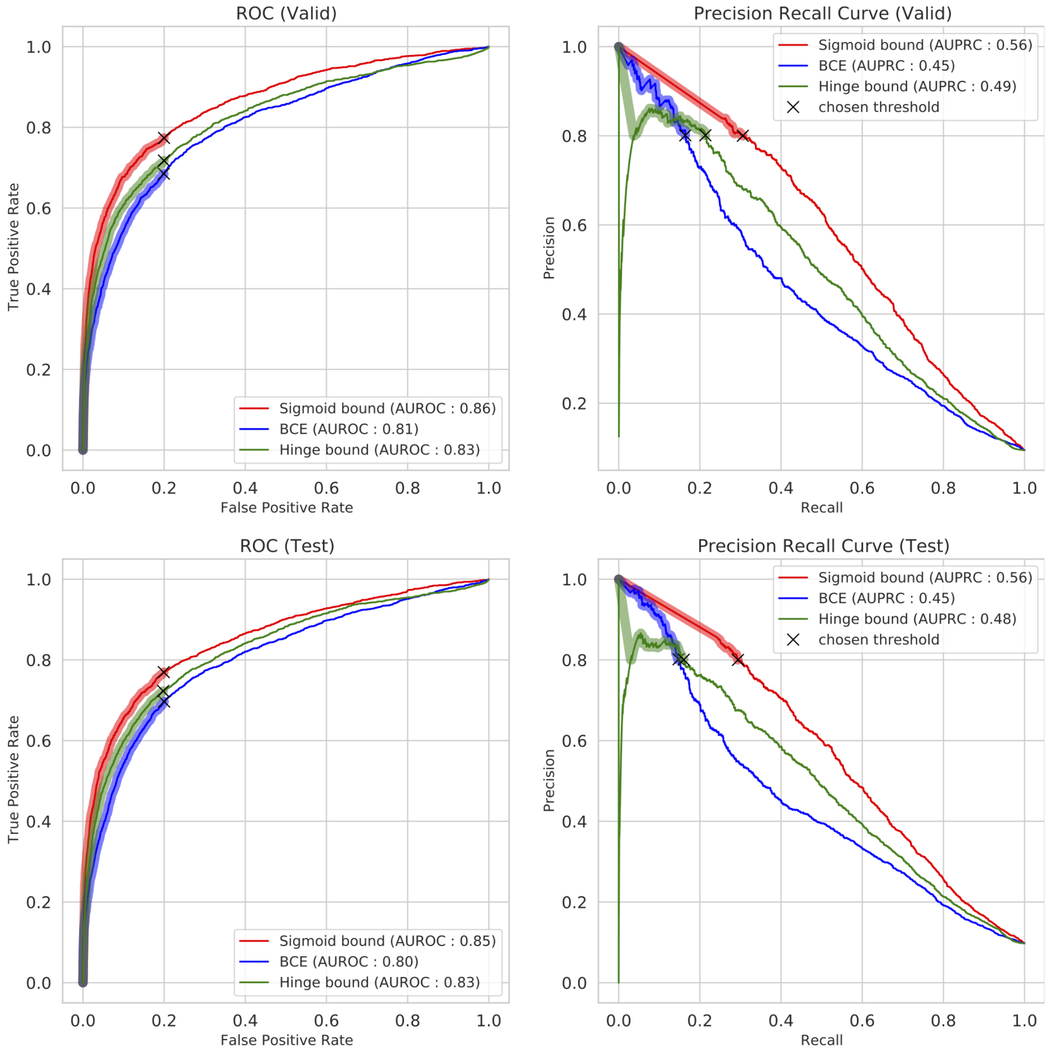

Figure B.3: **Receiver operating curve (ROC, left column) and precision-recall curve (PR, right column) for logistic regression models trained using different optimization objectives on the MIMIC dataset** (top row: validation set; bottom row: test set). Each curve shows tradeoffs in performance metrics possible while varying the decision threshold b of a single method. *Left column:* Receiver operating curve (ROC), comparing true positive rate (TPR, y-axis) to false positive rate (FPR, x-axis). *Right column:* Precision-recall curve, comparing precision (also known as positive predictive value (PPV), y-axis) to recall (also known as TPR, x-axis). Our sigmoid bound and Eban et al. (2017)'s hinge bound were trained to maximize recall subject to a constraint on precision: above 0.9 on training set; above 0.8 on validation set. *Darker line segments* indicate thresholds satisfying the desired precision above 0.8 on validation set. *The black "X" marker* indicates the selected operating point for each method (chosen to maximize our optimization objective on the validation set).

B.5 Sensitivity to Batch Size

We study our method's sensitivity to batch size in Fig. B.6, by plotting the precision and recall achieved after convergence for many random initializations at many possible settings of batch size (from 512 up to the entire dataset), while training on the MIMIC mortality risk prediction task in the main paper. From Fig. B.6, we conclude that while all batch sizes have some initializations outperform others in terms of precision and recall, overall the best runs using batch sizes of 512 (orange dots) and 1024 (green dots) seem to reach higher precision and recall values than the best runs at other batch sizes. Only runs with 512 and 1024 have precision exceed 0.25 on heldout data (validation and test). We suspect that the "folk wisdom" that small batches add helpful "noise" to stochastic gradient descent in order to avoid bad local optima may be the reason that 512 and 1024 work better than the larger tested batch sizes.

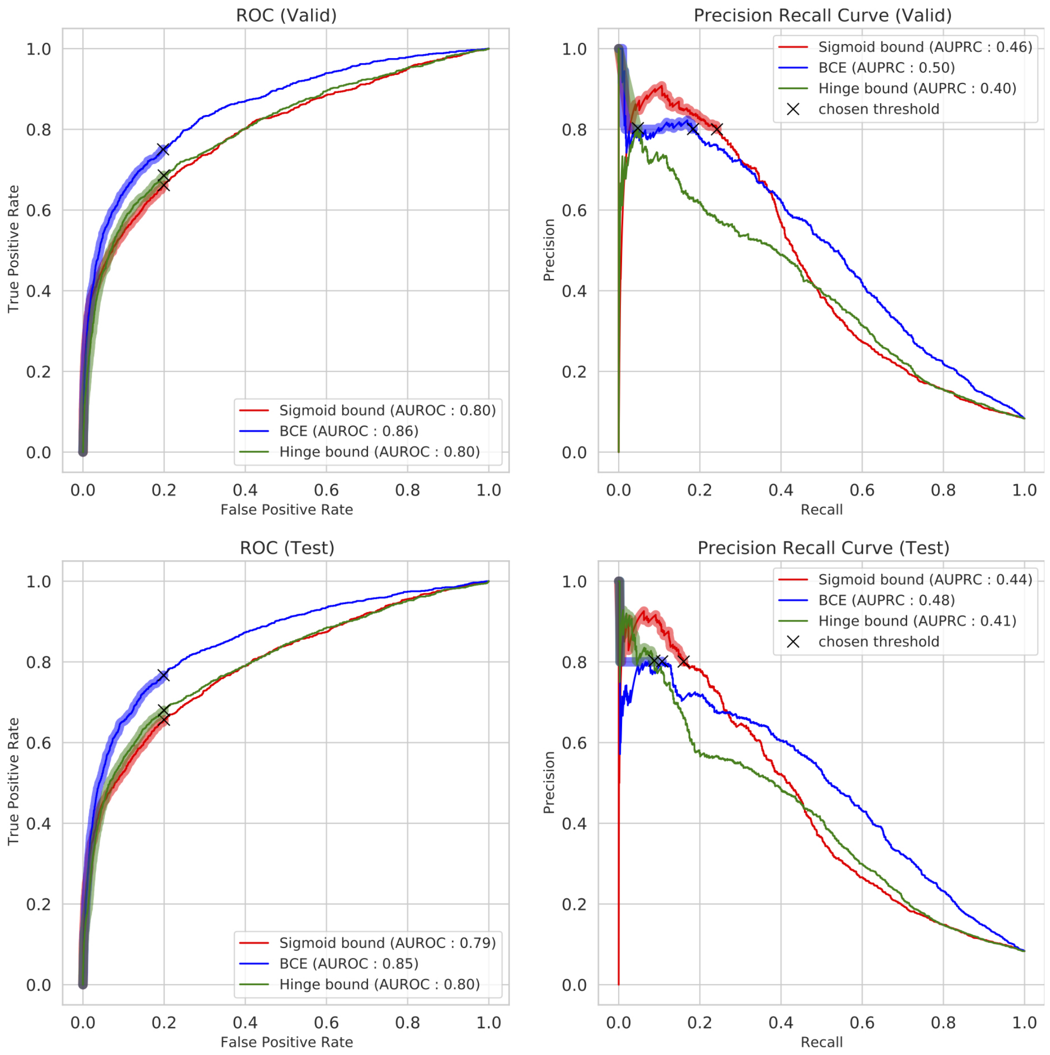

Figure B.4: **Receiver operating curve (ROC, left column) and precision-recall curve (PR, right column) for logistic regression models trained using different optimization objectives on the eICU dataset** (top row: validation set; bottom row: test set). Each curve shows tradeoffs in performance metrics possible while varying the decision threshold b of a single method. *Left column:* Receiver operating curve (ROC), comparing true positive rate (TPR, y-axis) to false positive rate (FPR, x-axis). *Right column:* Precision-recall curve, comparing precision (also known as positive predictive value (PPV), y-axis) to recall (also known as TPR, x-axis). Our sigmoid bound and Eban et al. (2017)'s hinge bound were trained to maximize recall subject to a constraint on precision: above 0.9 on training set; above 0.8 on validation set. *Darker line segments* indicate thresholds satisfying the desired precision above 0.8 on validation set. *The black "X" marker* indicates the selected operating point for each method (chosen to maximize our optimization objective on the validation set).

B.6 Demonstration of Models Satisfying Precision Constraint during Training

In figure B.7, we demonstrate our model's ability to ramp up to any desired precision during training. While training the MLP for sequence level predictions on MIMIC, we gradually step up the precision constraint (α) to 0.9 at every 50 epochs, and monitor the precision and recall. Consequently, we see that the model's empirical precision also improved to approximately satisfy the precision constraint (α). Note that ramping up the precision constraint gradually over epochs is only shown for visual validation of our models, and is not the optimal strategy during training, due to local optima at intermediate precision constraints. Training with a fixed precision constraint (α), using many random initializations, is the optimal strategy to train the models with our sigmoid bound.

C DATASET DETAILS

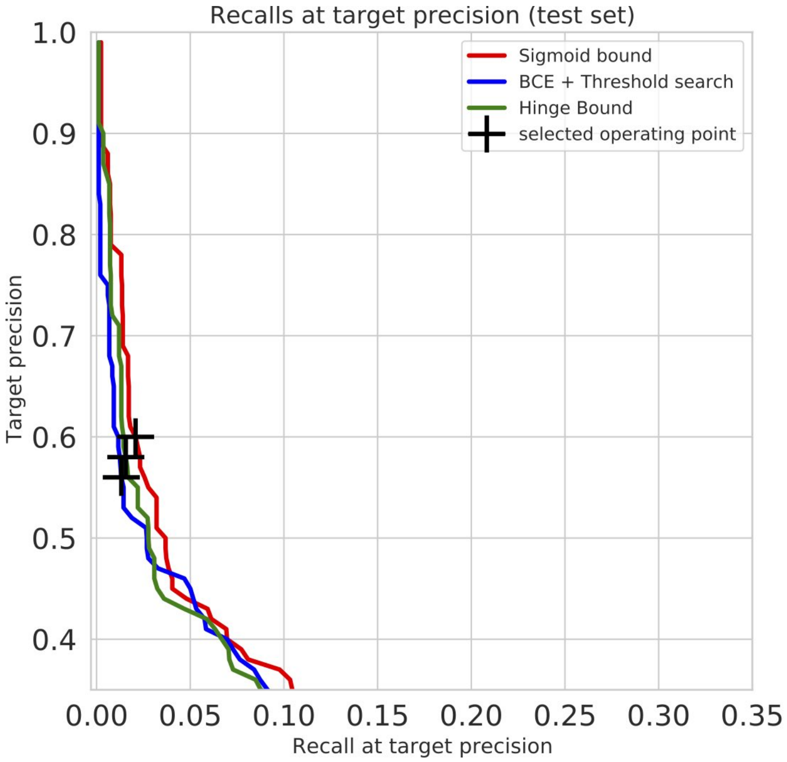

Figure B.5: Predicting in-hospital mortality on eICU in a deployable scenario (making a prediction every 12 hours), using target precisions of $\alpha = 0.7$ on train and $\alpha' = 0.6$ on valid. Our proposed sigmoid bound improves precision by 0.02 (100's of fewer false alarms across the 14518 patient-stays in the test set), and improves recall by 0.014 (100's more truly at-risk patients identified).

C.1 Per-Sequence Mortality Risk for MIMIC and eICU

We show the number of sequences in train, validation and test on both MIMIC and eICU in table C.1. We use 106 physiological measurements (2 demographics, 10 vitals and 94 labs) for MIMIC, and 17 physiological measurements (3 demographics, 8 vitals, and 6 labs) for eICU. MIMIC has a slightly higher percentage of positively labelled sequences (9.5%), compared to eICU (8.2%)

	MIMIC-III			eICU		
Split	Sequences	% positive outcomes	num. measurements	Sequences	% positive outcomes	num. measurements
Train	20682	9.54	106	43642	8.2	17
Valid	6895	9.48	106	14509	8.1	17
Test	6895	9.80	106	14518	8.2	17

Table C.1: **Dataset statistics for the per-sequence prediction scenario on MIMIC (left) and eICU (right) datasets**, showing counts of total number of examples (whole sequences) in the train, validation, and test splits, as well as the fraction of positive examples (mortality events).

C.2 Every-12-Hour Mortality Risk for MIMIC and eICU

In this more realistic scenario for mortality risk prediction, we make predictions every 12 hours throughout each patient stay in the MIMIC and eICU datasets, with the intended outcome now mortality in the next 24 hours. Table C.2 provides statistics of the number of segments we have in the dataset (recall that one stay can yield many segments, such as 0-12 hours, 0-24 hours, 0-36 hours, and so on). Again, we use 106 physiological measurements (2 demographics, 10 vitals and 94 labs) in the MIMIC dataset, and 17 measurements (3 demographics, 8 vitals, and 6 lab measurements) in the eICU dataset. The percentage of "positive" outcomes (using positive to denote the adverse event we wish to detect) drops considerably from the sequence level predictions case because for each long patient stays that end in an adverse outcome, at most 2 of many segments can be marked positive (all others will end outside the 24 hour window from the mortality event).

D EXPERIMENTAL DETAILS

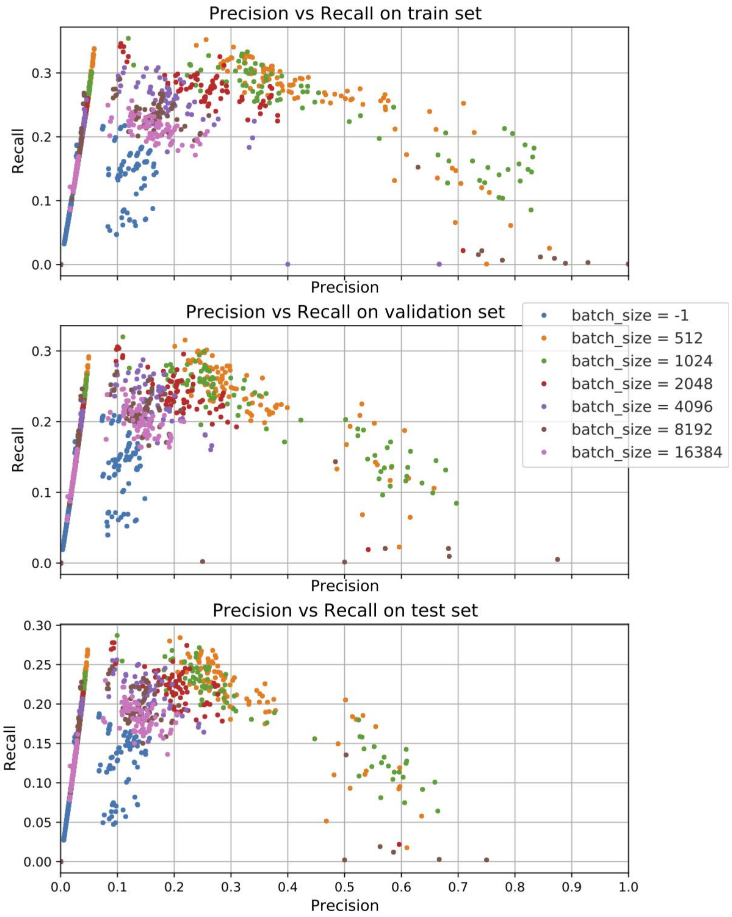

Figure B.6: Sensitivity of precision and recall as we vary the batch size from 512 up to the size of the entire training set. Rows show the training, validation, and test set performance on the MIMIC-III dataset mortality risk prediction task. Each dot represents the performance of one run of our sigmoid bound method optimized to convergence from one random initialization; we plot many runs for each batch-size setting to show the range of possible local optima. The best runs with batch sizes of 512 and 1024 surpass the best runs at any other tested batch size, as shown by the cluster of orange and green points that reach 0.45-0.75 precision and 0.18-0.26 recall on the test set, when training with the precision constraint, $\alpha = 0.7$. Batch size = -1 indicates the full training set.

D.1 Hyperparameters used

The hyperparameters used for the experiments with our proposed sigmoid bound, hinge bound, and binary cross entropy are shown in tables D.1, D.2 and D.3

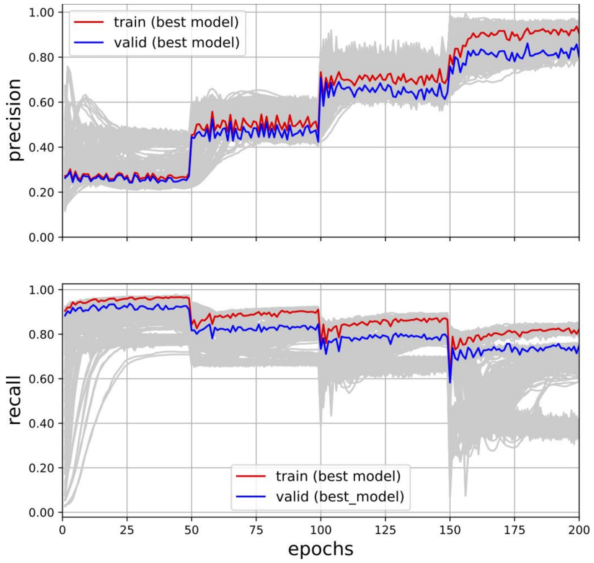

Figure B.7: Evidence of successful false-alarm-averse training of a model for MIMIC mortality risk prediction. Every 50 epochs, the desired minimum precision α is stair-stepped up, beginning at 0.225, moving at 0.225 increments, and ending at 0.9. After each step, the model's empirical precision is quickly improved to approximately satisfy the desired α via gradient-based learning. Each gray line is one run from a random initialization, the best performing run (maximizing recall while satisfying our precision goals on training and validation) is highlighted in color.

	MIMIC-III Predictions every 12 hours			eICU Predictions every 12 hours		
Split	Segments	% positive outcomes	num. measurements	Segments	% positive outcomes	num. measurements
Train	140773	2.9	106	322016	2.3	17
Valid	46451	2.9	106	106693	2.3	17
Test	46602	3.0	106	107335	2.3	17

Table C.2: **Dataset statistics for the every-12-hour prediction "deployment" scenario on MIMIC AND eICU**, showing counts of total number of examples (segments) in the train, validation, and test splits. As shown, the class imbalance is much greater than for the earlier per-sequence scenario.

Hyperparameter	**MIMIC** (*sequence level*)	**eICU** (*sequence level*)	**MIMIC** (*deployment scenario*)
weight decay	$1000, 10, 0.1, 1e^{-3}, \mathbf{1e^{-5}}, 1e^{-7}$	$1000, 10, 0.1, 1e^{-3}, 1e^{-5}, \mathbf{1e^{-7}}$	$1000, 10, 0.1, 1e^{-3}, \mathbf{1e^{-5}}, 1e^{-7}$
Lagrange multiplier λ	$\mathbf{10000}$	$\mathbf{10000}$	$100, 1000, \mathbf{10000}$
tolerance/scaling factors	$\gamma = \mathbf{7.0}, \delta = \mathbf{0.03}, \epsilon = \mathbf{0.9}$	$\gamma = \mathbf{7.0}, \delta = \mathbf{0.03}, \epsilon = \mathbf{0.9}$	$\gamma = \mathbf{7.0}, \delta = \mathbf{0.03}, \epsilon = \mathbf{0.75}$
batch size	$\mathbf{512}, 1024, 2048, 4096, 8192, 16384, -1$	$\mathbf{512}, 1024, 2048, 4096, 8192, 16384, -1$	$512, \mathbf{1024}, 2048, 4096, 8192, 16384, -1$
Optimizer	ADAM	ADAM	ADAM
learning rate	$0.001, \mathbf{0.0025}, 0.0001, 0.00025$	$0.001, \mathbf{0.0025}, 0.0001, 0.00025$	$0.001, 0.0025, 0.0001, \mathbf{0.00025}$
initialization gains	$1.0, \mathbf{1.5}, 3.0$	$1.0, \mathbf{1.5}, 3.0$	$\mathbf{1.0}, 1.5, 3.0$

Table D.1: Hyperparameters for models trained with the proposed sigmoid bounds with **50** random initializations for each hyperparameter. Optimal hyperparameters are highlighted in bold.

Hyperparameter	**MIMIC** (*sequence level*)	**eICU** (*sequence level*)	**MIMIC** (*deployment scenario*)
weight decay	$1000, 10, 0.1, \mathbf{1e^{-3}}, 1e^{-5}, 1e^{-7}$	$1000, 10, 0.1, 1e^{-3}, \mathbf{1e^{-5}}, 1e^{-7}$	$1000, 10, 0.1, 1e^{-3}, \mathbf{1e^{-5}}, 1e^{-7}$
Lagrange multiplier λ	$\mathbf{10000}$	$\mathbf{10000}$	$100, 1000, \mathbf{10000}$
tolerance/scaling factors	$\gamma = \mathbf{7.0}, \delta = \mathbf{0.03}, \epsilon = \mathbf{0.9}$	$\gamma = \mathbf{7.0}, \delta = \mathbf{0.03}, \epsilon = \mathbf{0.9}$	$\gamma = \mathbf{7.0}, \delta = \mathbf{0.03}, \epsilon = \mathbf{0.75}$
batch size	$\mathbf{512}, 1024, 2048, 4096, 8192, 16384, -1$	$\mathbf{512}, 1024, 2048, 4096, 8192, 16384, -1$	$\mathbf{512}, 1024, 2048, 4096, 8192, 16384, -1$
Optimizer	ADAM	ADAM	ADAM
learning rate	$\mathbf{0.001}, 0.0025, 0.0001, 0.00025$	$0.001, \mathbf{0.0025}, 0.0001, 0.00025$	$0.001, 0.0025, 0.0001, \mathbf{0.00025}$
initialization gains	$\mathbf{1.0}, 1.5, 3.0$	$1.0, \mathbf{1.5}, 3.0$	$1.0, \mathbf{1.5}, 3.0$

Table D.2: Hyperparameters for models trained with Eban et al. (2017)'s hinge bound with **50** random initializations for each hyperparameter. Optimal hyperparameters are highlighted in bold.

Hyperparameter	**MIMIC** (*sequence level*)	**eICU** (*sequence level*)	**MIMIC** (*deployment scenario*)
weight decay	$1000, 10, 0.1, 1e^{-3}, \mathbf{1e^{-5}}, 1e^{-7}$	$1000, 10, 0.1, \mathbf{1e^{-3}}, 1e^{-5}, 1e^{-7}$	$1000, 10, \mathbf{0.1}, 1e^{-3}, 1e^{-5}, 1e^{-7}$
batch size	$\mathbf{512}, 1024, 2048, 4096, 8192, 16384, -1$	$512, \mathbf{1024}, 2048, 4096, 8192, 16384, -1$	$512, 1024, 2048, \mathbf{4096}, 8192, 16384, -1$
Optimizer	ADAM	ADAM	ADAM
learning rate	$\mathbf{0.001}, 0.0025, 0.0001, 0.00025$	$0.001, \mathbf{0.0025}, 0.0001, 0.00025$	$0.001, 0.0025, 0.0001, \mathbf{0.00025}$

Table D.3: Hyperparameters for models trained with Binary cross entropy. Optimal hyperparameters are highlighted in bold.