Active Learning with Missing-Not-At-Random Outcomes

Alan Mishler ALAN.MISHLER@JPMORGAN.COM
Mohsen Ghassemi MOHSEN.GHASSEMI@JPMCHASE.COM
Alec Koppel ALEC.KOPPEL@JPMCHASE.COM
Sumitra Ganesh SUMITRA.GANESH@JPMORGAN.COM
J.P. Morgan AI Research

Abstract

When outcomes in training data are missing not at random (MNAR), predictors that are trained on that data can be arbitrarily biased. In some cases, however, batches of missing outcomes can be recovered at some cost, giving rise to a pool-based active learning setting. Previous active learning approaches implicitly treat all labeled data as having come from the same distribution, whereas in the MNAR setting, the training data and the initial unlabeled pool have different distributions. We propose MNAR-Aware Active Learning (MAAL), an active learning procedure that takes this into account and takes advantage of information that the missingness indicator carries about the outcome. We additionally consider acquisition functions that are attuned to the MNAR setting. Experiments on a large set of classification benchmark datasets demonstrate the benefits of our proposed approach over standard active and passive learning approaches.

Keywords: Missing not at random, pool-based, active learning

1. Introduction

Missing data is common in real-world applications, and often the missingness is systematically related to the missing values. This occurs for example in survey research, where respondents with more negative outlooks may be more likely to decline to respond (Groves et al., 2006), and in clinical trials, where an unmeasured demographic attribute may influence patients' treatment response and their likelihood of dropping out of the trial (Dziura et al., 2013).

In scenarios such as these, the target outcome (attitude towards a survey topic, treatment response) is said to be *missing not at random* (MNAR) (Rubin, 1976). MNAR datasets can induce arbitrarily large bias in downstream predictors. Sometimes, however, it is possible to recover some of the missing outcomes. For example, when a customer applies for a loan with Banks 1 and 2 and ultimately chooses Bank 2, Bank 1 does not observe the APR that the customer received. However, Bank 1 may have the opportunity to buy data from a data broker which contains this information. Given the cost associated with data collection, how should one decide which instances to uncover to maximally improve model performance?

This question may be formalized as an instance of pool-based active learning (Settles, 2009; Hino, 2020; Kumar and Gupta, 2020), in which one has access to a large pool of unlabeled data and a smaller set of labeled data and must sequentially decide how to expand

the labeled data. While a wide variety of active learning methods have been developed (e.g. Golovin et al. (2010); Houlsby et al. (2011); Balcan et al. (2006); Castro and Nowak (2008); see Kumar and Gupta (2020) for a recent review), to the best of our knowledge no previous work has explicitly considered active learning in the MNAR setting.

In this work, we propose MNAR-Aware Active Learning (MAAL). MAAL is sensitive to the fact that in the MNAR setting, the distributions of the labeled and unlabeled are different, and it takes advantage of the fact that the missingness indicator carries residual information about the outcome even once the other features have been accounted for. We formalize this fact in terms of the conditional mutual information between the missingness indicator and the outcome given the other features. Additionally, we consider acquisition functions that are attuned to the MNAR setting. Experiments show that our proposed methods yield improvements over baseline approaches drawn from the literature.

2. Problem Setup

Consider a distribution $\mathbb{P}$ over (X, D, Y), where X is a set of features, Y is an outcome or label, and $D \in \{0, 1\}$ is a missingness indicator for Y. We observe i.i.d. samples $\{(X_i, D_i, D_i Y_i\}_{i=1}^n \sim \mathbb{P}$, such that Y is only observed when $D = 1$ is otherwise set (arbitrarily) to 0. We partition the data into the labeled set $\mathcal{L} = \{(X_i, D_i, D_i Y_i) : D_i = 1\}$ and the unlabeled set $\mathcal{U} = \{(X_i, D_i, D_i Y_i) : D_i = 0\}$. We define the following key quantities:

$$\begin{aligned} m(x) &= \mathbb{E}[Y \mid X = x] \\ m_d(x) &= \mathbb{E}[Y \mid X = x, D = d], \text{ for } d \in \{0, 1\} \\ \pi(x) &= \mathbb{P}(D = 1 \mid X = x) \end{aligned} \tag{1}$$

We refer to $\pi(x)$ as the missingness *propensity*. Note that we have

$$m(x) = m_1(x)\pi(x) + m_0(x)(1 - \pi(x)) \tag{2}$$

by the law of total expectation. Our goal is to construct a model to predict Y from X, so as to minimize either mean squared error (for regression) or 0-1 error (for classification). In other words, we seek a model $\widehat{m}(x)$ that estimates $m(x)$ as accurately as possible.

We denote by $\widehat{m}_1(x)$ any model which is trained on the labeled data $\mathcal{L}$. If $D \not\perp\!\!\!\perp Y \mid X$, meaning D is not independent of Y conditional on X, then Y is said to be MNAR. In the MNAR setting, $\mathbb{P}(Y \mid X, D = 1) \neq \mathbb{P}(Y \mid X)$, so $\widehat{m}_1(x)$ may be arbitrarily biased with respect to $m(x)$. We consider how to reduce this bias in an active learning context. The expansion in (2) motivates our approach (in Section 3), which involves training separate models $\widehat{\pi}(x), \widehat{m}_0(x), \widehat{m}_1(x)$ and combining them into a model $\widehat{m}(x)$.

2.1 Pool-Based Active Learning

We suppose that the user may pay to reveal (aka "acquire") some subset of the outcomes Y_i in the unlabeled training set $\mathcal{U}$. We assume that outcomes will be acquired in batches of size B, in rounds $t = 1, \ldots T$. On each round, an *acquisition function* is used to select the next batch of outcomes to acquire, and then the current model $\widehat{m}(x)$ is updated given the new data. We consider two questions at each round: (1) Which outcomes should be

acquired? and (2) How should $\widehat{m}(x)$ be updated? The MNAR setting suggests answers to both these questions that differ from a non-MNAR setting. We consider question (2) first in Section 3 and question (1) in Section 3.1. We experimentally evaluate proposed methods developed to address these questions in Sections 4 and 5.

3. MNAR-Aware Active Learning

Standard active learning does not distinguish between data where $D = 0$ and $D = 1$: as data is acquired, it is treated as indistinguishable from previously labeled data for the purposes of model training. In the MNAR setting, however, by definition $D \not\perp\!\!\!\perp Y \mid X$, which means that D carries residual information about Y once the features X have been accounted for. Given this, our proposed approach utilizes D in the model as soon as data from the pool has been acquired, i.e. starting at $t = 1$. We refer to this as MNAR-Aware Active Learning (MAAL: Algorithm 1).

In MAAL, the entire dataset $\mathcal{D} = \mathcal{L} \cup \mathcal{U}$ is to used to train a missingness model $\widehat{\pi}(x) = \widehat{\mathbb{P}}(D = 1 \mid X = x)$, while the labeled dataset $\mathcal{L}$ is used to train an initial model $\widehat{m}(x) = \widehat{m}_1(x) = \widehat{E}[Y \mid X = x, D = 1]$. (Before active learning begins, the best we can do is train on the observed data.) After each active learning iteration, a model $f(x, d) = \widehat{E}[Y \mid X = x, D = d]$ is trained using the updated observed data; that is, we regress Y on both X and D. The individual components $f(x, 1) = \widehat{m}_1(x)$ and $f(x, 0) = \widehat{m}_0(x)$ are used to generate an updated model $\widehat{m}(x) = \widehat{\pi}(x)\widehat{m}_1(x) + (1 - \widehat{\pi}(x))\widehat{m}_0(x)$.

An alternative to this procedure is to simply update an estimate of $\widehat{m}_0(x)$ on each iteration. This would make sense if we believed that $m_1(X)$ and $m_0(X)$ were independent, so that only $\mathcal{L}$ contained information about m_1 and only $\mathcal{U}$ contained information about m_0. In general, however, we expect that $m_1(X)$ and $m_0(X)$ will be correlated, so that both estimates will improve with additional training data.

How much do we stand to gain by regressing Y on (X, D) after each round, rather than simply regressing Y on X as in standard active learning? The answer depends on the residual information that D carries about Y. This can be expressed as $I(D; Y \mid X)$, the conditional mutual information of D and Y given X. Proposition 1 formalizes the impact of adding D to the feature set on the minimum attainable MSE of the predictor.

Proposition 1 *Consider all joint measures $\mathbb{Q}(X, Y, D)$ such that $I(X; Y) = C_1$ and $I(D; Y \mid X)$ are fixed. Define L_X as the minimum attainable mean squared error of any predictor $f : \mathcal{X} \mapsto \mathcal{Y}$ and $L_{X,D}$ as the minimum attainable mean squared error of any predictor $f : \mathcal{X} \times \{0, 1\} \mapsto \mathcal{Y}$. We have*

$$\frac{L_X}{L_{X,D}} = e^{2I(D;Y|X)}. \tag{3}$$

In particular, when $Y = m(X, D) + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$ is Gaussian noise with some variance $\sigma^2 < \infty$, we have

$$\frac{\mathbb{E}_{\mathbb{P}}[(Y - m(X))^2]}{\mathbb{E}_{\mathbb{P}}[(Y - m(X, D))^2]} \geq e^{2I(D;Y|X)}. \tag{4}$$

See Appendix A for the proof. The first part of Proposition 1 says that the ratio of the minimum attainable MSEs in predictors that use X vs. predictors that use (X, D) as features

Algorithm 1 MNAR-Aware Active Learning (MAAL)

Data: Initial dataset $\mathcal{D} = \mathcal{L} \cup \mathcal{U}$
Input: Acquisition function $\alpha : \mathcal{X} \times \mathcal{L} \mapsto \mathbb{R}$
Train models $\hat{m}_1 : \mathcal{X} \mapsto \mathbb{R}$ using $\mathcal{L}$ and $\hat{\pi} : \mathcal{X} \mapsto [0, 1]$ using $\mathcal{D}$.
$\hat{m}(x) \leftarrow \hat{m}_1(x)$
For batch t in $1, \ldots, T$:
  Acquire new batch of labeled data $\mathcal{L}^t$ using acquisition function
  $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{L}^t$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{L}^t$
  Train new model $f(x, d) : \mathcal{X} \times \{0, 1\} \mapsto \mathbb{R}$ using $\mathcal{L}$
  $\hat{m}(x) \leftarrow \hat{\pi}(x) f(x, 1) + (1 - \hat{\pi}(x)) f(x, 0)$
Return: $\hat{m}(x)$

is exponential in the conditional mutual information between D and Y given X. The more residual information about Y that D contains, the greater the potential reduction in error that results from including D in the predictor. The second part of Proposition 1 identifies a data generating process under which a reduction at least this large is actually achievable[1].

Error estimates and hyperparameter tuning In practice, $f(x, d)$ in Algorithm 1 is likely to be the output of a flexible learner such a neural network or gradient boosting algorithm. In this case, hyperparameter tuning may be employed at each update. The default approach to this involves cross-validation with error estimates $M^{-1} \sum_{i=1}^{M} \ell(f(X_i), Y_i)$ for folds of size M using the observed data. However, this is best understood as giving an estimate of $\mathbb{E}[\ell(f(X), Y) \mid D = 1]$, whereas the true error is

$$\begin{aligned} \mathbb{E}[\ell(f(X), Y)] =& \mathbb{E}[\ell(f(X), Y) \mid D = 1]\mathbb{P}(D = 1) + \\ & \mathbb{E}[\ell(f(X), Y) \mid D = 0]\mathbb{P}(D = 0) \end{aligned} \tag{5}$$

A better error estimator for purposes of tracking and optimizing model performance should take this into account, for example by estimating the error conditional on $D = 0$ and $D = 1$ and then taking their sum weighted by the prevalence of $D = 1$ vs. $D = 0$. These details are left implicit in the training of $f(x, d)$ in Algorithm 1.

3.1 MNAR-Aware Acquisition Functions

While MAAL is agnostic to the choice of acquisition function, we consider what kind of acquisition function would be best suited to the MNAR setting. Most active learning approaches use predictions from the current model as components of the acquisition function. For example, in a classification setting, the max-entropy approach assigns weights proportional to the entropy of the predicted class memberships (Campbell et al., 2000). Since the proposed model update on each iteration in Algorithm 1 is distinct from the usual active learning approach, it will yield different acquisition function values than a model trained only on X.

1. See Bertsimas et al. (2021), Proposition 1, for a result in the same vein.

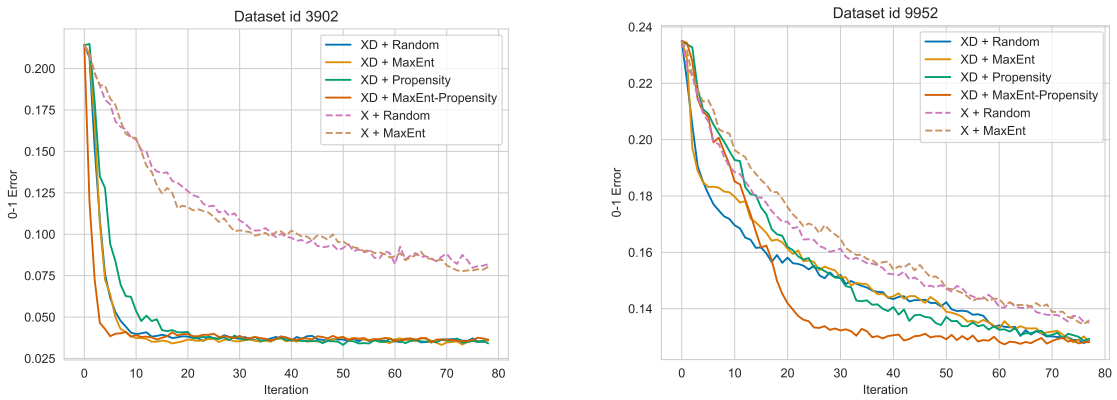

Figure 1: Active learning results for six approaches ([method] + [acquisition function]) on two datasets. XD refers to the MNAR-aware approach in Algorithm 1; X refers to the baseline approach which just updates a predictor $\widehat{m}_1(x)$ on each iteration. The four acquisition functions are described in Section 3. MNAR-aware learning outperforms the baseline approach in general in terms of both speed of learning (how quickly the error decreases) and the final error achieved (once all the data has been acquired).

In addition to this, however, we consider utilizing the propensity weight $(1 - \widehat{\pi}(x))$, either as an acquisition function by itself or as a multiplier applied to any other acquisition function. The motivation is that when $\pi(x)$ is large, the primary contribution to $m(x)$ comes from $m_1(x)$, whereas when $\pi(x)$ is small, the primary contribution comes from $m_0(x)$, so this favors the acquisition of data points precisely where we most care about estimating $m_0(x)$ well.

The use of $(1 - \widehat{\pi}(x))$ as an acquisition function is similar to the *counterfactual propensity acquisition* proposed by Jesson et al. (2021). Although that method was empirically suboptimal relative to other methods proposed methods in the same paper, the setting was different from ours: the target was a causal effect rather than an optimal predictor, and the authors assumed *unconfoundedness*, which is equivalent to assuming that the outcomes are not MNAR in the context of causal inference.

4. Active Learning Experiments

We compare our MNAR-aware method in Algorithm 1 to a *baseline* method which simply updates a predictor $\widehat{m}_1(x)$ by regressing Y on X using the labeled data on each iteration. We consider a binary classification setting. We compare four acquisition functions: (1) *Random*, which selects batches of size B uniformly at random (also known as *passive learning*); (2) *MaxEnt* (maximum entropy), which identifies the B points where the predicted class membership probabilities have the highest entropy (Campbell et al., 2000); (3) *Propensity*, which ranks the values in $\mathcal{U}$ by $(1 - \widehat{\pi}(X))$ and selects the top B remaining points at each iteration; and (4) *MaxEnt-Propensity*, which multiplies the entropy values by $(1 - \widehat{\pi}(x))$ and selects the B points with the highest values. We utilize the random and MaxEnt acquisition functions with the baseline method, and we utilize all four acquisition functions with the

MNAR-aware method. As shorthand, we use XD to refer to the MNAR-aware method and X for the baseline method, reflecting the features utilized in each.

We use 34 datasets from the public OpenML-CC18 benchmark, a set of real-world classification tasks designed for machine learning benchmarking (Bischl et al., 2021). Full details of the datasets and procedure are given in Appendix B.

5. Results

We compute the area under the curve (AUC) and 0-1 error for each method. Results are averaged over the 10 runs. In case the runs were different lengths, due to randomness in the size of the labeled vs. unlabeled datasets $\mathcal{L}$ and $\mathcal{U}$, we first truncate the runs to the shortest length among the 10. Figure 1 shows results for two of the datasets. The figures for the remaining datasets are in Appendix C. Table 1 in Appendix C contains the areas under the curve (AUCs) and 0-1 errors.

In 31 out of 34 datasets, the smallest AUC comes from the MNAR-aware method. The smallest AUC most frequently comes from the MNAR-aware method coupled with the *MaxEnt-Propensity* acquisition function. In 32 out of 34 datasets, the MNAR-aware method achieves the smallest 0-1 error among the methods, though in three of these cases there is a tie with the baseline method. Only in two datasets does the baseline method achieve a smaller final error than the MNAR-aware method. The differences in performance between the acquisition functions in terms of error is largely an artifact of the fact that in some of the larger datasets, the pool $\mathcal{U}$ was not exhausted by the end of the active learning procedure. When the pool is exhausted, every method concludes with access to the same training data, so the distinctions between acquisition functions would disappear. As expressed by Proposition 1, however, the distinction between the MNAR-aware method and the baseline method does not disappear.

Note additionally that the severity of the MNAR depends not just on the relationship between Y and D but on the relationship between Y and X. If Y were a deterministic function of X, for example, then it would not be MNAR regardless of the distribution of (D, Y). This dynamic is reflected in Figure 1; in (a), the difference in final performance between the two baseline approaches and the four MNAR-aware approaches is larger than in (b). This is presumably because the conditional mutual information $I(D; Y \mid X)$ is higher in (a) than in (b), which means there is more residual information for the MNAR-aware method to capture.

6. Conclusion

We studied the problem of pool-based active learning for training predictors in a setting where outcomes in the initial training data are missing not at random (MNAR). We proposed MNAR-Aware Active Learning (MAAL) (Algorithm 1), which is sensitive to the fact that the training data and the unlabeled pool come from different distributions and which utilizes the residual correlation between the missingness indicator and outcome to improve the predictor. We also considered how (an estimate of) the missingess propensity may be used as a multiplier to modify existing acquisition functions or as an acquisition function in its own right. Our proposed methods improve over existing baselines on a wide range of datasets.

Appendix A. Proof of Proposition 1

Proof Define $\mathcal{Q}_{I_1,I_2} = \{\mathbb{Q}(X,Y,D) : I(X;Y) = I_1,\ I(D;Y \mid X) = I_2\}$ and $L_X^{I_1,I_2} = \sup\{L : \min_{\substack{f:\mathcal{X}\mapsto\mathbb{R} \\ \mathbb{Q}(X,Y,D)}} \mathbb{E}_{\mathbb{Q}}[(Y - f(X))^2] \geq L,\ \forall \mathbb{Q} \in \mathcal{Q}^{I_1,I_2},\}$ and $L_{X,D}^{I_1,I_2} = \sup\{L : \min_{\substack{f:\mathcal{X}\mapsto\mathbb{R} \\ \mathbb{Q}(X,Y,D)}} \mathbb{E}_{\mathbb{Q}}[(Y - f(X,D))^2] \geq L,\ \forall \mathbb{Q} \in \mathcal{Q}^{I_1,I_2},\}$. We drop superscripts I_1, I_2 for ease of notation. We have

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y \mid X) \\ &= H(Y) - \mathbb{E}_{\mathbb{P}} \log \mathbb{P}(Y \mid X) \\ &= H(Y) - \mathbb{E}_{\mathbb{P}} \log \mathbb{Q}(Y \mid X) + D_{KL}(\mathbb{P} \parallel \mathbb{Q}) \\ &\geq H(Y) - \mathbb{E}_{\mathbb{P}} \log \mathbb{Q}(Y \mid X) \end{aligned} \tag{6}$$

where $\mathbb{Q}$ is any joint measure defined on the same domain as $\mathbb{P}$ and $D_{KL}(\cdot \parallel \cdot)$ is the KL-divergence. Let $\mathbb{Q}(Y \mid X = x) = \mathcal{N}(m(X = x),\ \mathbb{E}_{\mathbb{P}}[(Y - m(X))^2 \mid X = x])$. We have

$$I(X;Y) \geq H(Y) - \frac{1}{2} \log \mathbb{E}_{\mathbb{P}}[(Y - m(X))^2] - \frac{1}{2} \log 2\pi e. \tag{7}$$

Therefore,

$$\log L_X = 2H(Y) - 2I(X;Y) - \log 2\pi e.$$

Similarly,

$$\log L_{X,D} = 2H(Y) - 2I\big((X,D);Y\big) - \log 2\pi e.$$

Therefore,

$$\begin{aligned} \log \frac{L_X}{L_{X,D}} &= 2I\big((X,D);Y\big) - 2I(X;Y) \\ &= 2I(D;Y \mid X) \end{aligned}$$

Simple algebraic operations give equality (3). Inequality (4) follows from the fact that (7) holds with equality when $Y|X$ is a Gaussian distribution. ■

Appendix B. Experiment Details

Datasets and preprocessing We utilized datasets from the public OpenML-CC18 benchmark, a set of real-world classification tasks designed for machine learning benchmarking (Bischl et al., 2021). We restricted to binary classification tasks, yielding 35 datasets. (Our results include 34 datasets: dataset 3918 was excluded due to runtime errors.) We preprocessed them in the same manner described in Bahri et al. (2022). In order to induce MNAR, in each dataset we set $\mathbb{P}(D = 1 \mid Y = 0) = 0.4$ and $\mathbb{P}(D = 1 \mid Y = 1) = 0.7$ and then randomly sampled the missingness indicator D according to these probabilities. We used a train/test split of 80%/20% and split the train data into the labeled and unlabeled sets $\mathcal{L}$ and $\mathcal{U}$. We trained a *baseline* xgboost classifier $\widehat{m}_1(x)$ on $\mathcal{L}$ and an xgboost classifier $\widehat{\pi}(x)$ on the full train data.

In order to ensure a sufficient "runway" for active learning, we restricted each training set to a random sample of size 1000 and reserved the remaining data for the test set. For initial datasets smaller than 1000, we instead did a 70%/30% train/test split. To induce MNAR, for each training dataset, we set the missingness probability when $Y = 0$ to $\mathbb{P}(D = 0 \mid Y = 0) = 0.9$ and then computed the missingness probability $\mathbb{P}(D = 0 \mid Y = 1)$ required to achieve a marginal missingness probability of $\mathbb{P}(D = 0) = 0.8$ in the training data. (When this wasn't possible, we instead set $\mathbb{P}(D = 0 \mid Y = 1) = 0.9$ and then computed $\mathbb{P}(D = 0 \mid Y = 0)$ accordingly.) On each active learning iteration, we sampled D at random according to the specified probabilities. This resulted in initial labeled datasets $\mathcal{L}$ of size at most roughly 200 and active learning pools $\mathcal{U}$ of size at most roughly 800.

Active learning procedure We utilized xgboost classifiers for $\widehat{\pi}(x)$, $\widehat{m}_1(x)$, and the MNAR-aware learner $f(x, d)$. We set the batch size to $B = 10$ and drew samples from the pool $\mathcal{U}$ until it is exhausted or until 1000 batches have been drawn. Each active learning procedure was repeated 10 times.

Appendix C. Experiment Results

	AUC						Error					
	XD				X		XD				X	
Dataset	Rand	Max	Prop	MP	Rand	Max	Rand	Max	Prop	MP	Rand	Max
3	1	1.015	1.367	1.203	**0.967**	1.004	0.014	**0.013**	**0.013**	**0.013**	0.014	**0.013**
15	1	**0.899**	1.044	**0.899**	1.011	0.906	0.042	**0.036**	0.042	**0.036**	0.041	**0.036**
29	**1**	1.006	1.049	1.066	1.099	1.120	**0.139**	0.140	0.145	0.146	0.154	0.157
31	1	1.046	1.017	**0.964**	1.140	1.190	0.229	0.230	0.229	**0.220**	0.249	0.252
37	1	1.095	0.976	**0.955**	1.180	1.203	0.234	0.232	0.237	**0.223**	0.255	0.245
43	**1**	1.004	1.151	1.066	1.047	1.058	**0.059**	**0.059**	**0.059**	0.060	0.061	0.061
49	1	1.103	1.701	1.671	**0.945**	1.020	0.018	0.019	0.021	0.017	0.016	**0.014**
219	1	**0.995**	1.108	1.085	1.031	1.021	0.200	0.200	0.199	**0.198**	0.206	0.204
3021	1	1.016	1.325	**0.872**	1.002	1.017	**0.020**	**0.020**	**0.020**	**0.020**	0.021	**0.020**
3902	1	0.979	1.081	**0.919**	2.550	2.514	0.036	0.036	**0.035**	0.037	0.081	0.079
3903	1	1.003	0.993	**0.946**	1.003	0.993	0.100	0.100	0.099	**0.098**	0.101	0.099
3904	1	0.980	1.053	**0.975**	1.493	1.490	**0.173**	**0.173**	**0.173**	**0.173**	0.214	0.215
3913	1	0.996	1.133	**0.967**	1.395	1.404	**0.121**	0.124	0.125	0.122	0.170	0.166
3917	1	1.010	1.094	**0.939**	1.741	1.749	**0.103**	**0.103**	0.105	0.104	0.153	0.151
7592	1	**0.987**	1.173	1.140	1.061	1.054	**0.154**	**0.154**	0.162	**0.154**	0.165	0.164
9910	1	**0.983**	1.084	1.071	1.047	1.038	0.233	0.232	0.233	**0.231**	0.244	0.242
9946	1	0.958	0.946	**0.881**	1.007	0.967	0.052	0.053	**0.050**	0.056	0.053	0.055
9952	1	1.013	1.024	**0.968**	1.072	1.094	**0.128**	0.129	0.129	**0.128**	0.136	0.135
9957	1	**0.978**	1.077	1.007	1.089	1.049	0.134	**0.130**	0.135	0.133	0.141	0.139
9971	1	0.999	1.033	**0.926**	1.352	1.341	0.228	0.228	0.230	**0.227**	0.304	0.297
9976	**1**	1.008	1.018	1.021	1.036	1.055	0.287	**0.279**	0.283	0.282	0.295	0.295
9977	**1**	1.067	1.475	1.446	1.108	1.162	**0.051**	0.052	0.052	0.052	0.056	0.057
9978	1	0.989	1.013	**0.987**	1.001	0.990	**0.056**	0.057	0.057	0.057	**0.056**	0.057
10093	1	1.001	1.365	1.058	1.005	**0.986**	0.011	0.012	**0.008**	0.009	0.011	0.012
10101	1	1.004	1.061	**0.965**	1.318	1.282	0.204	0.203	0.204	**0.201**	0.227	0.231
14952	**1**	1.019	1.120	1.091	1.015	1.034	**0.063**	0.065	**0.063**	**0.063**	0.064	0.066
14954	**1**	1.018	1.046	1.062	1.066	1.080	0.198	**0.194**	0.197	0.197	0.223	0.220
14965	1	1.019	0.968	**0.959**	1.638	1.634	**0.077**	**0.077**	**0.077**	**0.077**	0.114	0.114
125920	1	1.031	0.984	**0.960**	1.149	1.178	**0.406**	0.415	0.415	0.419	0.479	0.485
146819	1	**0.932**	0.984	0.998	1.001	0.934	0.044	0.044	**0.042**	0.043	0.044	0.043
146820	1	0.988	1.155	**0.879**	1.005	0.996	0.020	0.020	0.020	**0.019**	0.021	0.020
167120	**1**	1.003	1.001	1.002	1.008	1.009	**0.491**	**0.491**	**0.491**	**0.491**	0.495	0.496
167125	1	**0.964**	1.665	1.621	1.155	1.096	**0.026**	**0.026**	0.027	**0.026**	0.030	0.031
167141	1	1.010	1.083	**0.968**	1.342	1.323	0.053	0.054	0.053	0.053	0.053	**0.052**

Table 1: AUCs and and 0-1 error rates for MNAR-aware active learning (XD) vs. the baseline approach (X), with 4 acquisition functions: random (Rand), MaxEnt (Max), propensity (Prop), and MaxEnt-Propensity (MP). AUCs are normalized by the value for XD + Rand. All values are averaged over 10 runs. Errors are then calculated as the average of the errors after the final three active learning batches, to reduce variance. Bold values indicate the smallest AUC or error across the 6 methods. In 31 out of 34 datasets, MNAR-aware active learning results in the smallest AUC, with the smallest value usually achieved by MNAR-aware active learning + the MaxEnt-Propensity acquisition function. MNAR-aware active learning almost always results in the smallest final error, in accordance with Proposition 1.

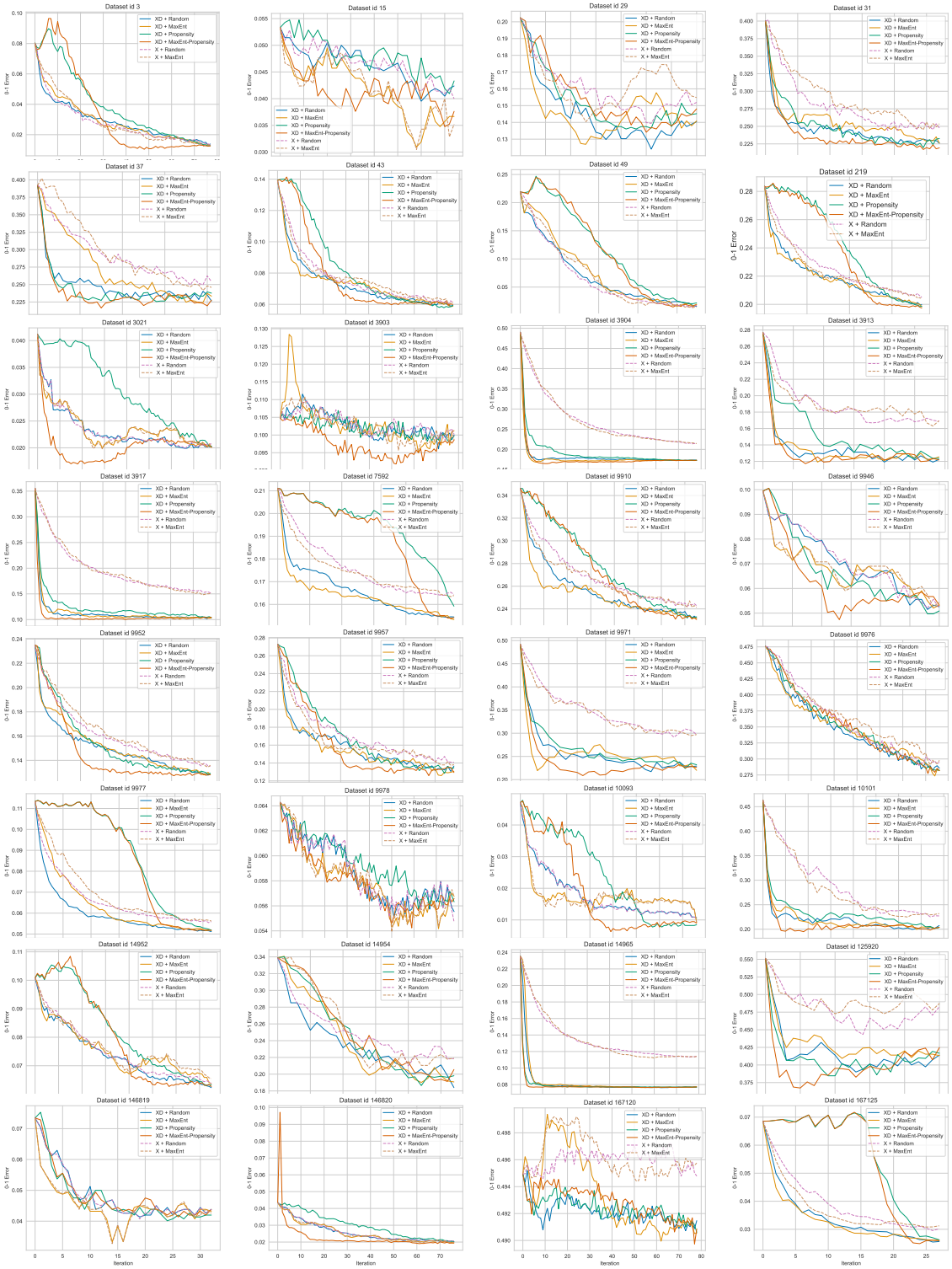

Figure 2: Full active learning results, excluding the results shown in Figure 1.

References

Dara Bahri, Heinrich Jiang, Tal Schuster, and Afshin Rostamizadeh. Is margin all you need? An extensive empirical study of active learning on tabular data, October 2022. URL `http://arxiv.org/abs/2210.03822`.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 65–72, 2006.

Dimitris Bertsimas, Arthur Delarue, and Jean Pauphilet. Prediction with Missing Data. *arXiv:2104.03158 [cs, stat]*, April 2021. URL `http://arxiv.org/abs/2104.03158`.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael Gomes Mantovani, Jan N. van Rijn, and Joaquin Vanschoren. OpenML benchmarking suites. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Colin Campbell, Nello Cristianini, and Alex J. Smola. Query learning with large margin classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 111–118, San Francisco, CA, USA, 2000.

Rui M Castro and Robert D Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353, 2008.

James D Dziura, Lori A Post, Qing Zhao, Zhixuan Fu, and Peter Peduzzi. Strategies for dealing with Missing data in clinical trials: From design to Analysis. *Yale Journal of Biology and Medicine*, 86:343–358, 2013.

Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal bayesian active learning with noisy observations. *Advances in Neural Information Processing Systems*, 23, 2010.

Robert M. Groves, Mick P. Couper, Stanley Presser, Eleanor Singer, Roger Tourangeau, Giorgina Piani Acosta, and Lindsay Nelson. Experiments in Producing Nonresponse Bias. *Public Opinion Quarterly*, 70(5):720–736, 2006.

Hideitsu Hino. Active learning: Problem settings and recent developments. *arXiv preprint arXiv:2012.04225*, 2020.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011.

Andrew Jesson, Panagiotis Tigas, Joost van Amersfoort, Andreas Kirsch, Uri Shalit, and Yarin Gal. Causal-bald: Deep bayesian active learning of outcomes to infer treatment-effects from observational data. *Advances in Neural Information Processing Systems*, 34: 30465–30478, 2021.

Punit Kumar and Atul Gupta. Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey. *Journal of Computer Science and Technology*, 35(4):913–945, July 2020.

Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.