Model-based metrics: Sample-efficient estimates of predictive model subpopulation performance

Andrew C. Miller ACMILLER@APPLE.COM
Apple

Leon A. Gatys LGATYS@APPLE.COM
Apple

Joseph Futoma JFUTOMA@APPLE.COM
Apple

Emily Fox EMILY_FOX@APPLE.COM
Apple

Abstract

Machine learning models — now commonly developed to screen, diagnose, or predict health conditions — are evaluated with a variety of performance metrics. An important first step in assessing the practical utility of a model is to evaluate its average performance over a population of interest. In many settings, it is also critical that the model makes good predictions within predefined subpopulations. For instance, showing that a model is fair or equitable requires evaluating the model's performance in different demographic subgroups. However, subpopulation performance metrics are typically computed using only data from that subgroup, resulting in higher variance estimates for smaller groups. We devise a procedure to measure subpopulation performance that can be more sample-efficient than the typical estimator. We propose using an *evaluation model* — a model that describes the conditional distribution of the predictive model score — to form *model-based metric* (MBM) estimates. Our procedure incorporates model checking and validation, and we propose a computationally efficient approximation of the traditional nonparametric bootstrap to form confidence intervals. We evaluate MBMs on two tasks: a semi-synthetic setting where ground truth metrics are available and a real-world hospital readmission prediction task. We find that MBMs consistently produce more accurate and lower variance estimates of model performance, particularly for small subpopulations.

1. Introduction

Machine learning (ML) is increasingly used to screen, diagnose, and predict health conditions. As of March 2020, at least 222 medical devices using ML were approved by the US Food and Drug administration, spanning medical specialties including radiology, cardiology, and ophthalmology (Muehlematter et al., 2021). Ensuring that a model is suitable for deployment in a real-world setting requires quantifying its performance using evaluation metrics such as the area under the receiver operating characteristic curve (AUC), positive predictive value (PPV), and false positive rate (FPR); robustly estimating model perfor-

mance is of particular importance in medical contexts. It is also important to assess a model's performance *within* subpopulations of interest, which is a necessary step in diagnosing issues with the fairness or equity of model predictions, and to identify avenues for model improvement.

The typical way to estimate the performance of a model on subgroups is to compute the relevant set of metrics on the subpopulation *subsample*. However, subsample-based estimates of model performance carry more uncertainty in smaller subpopulations. This becomes especially problematic when we examine groups defined by the intersection of multiple demographic categories (e.g., sex *and* age). And in clinical studies it can be challenging to gather more data from such underrepresented populations.

Consider a hypothetical ML model to predict future cardiac events with the goal of screening individuals into "low" and "elevated" risk categories defined by the Framingham cardiovascular disease risk score (D'agostino et al., 2008). To scrutinize the accuracy and fairness of this model, we measure its performance on demographic subpopulations and form estimates of metrics such as AUC, PPV, and FPR within each group. Suppose that our validation cohort consists of only a few thousand individuals, a modest sample size that may occur in many different medical applications. If our data is representative of the entire United States, we would only expect about 1.3% of individuals to be non-Hispanic African American males between 39 and 52 years old. In our observed sample, this would likely correspond to a few dozen individuals at most. At such a small sample size, our estimates of various model metrics for this group will be extremely imprecise, impeding our ability to improve the model's performance within this population.

The crux of the problem is simple: *restricting estimates to subpopulation subsamples does not use all the information available*. Individuals with similar, but not exactly the same, demographic labels may contain information that can improve subpopulation estimates. For instance, in our cardiac risk application, it may be that non-Hispanic Caucasian males between 39 and 52, or non-Hispanic African American males between 26 and 39, contain helpful information about the non-Hispanic African American males between 39 to 52 subgroup. This is exactly the benefit of multi-level modeling: share information between groups in a principled manner to improve the estimate within each group (Gelman et al., 2013).

Following this idea, we propose the use of a meta-analytic *evaluation model* to estimate the subpopulation performance of a *predictive model* (e.g., the hypothetical cardiac risk model). The evaluation model approximates the distribution of the predictive model score conditioned on subpopulation and other covariate information, and is used to form what we will call a *model-based metric* (MBM) estimate. The evaluation model is fit using all of the data, and by design can incorporate information from the entire sample into each subpopulation estimate. Figure 1 illustrates this partial pooling idea and highlights subpopulation size imbalances in the US.

We propose a procedure to fit, validate, and ultimately use evaluation models to estimate common performance metrics within subpopulations. To validate the evaluation models, we use stratified cross validation to compare out-of-sample log-likelihoods between various evaluation models and to a simple nonparametric kernel density estimate as a baseline. To obtain confidence intervals for the downstream MBM estimates of model performance within subpopulations, we use bootstrap samples. Furthermore, we propose a novel approximate

bootstrap procedure based on importance weighting to avoid fitting a new, computationally expensive evaluation model for each bootstrap replicate (e.g. for Bayesian multilevel models fit via Markov chain Monte Carlo).

In Section 5 we study the empirical performance of our estimators. In many scenarios of interest where only a small dataset has been collected, ground truth metrics are unavailable, complicating the validation of the proposed estimators. As such, we construct two experimental scenarios: (i) a semi-synthetic prediction task that uses demographic and biomarker statistics that match the US population for a cardiovascular risk prediction application, and (ii) a hospital readmission prediction task using data from a large, multi-center study including various clinical attributes (Strack et al., 2014). In each setting, we either simulate or use the large dataset to compute ground truth performance metrics in each subpopulation. We then compare these ground truth values to estimates derived from smaller subsamples of the full data, mimicking the more typical restricted data setting. We compare the accuracy and coverage statistics for subpopulation subsample estimates with the proposed MBM estimates. Our findings demonstrate that our approach can be much more accurate in smaller subpopulations than the typical subsample estimator.

Generalizable Insights about Machine Learning in the Context of Healthcare

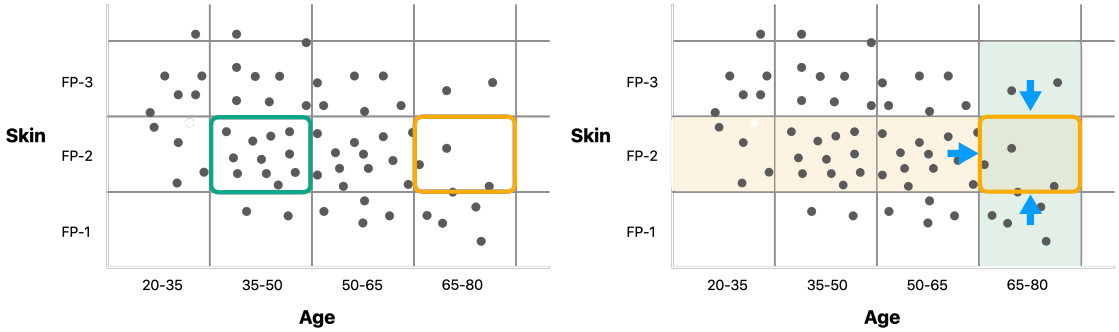
We show that sophisticated statistical techniques can improve the evaluation of ML model performance in smaller subpopulations. Measuring model performance is a crucial part of developing machine learning models to be integrated into clinical settings. In health applications, it is especially important to understand the pattern of successes and failures of a model on a diverse set of subpopulations. Increasing the precision with which we estimate a model's performance on small subgroups can help us build models that are more accurate and fair — they can highlight subpopulations where we may want to improve the model, or collect additional data through targeted recruitment efforts. Through a thorough empirical study, we examine the benefits and shortcomings of our approach, and discuss how we might cope with problems that evade the evaluation model. For instance, we highlight when it may be best to revert back to the traditional subsample estimators.

In many health applications we are data-bound, constrained by whatever retrospective data has already been collected. The aim of this work is to squeeze as much information as possible out of the data at hand. More broadly, we adopt the view that predictive model evaluation is itself a data analysis problem, and we should use all of the statistical tools available to help us understand our machine learning models and their potential effects and limitations.

2. Background

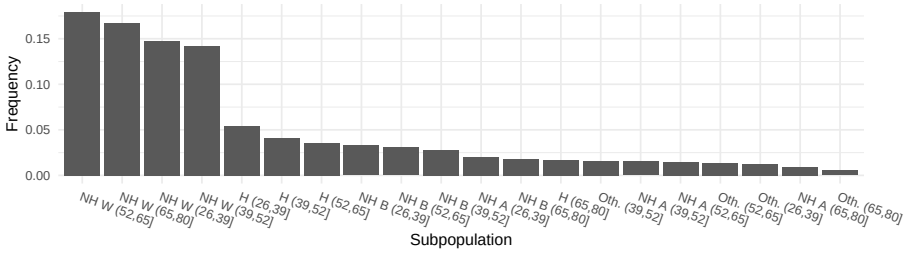
Consider the setting where a statistical machine learning model (i.e., the *prediction model*) outputs a continuous-valued score to predict a binary class label.[1] For binary classification, the predictive model score is simply the log conditional probability of the positive class,

1. The extension to multiple discrete categories is straightforward, and we leave the extension to real-valued outcomes to future work.



(a) Subpopulation samples

(b) Sharing information with model-based estimates



(c) Subpopulation frequencies — NHANES

Figure 1: Method overview: (a) For subpopulations determined by age and Fitzpatrick skin type (FP) bins, we evaluate model performance *within* each subpopulation. Some subpopulations are more populous (green) than others (gold). Less data leads to noisier metric estimates. (b) We propose model-based metric estimators, which relies on a joint estimate of the class conditionals informed by all observations. The model incorporates more information into metric estimates for small subpopulations. For example, the subpopulation outlined in gold will be informed by other observations with the same skin type (FP-2) and the same age bin (65-80). This leads to more sample-efficient and informative estimates of model performance. (c) Subpopulation frequencies, defined by race and age categories, in the National Health and Nutrition Examination Survey, designed to be representative of the United States population. Some subpopulations are over $10\times$ larger than others.

$\ln Pr(Y = 1 \,|\, \texttt{input})$, where the data `input` is whatever input data the predictive model conditions on (e.g., an image, laboratory values, vital signs). Additionally, we observe subpopulation information and additional relevant covariates. We denote these variables:

- $Y \in \mathcal{Y} = \{0, 1\}$: the binary outcome of interest, e.g., low or elevated disease risk;
- $S \in \mathbb{R}$: the predictive model score, e.g., from a machine learning model. In a binary classification setting, we define $S \triangleq \ln Pr(Y = 1 \,|\, \texttt{input})$;
- $A \in \mathcal{A}$: the discrete subpopulations defined by demographic attributes, e.g., age, sex, and race categories;
- $X \in \mathcal{X}$: additional covariates to condition on, e.g., BMI or cholesterol.

Our goal is to measure the prediction model's performance on each subpopulation as precisely as possible. Given a predictive model with score $S = s$ and true class $Y = y$, we evaluate the model using a set of performance metrics.[2]

As a concrete example, consider the *area under the receiver operating characteristic curve* (AUC) — a ubiquitous performance metric — conditioned on a particular subpopulation a. The AUC is equivalent to the probability of correctly ranking two independent observations, i and j, one from each class:

$$AUC(a) = Pr(S_j > S_i \,|\, A_i = A_j = a, Y_i = 0, Y_j = 1) \tag{1}$$

$$= \mathbb{E}_{Pr(S \,|\, Y=1, A=a) Pr(S' \,|\, Y=0, A=a)}[\mathbf{1}(S > S')] \,. \tag{2}$$

A high AUC indicates that the model prediction is more likely to correctly rank two randomly selected individuals with different outcome (Y) values, while an AUC of 0.5 indicates that a model performs this ranking no better than chance. Crucially, note that the AUC is a function of the *class conditional distributions* of the prediction model score, $Pr(S \,|\, Y = 1, A = a)$ and $Pr(S' \,|\, Y = 0, A = a)$.

Another common metric is the *positive predicted value* (PPV) at threshold τ (i.e., when $S > \tau$, the model predicts a positive label $Y = 1$). Again, this metric can be expressed as a function of the class conditional distributions of the prediction model score,

$$PPV_\tau(a) = \frac{Pr(S > \tau, Y = 1 \,|\, A = a)}{Pr(S > \tau, Y = 1 \,|\, A = a) + Pr(S > \tau, Y = 0 \,|\, A = a)}, \tag{3}$$

which is a function of both the class conditional distributions of the score S and the overall prevalence of positive examples within subpopulation a. PPV offers a different view into model performance, as it is the probability that a subject is actually in the positive class $Y = 1$ given a positive prediction $S > \tau$.

Lastly, the *false positive rate* (FPR) describes the frequency of negative examples incorrectly classified to be positive, and can be written

$$FPR_\tau(a) = Pr(S > \tau \,|\, Y = 0, A = a) = \mathbb{E}_{Pr(S \,|\, Y=0, A=a)} \left[\mathbf{1}(S > \tau)\right] \,, \tag{4}$$

which is a function of only the $Y = 0$ class conditional distribution.

In general, we denote the subpopulation performance metric we wish to estimate as $\theta(a)$, a functional of the class conditional distributions,

$$\theta(a) = f\Big(\big\{Pr(S \,|\, Y = y, A = a)\big\}_{y \in \{0,1\}}\Big) \,, \tag{5}$$

where the form of $f(\cdot)$ specifies the desired performance metric.

2.1. Non-parametric AUC, PPV, and FPR estimators

Typically we only have access to a sample from these conditional distributions, necessitating sample-based estimators of these metrics. Denote the observed data $\mathcal{D} = \{(a_n, x_n, y_n, s_n)\}_{n=1}^N$, corresponding to, respectively, demographic attributes, covariates, the true outcome class, and the ML model score for subject n for each of N total subjects. Let $N_y = \{n : y_n = y\}$.

2. We use lower case letters to denote realizations of these random variables (e.g., $A = a$).

AUC estimators: The normalized Mann-Whitney U-Statistic. A common estimator for the AUC is the unbiased Mann-Whitney U-statistic. Given a sample of scores from positive ($y_n = 1$) examples $\{s_n : n \in N_1\}$ and negative ($y_n = 0$) examples $\{s_n : n \in N_0\}$, the U-statistic estimator of the AUC is

$$\hat{\theta} = \frac{1}{|N_1||N_0|} \sum_{n_1 \in N_1} \sum_{n_0 \in N_0} \mathbf{1}(s_{n_1} > s_{n_0}) \,. \tag{6}$$

To estimate frequentist confidence intervals, a common strategy is to compute statistics of bootstrap samples (Efron, 1992). Another common way to compute the AUC metric is by numerically integrating the empirical ROC curve directly. This can be accomplished in quasi-linear time and may be more suitable for large samples.

Threshold estimators: FPR, PPV To compute metrics with respect to a fixed threshold τ, estimators are typically simple functions of the confusion matrix. For example, the false positive rate is typically estimated by the empirical frequency of negative examples incorrectly classified as positive,

$$FPR_\tau = \frac{1}{N} \sum_{n_0 \in N_0} \mathbf{1}(s_{n_0} > \tau) \,. \tag{7}$$

Likewise, the positive predictive value is the empirical proportion of true positives among the set of samples classified as positive,

$$PPV_\tau = \frac{|\{n : s_n > \tau, y_n = 1\}|}{|\{n : s_n > \tau\}|} \,. \tag{8}$$

To form a subpopulation-specific estimate for group $A = a$, all of these non-parametric sample-based estimators are simply restricted to examples from a. This restriction reduces the sample size, increasing the variance of the estimator. For large subpopulations, the scale of the variance may be small enough to be acceptable, but for smaller subpopulations the high variance caused by this restriction can result in uninformative estimates of performance.

3. Model-based metric evaluation

The non-parametric sample-based estimators described in the previous section are simple to implement and admit favorable theoretical properties (e.g., unbiasedness). However, naively applying these estimators to small subpopulations will yield noisy estimates. We propose *model-based metric* (MBM) estimates to form sample-efficient estimates of common predictive model metrics. The key idea is that when forming an estimate for subpopulation a, we can borrow information from other, similar subpopulations. To borrow information, we specify an *evaluation model* — a joint model of the prediction score S given class Y, subpopulation A, and covariate information X — over the entire observed population. Specifying a model also allows us to incorporate information from covariates, which can help us more precisely specify this joint distribution. As discussed in the previous section, common performance metrics are functions of these conditional distributions, and can be computed using the fitted evaluation model.

Using an evaluation model is similar to specifying any model to analyze a dataset, requiring a plausible parametric form, an algorithm to fit the model to observations, and a procedure to criticise and choose between models. Here, we describe the steps of our model-based metric evaluation process: (i) specifying the evaluation model, (ii) fitting the evaluation model to data, (iii) checking and validating model fit, (iv) computing model-based metric estimates, and (v) efficiently computing confidence intervals for these model-based estimates.

3.1. Evaluation model specification

The evaluation model is a parametric model of the class conditional distribution of the model score, parameterized by λ, which we denote

$$S \,|\, \{A = a, X = x, Y = y\} \sim Pr_{\mathcal{M}}(S \,|\, A = a, X = x, Y = y, \lambda)\,, \tag{9}$$

where $\mathcal{M}$ indicates that this conditional distribution is defined by the chosen model class $\mathcal{M}$, a indicates the subpopulation, x are additional covariates, and y is the true (binary) class. As a concrete example, for a continuous-valued S, a common assumption might be that the response is conditionally Gaussian,

$$S \,|\, \{A = a, X = x, Y = y\} \sim \mathcal{N}(\mu_{a,x,y}, \sigma^2_{a,x,y})\,, \tag{10}$$

where the mean and variance of this model, $\lambda = \{\mu_{a,x,y}, \sigma^2_{a,x,y} : a \in \mathcal{A}, x \in \mathcal{X}, y \in \mathcal{Y}\}$, are determined by the subpopulation, covariate, and class information.

If we treat each subpopulation determined by a, x, and y independently ("no-pooling," in the parlance of multi-level modeling), we recover a parametric version of the fully independent subsample estimators described in Section 2. However, when we specify shared structure in the parameters $\mu_{a,x,y}$ and $\sigma^2_{a,x,y}$, the model-based subpopulation estimates can use relevant information from related subpopulations — in essence exploring the bias-variance tradeoff. In our empirical analysis, we explore a range of linear models, from simple additive models with homoscedastic errors to more complex multi-level models with pairwise effects and heteroscedastic errors (see Appendix A.2 for details). We emphasize that our approach is fully general, and that many other functional forms for evaluation models are possible for other applications.

3.2. Evaluation model inference

For a given evaluation model class $\mathcal{M}$, we must fit the evaluation model parameters λ to best describe the observed predictive model score data using the available demographic and covariate information. Adopting a Bayesian approach, we integrate over our posterior uncertainty in λ to form the class conditional distributions of interest,

$$Pr_{\mathcal{M}}(S \,|\, A = a, X = x, Y = y, \mathcal{D}) = \int Pr_{\mathcal{M}}(S \,|\, A = a, X = x, Y = y, \lambda) Pr_{\mathcal{M}}(\lambda \,|\, \mathcal{D}) d\lambda\,. \tag{11}$$

This is the posterior predictive distribution, which is the model likelihood averaged over the posterior $Pr_{\mathcal{M}}(\lambda \,|\, \mathcal{D})$ given *all of the data* $\mathcal{D}$. This averaging results in a more expressive mixture distribution than the model conditioned on a single value of λ.

As discussed above, metrics of interest — such as AUC, PPV, and FPR — are functions of this class conditional distribution. Analogously, our model-based metric estimates are functions of the posterior predictive distribution of Equation 11. To compute such metrics, we use Markov chain Monte Carlo (MCMC) to approximate the posterior predictive distribution. We simulate R samples, $\{\lambda^{(r)}\}_{r=1}^{R}$ from the posterior distribution using the No U-turn Sampler (NUTS) (Hoffman and Gelman, 2014), a variant of Hamiltonian Monte Carlo. These samples can then first be used to perform model checking and validation, and then compute model-based metrics as described in Section 3.4.

3.3. Evaluation model checking

When forming a model-based estimate, a first order concern is the fidelity of the evaluation model to the true distribution generating the observed prediction model scores. To validate and select a model, we compare the out-of-sample log-likelihood estimates between evaluation models as well as a kernel density estimate (KDE) formed within each subpopulation $A = a$ as a baseline. The KDE is a surrogate model for the empirical metrics — if the KDE's out-of-sample log likelihood is significantly better than the evaluation model's, this indicates that the evaluation model is likely overfit and will yield invalid results. In this event, we iterate on the model class. If no evaluation model is successful in comparison to the KDE, we suggest reverting back to the non-parametric subsample estimator. While we focus on model score comparison, we also suggest graphical posterior predictive checks (Gelman et al., 2000).

The underlying assumption of this approach is that a more accurate approximation of the class conditional distribution will result in a more accurate metric estimate. While we simply compare cross-validated log-likelihood for each model, a more conservative approach could treat the KDE as the null and only accept the MBM estimator if some test statistic is significantly greater. Further, comparisons between models (particularly on a shared sample) may be more reliable than estimating the true generalization error for each model (Recht et al., 2018).

3.4. Computing model-based metric estimates

To compute model-based metric estimates, we use samples from the posterior predictive distribution. For each posterior sample $\lambda^{(r)}$, the model scores $s_1, \ldots, s_n$ are simulated from the model conditional distribution, resulting in simulations $\{s_{n,r}^{(sim)}, y_n, a_n, x_n\}_{n=1,r=1}^{N,R}$. Intuitively, this is a larger dataset of $N \times R$ model-based simulations that can then be plugged into the standard nonparametric AUC, FPR, and PPV estimators, resulting in an approximation of the Bayes estimate of each model metric.

3.5. Confidence intervals and approximating the bootstrap

For an evaluation model $\mathcal{M}$ and subpopulation, the theoretical Bayes estimate of a model metric is a *point estimate* — a scalar that is a deterministic function of the observed data $\mathcal{D}$. To see this, consider that as the number of posterior simulations R grows, we eliminate Monte Carlo error from the MCMC sampling routine. As such, the posterior predictive

class conditional distributions become *deterministic* functions of a, x, y, and $\mathcal{D}$, making the corresponding metric estimate also deterministic (though in practice, with finite R, there will be a small amount of Monte Carlo error).

Typically, we also want to characterize the uncertainty of this point estimate, for example, by computing frequentist confidence intervals. One popular approach is the bootstrap (Efron, 1992). In our setting, this involves sampling a boostrapped dataset $\mathcal{D}_b$, simulating from the posterior predictive distribution given $\mathcal{D}_b$, and computing the model-based metric estimate — repeated B times, where practically B is at least 100. Bootstrapping Bayes estimates is similar to using a bagged posterior (Bühlmann, 2014; Huggins and Miller, 2019, 2020).

While conceptually straightforward, it can be computationally prohibitive to simulate R posterior samples $B > 100$ times, particularly for hierarchical models with high-dimensional parameters λ. For such models, we propose an approximation to the bootstrap that reuses a single set of posterior samples given the full data $\mathcal{D}$. To generate a single bootstrap estimate, we re-weight each posterior sample in a way that approximates the bootstrapped posterior. This approach follows a similar strategy to the approximate leave-one-out cross validation estimates developed in Vehtari et al. (2017) and Vehtari et al. (2015).

Concretely, given a set of posterior simulations, $\Lambda \triangleq \{\lambda^{(r)}\}_{r=1}^{R}$, $\lambda^{(r)} \sim Pr_{\mathcal{M}}(\lambda \,|\, \mathcal{D})$, we use importance-weighted posteriors to approximate a set of bootstrapped estimators. For each bootstrap dataset $\{\mathcal{D}_b\}_{b=1}^{B}$, we compute truncated self-normalized weights for each posterior sample r as follows:

$$\tilde{w}_{b,r} = \frac{Pr_{\mathcal{M}}\left(\lambda^{(r)} \,|\, \mathcal{D}_b\right)}{Pr_{\mathcal{M}}\left(\lambda^{(r)} \,|\, \mathcal{D}\right)} = \frac{Pr_{\mathcal{M}}\left(\mathcal{D}_b \,|\, \lambda^{(r)}\right)}{Pr_{\mathcal{M}}\left(\mathcal{D} \,|\, \lambda^{(r)}\right)} \tag{12}$$

$$\tilde{w}_{b,r}^{(trunc)} = \min\left(\tilde{w}_{b,r}, \sqrt{R} \cdot \frac{1}{R}\sum_{r=1}^{R} \tilde{w}_{b,r}\right) \tag{13}$$

$$w_{b,r} = \frac{\tilde{w}_{b,r}^{(trunc)}}{\sum_{r=1}^{R} \tilde{w}_{b,r}^{(trunc)}}\,. \tag{14}$$

The evaluation model prior terms cancel out, simplifying to the likelihood ratio between the bootstrap dataset $\mathcal{D}_b$ and the original dataset $\mathcal{D}$. Next, we construct a set of posterior samples approximately distributed according to the *bootstrapped posterior*, $\lambda_b^{(r)} \sim Pr_{\mathcal{M}}(\lambda \,|\, \mathcal{D}_b)$, by sampling from the original set of posterior samples Λ proportional to the truncated weights with replacement, $\lambda_b^{(r)} \sim \text{Categorical}(\Lambda, w_b)$, resulting in an approximate bootstrap posterior sample $\Lambda_b = \{\lambda_b^{(r)}\}_{r=1}^{R}$ We then compute the desired model-based metrics for each sample Λ_b, and repeat the process B times.

Crucially, the importance-weighted approximation requires computing posterior samples for only one model, while the typical bootstrap estimator requires computing posterior samples for *each* of the B bootstrapped datasets. Figure 7 in the Appendix shows that the confidence intervals for different performance metrics obtained via this efficient approximation closely resemble the intervals obtained from the standard (expensive) bootstrapping procedure.

4. Method summary

To summarize, we propose the following procedure to produce model-based estimates of common ML model performance metrics in subpopulations:

- Specify an evaluation model, $Pr_{\mathcal{M}}(S \mid A = a, X = x, Y = y, \lambda)$.
- Simulate posterior samples from the evaluation model, $\{\lambda^{(r)}\}_{r=1}^{R}$, $\lambda^{(r)} \sim Pr_{\mathcal{M}}(\lambda \mid \mathcal{D})$, given a dataset $\mathcal{D}$ of predicted scores from the ML model.
- Validate the evaluation model by estimating out-of-sample performance with stratified cross validation (or approximate leave-one-out cross validation (Vehtari et al., 2015))
  - Compare to subpopulation-specific KDE log-likelihoods: if the evaluation model underperforms the KDE on a subpopulation, revert to the typical subsample-based estimate of model performance, or revise the evaluation model and refit.
- Form model-based metric estimates for each subpopulation from posterior predictive simulations $\{s_{n,r}^{(sim)}\}$, $\theta(x) = f(\{s_{n,r}^{(sim)}\})$.
- Form bootstrap-simulated model-based metric estimates:
  - Simulate bootstrap dataset $\mathcal{D}_b^*$.
  - Compute posterior samples using $\mathcal{D}_b^*$ (alternatively, approximate $Pr_{\mathcal{M}}(\lambda \mid \mathcal{D}_b^*)$ with importance weights).
  - Compute relevant model-based metric estimates $\theta_b^*(x)$ for each subpopulation x.
- Return point estimate $\theta(x)$ and bootstrap samples $\{\theta_b^*(x)\}_{b=1}^{B}$.

This procedure results in a point estimate and bootstrap samples of the performance metrics of interest. In addition to the subpopulation-specific metrics, comparisons between subpopulations can also be computed using the evaluation model posterior samples. See Appendix A.2 for evaluation models used in our empirical study.

4.1. Related work

The introduction of a model and Bayesian inference for estimating model metrics has been discussed before, for computing accuracy (Benavoli et al., 2017), true positive rate (Johnson et al., 2019), and precision-recall (Goutte and Gaussier, 2005). The most similar work to ours focuses on the use of unlabeled data to tighten estimates of subpopulation performance, using a model parameterized directly by the metric of interest (e.g., TPR) (Ji et al., 2020). This work differs from ours in two important ways: (i) their approach focuses on specific threshold-based metrics (e.g., TPR, FPR), and does not address statistics like the AUC or other concordance metrics, and (ii) uncertainty is characterized by Bayesian credible intervals, and not frequentist confidence intervals. Our framework is more general, as we model the class conditional distribution of the predictive model score directly, from which all metrics of interest can be computed in a single procedure. As such, our evaluation models are not restricted to just one metric defined at one particular threshold.

Regarding estimate uncertainty, in a series of articles, Huggins and Miller (2019) and Huggins and Miller (2020) discuss "bagged posteriors", or an ensemble of posterior distributions conditioned on bootstrapped datasets for forming robust estimates and for model selection.

They explore the theoretical properties of such estimators, showcasing their robustness in the presence of model misspecification, motivating their use in constructing our estimator. They note that the bagged posterior combines the desirable properties of Bayesian and frequentist methods: flexible hierarchical estimators that average out nuisance parameters with robustness to sampling variability.

Our problem setting and model-based approach are similar in spirit to the goals of meta-analysis (Lipsey and Wilson, 2001), where Bayesian multi-level models have been used extensively to partially pool information across related groups and studies (Gelman et al., 2013). Oakden-Rayner and Palmer (2020) discusses the intersection of meta-analysis and machine learning model evaluation, focusing on the comparison of model predictions to predictions from a panel of multiple experts. This work makes clear that simple averages of sensitivity and specificity across experts typically underestimate expert performance. Combining expert labels requires a meta-analytic approach, which is conceptually related to the way we construct model-based estimates of metrics such as AUC.

Lastly, Hanczar et al. (2010) discuss reliability of small-sample ROC (and FPR) estimates in biological settings. They caution that a procedure for error estimation can only provide as much information as is present in the data at hand; unrepresentative samples will yield poor error estimates. We observe the same phenomenon, and we attempt to address such shortcomings within individual subpopulations by incorporating related information into our subpopulation estimates.

5. Empirical evaluation

We apply model-based metric estimates in two settings. First, we consider a semi-synthetic scenario using demographic and biomarker statistics from the National Health and Nutrition Examination Survey (NHANES) (National Center for Health Statistics, 2017) to mimic a cardiovascular disease risk prediction task, creating a scenario where ground truth metrics are available. Then, we evaluate a model for predicting hospital readmission among diabetes patients, using data from a large, multi-center study (Strack et al., 2014).

While evaluation models can be in any flexible model class — e.g., a Bayesian neural network or a Gaussian process — we focus on a set of increasingly flexible models for which statistical inference is quite reliable. In doing so, we avoid concerns about poor posterior inference influencing the accuracy of our metric estimates. We define a sequence of increasingly complex hierarchical models (described below) and use MCMC for statistical inference. To specify and fit evaluation models, we use *Stan* (Carpenter et al., 2017) and the *brms* package in R (Bürkner, 2017).

5.1. Semi-synthetic Framingham risk prediction (`semi`)

Problem setup We set up a semi-synthetic prediction task that targets the Framingham risk score, a metric of cardiovascular health that aggregates age, blood pressure, cholesterol, diabetes, hypertension, and smoking status (D'agostino et al., 2008). To simulate an observation, we first randomly draw from groups defined by age bins, sex, race, and BMI in proportion to their empirical frequencies in NHANES. For each simulated individual, we then sample Framingham risk factors from group-specific aggregated statistics in the

NHANES population. We construct an outcome variable by computing the Framingham risk score (D'agostino et al., 2008), and using 10% risk as a cutoff to define two classes — low and elevated risk of cardiovascular disease (Bosomworth, 2011).[3]

To mimic a machine learning model predicting this synthetic target, we simulate prediction model scores S in three ways: (i) a score with no relationship to subpopulation group and covariates (`none` in the results), (ii) a score with a simple additive relationship to age, race, sex, and BMI (`simple`), and (iii) a score with interactions between race and age, and sex and BMI (`interactions`). Additionally, for each type of simulated prediction model scores, we add either homoscedastic or heteroscedastic (denoted `-hetero` in the results) noise. See Appendix A.1 for full details of each of the six settings.

For each of these six simulated ML models, we generate a large dataset with $N^{(pop)} = 5 \times 10^6$ observations to use as ground truth. We then take a smaller subsample (e.g., $N = 10{,}000$, $N = 5{,}000$, and $N = 1{,}000$) to simulate a more realistic limited data setting. We construct MBM estimates and the nonparametric subsample-based estimates using this smaller dataset.

Evaluation models We report the AUC, FPR, and PPV metric estimates from both the typical subsample estimator (denoted `empirical`) and a set of evaluation models — thresholds for FPR and PPV are set such that the full population FPR is 1%. We use a set of evaluation models of varying complexity:

- `fixed.b`: a fixed effects model with the following linear predictors: demographics (gender, race, age bin), BMI, true class Y, and additional covariates (blood pressure and lipids).
- `fixed.c`: a fixed effects model containing all marginal and pairwise interactions between demographics, BMI, and Y, as well as the additional covariates in `fixed.b`.
- `rand.a`: a random effects model consisting of a random intercept for all marginal and pairwise interactions between demographics, BMI, and Y, as well as the additional covariates in `fixed.b` as fixed effects.

We also explored heteroscedastic evaluation models with a noise variance modeled as log-linear with subpopulation covariates, but found results to be similar to those with constant variance. Full experiment and evaluation model details are in Appendix A.1.

5.2. Diabetes hospital readmission prediction (`dm`)

Problem setup We also apply MBMs to a more complex scenario involving a real healthcare prediction task. We use a large, publicly available health dataset that was constructed to understand potential factors driving hospital readmissions among individuals with diabetes (Strack et al., 2014; Dua and Graff, 2017). After filtering to only adult admissions and removing observations with missing demographics, the dataset consists of 96,229 hospital admissions among 67,467 unique individuals. The prediction problem is to use information

3. We emphasize that this is a hypothetical prediction task, and highlight that the development of the Framingham cardiovascular risk score itself suffered from exactly the subpopulation recruitment issues that we describe, necessitating follow-up studies targeting underrepresented groups (Kanaya et al., 2013).

available on discharge (e.g. length of stay, principal diagnoses, hemoglobin A1c result, diabetes medications) to predict the binary outcome whether the individual is ever readmitted to the hospital for a diabetes-related problem.

We first construct an XGBoost (Chen and Guestrin, 2016) model to predict the binary outcome of interest, using a total of 105 features after categorical variables are one-hot encoded, treating the predictions made by this model as a form of ground truth. We use this predictive score to compute subpopulation AUCs (and FPRs and PPVs) using the typical non-parametric estimators. Next, we take a random subsample of size 5,000 patients, and apply our MBM estimators to the subsample. We then draw similar comparisons as for `semi` — we measure the accuracy and coverage of model-based estimators and the subsample non-parametric estimators to the full-sample non-parametric estimators. We repeat this sub-sampling procedure 10 times to estimate statistics over the data distribution.

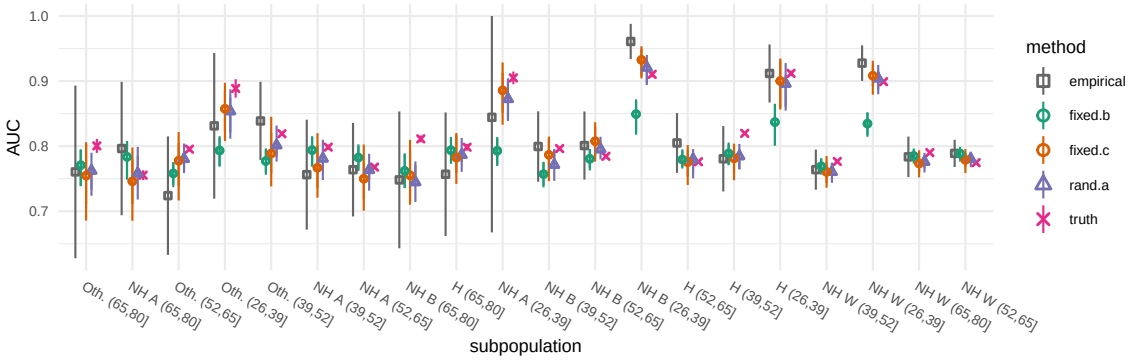
Evaluation models We use a similar set of evaluation models as in the `semi` task, differing only in the specific subpopulations and additional covariates. `fixed.b` again denotes a fixed effects model with demographic subpopulation and additional covariates as linear predictors. Similarly, `fixed.c` contains the same variables as in `fixed.b`, but now allows for all marginal and pairwise interactions between demographic variables. Lastly, `rand.a` is a random effects model with intercepts for all marginal and pairwise interactions between demographic variables, with additional covariates as fixed effects. Full details on the `dm` experiment settings and evaluation models can be found in the Appendix A.2.

5.3. Results

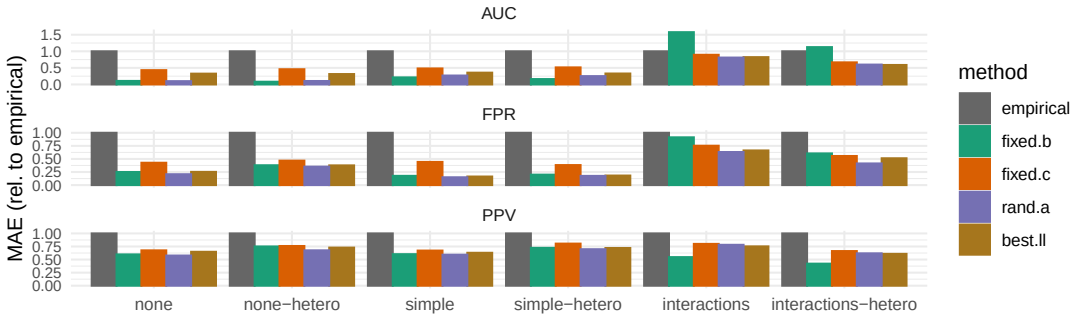
We highlight the behavior of MBMs and findings from our two experimental settings.

MBMs generally improve estimates of AUC, FPR, and PPV. Figure 2 depicts results from the `semi` task, obtained from a random subset of $N = 10{,}000$ observations. Figure 2b shows the relative error in estimating AUC, FPR, and PPV, averaged over all subpopulations. `best.ll` denotes the results obtained by selecting the best-fitting evaluation model (i.e., either the `empirical` KDE, or one of `fixed.b`, `fixed.c`, or `rand.a`) within each subpopulation, and then averaging these results across subpopulations. Across all six simulation settings, at least one of the MBMs produces more accurate estimates of AUC, FPR, and PPV compared to the non-parametric subsample estimates (`empirical`).

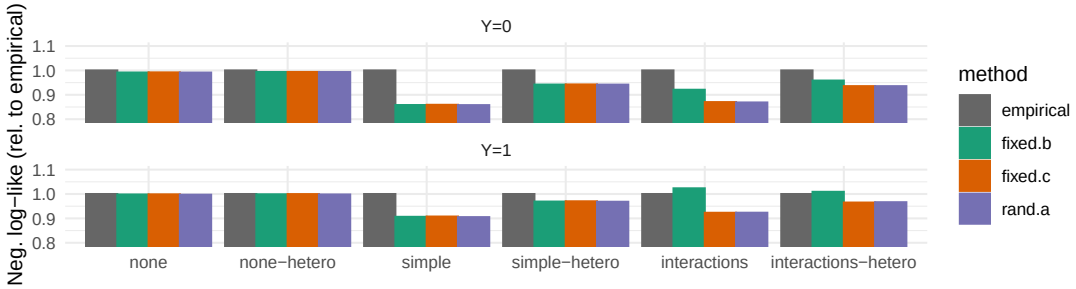
Improvements are most substantial for smaller subpopulations. Figure 2a depicts AUC estimates for the `semi` experiment, with subpopulations ordered left to right from smallest to largest. Comparing estimates and uncertainties to the true population AUC, we find that the more expressive evaluation models `fixed.c` and `rand.a` form much more accurate estimates, particularly in subpoulations with smaller samples. Figure 6 further illustrates that this effect is even more pronounced when the overall subsample size is smaller. Figure 3b shows estimates of the AUC for different subpopulations on the `dm` dataset. As with `semi`, the improvements are most dramatic for smaller groups; MBMs always have tighter confidence intervals than the empirical estimator, and often have more accurate point estimates. Figure 5 in the appendix contains additional AUC results for other sample sizes.



(*a*) AUC estimates and 95% confidence intervals by subpopulation, under the most complex `interactions-hetero` simulation.



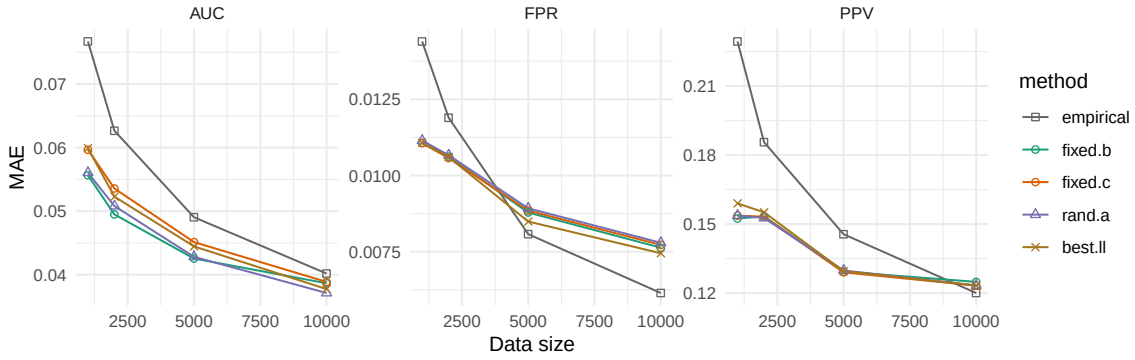
(*b*) Relative mean absolute percentage error (lower is better — 1 is scaled to the empirical subsample estimator error).



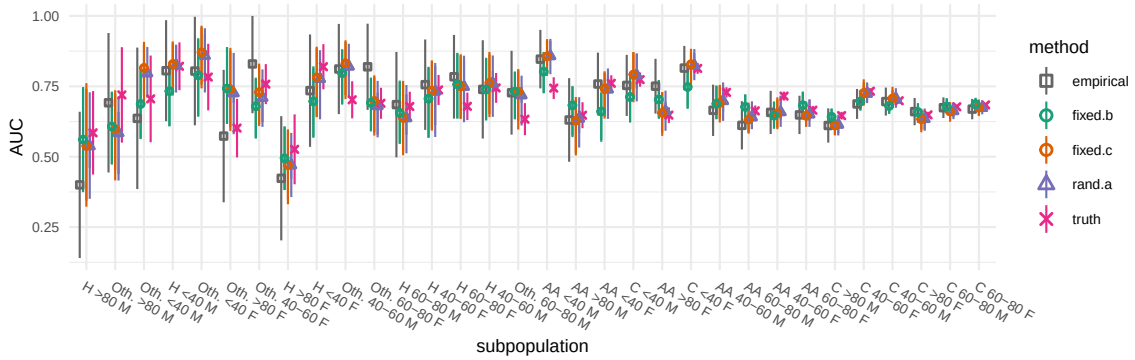
(*c*) Relative negative log-likelihood (lower is better — 1 is scaled to the KDE).

Figure 2: Results from a run of the `semi` experiment. Descriptions of each method can be found in the text or Appendix. Results are obtained using a random sample of 10,000 individuals from the total simulated population of five million. Abbreviations for demographic subgroups: "H": Hispanic, "NH A": non-Hispanic Asian, "NH B": non-Hispanic African American, "NH W": non-Hispanic Caucasian.

For the `dm` experiment, Figure 3a shows the mean absolute error for MBMs and the sample-based estimators as a function of the subsample size (N), averaging over subpopulations. We see a strong sample-size effect: MBMs offer the greatest improvements over the typical



(*a*) Estimator error by sample size



(*b*) Subpopulation AUC estimator comparison

Figure 3: Results of the `dm` experiment. (a) Estimator error as a function of data size — we observe larger benefits in the smaller sample range across all three statistics. (b) A breakdown of AUC estimates for each subpopulation, comparing to the "true" AUC (based on the full dataset of 96,229 observations). The empirical and various MBM estimates are from subsamples of size 5,000. Subpopulation abbreviations: "H": Hispanic, "Oth.": Other, "AA": African American, "C": Caucasian, "M": Male, "F": Female

subsample-based estimator when sample sizes are smallest. Figure 4 contains additional results for both datasets showing performance broken down by subpopulation size.

Model misspecification affects estimate accuracy. A complex but misspecified evaluation model can perform worse than a much simpler model. In Figure 2b the simpler evaluation model `fixed.b` consistently outperforms the more complex `fixed.c` in the four `none` and `simple` settings. This makes sense, as the many pairwise interactions in `fixed.c` are not necessary to well capture the true data generating mechanism.

However, overly simple models can also perform worse than the standard empirical estimator. In the two `interactions` settings, the `fixed.b` model-based estimates have much higher error in estimating AUC than the sample-based estimator. This is also clear from Figure 2a, even for some of the larger subgroups. Fortunately, inspecting out-of-sample log-likelihoods in Figure 2c identifies these as situations where the estimate should not be trusted. Likelihoods for the `fixed.b` model when $Y = 1$ are worse than the KDE baseline, indicating that one should either iterate on the model or revert to the empirical estimator.

Importance-weighted bootstrap confidence intervals are a faithful approximation with reasonable coverage for AUC estimates. Figure 7 in the appendix compares our proposed approximate bootstrap procedure to exact bootstrapping, and shows good agreement in their respective confidence intervals. Figure 8 summarizes subpopulation estimate coverage properties. MBMs exhibit slight overconfidence, even when using the traditional bootstrap resampling estimator, but the importance-weighted and bootstrap estimates achieve similar coverage.

6. Discussion

There is no substitute for better data — validating a prediction model with limited information is a fraught exercise. However, in some situations we may have a modest sample size overall but limited coverage for some specific subpopulations. We have shown that we may be able to leverage information from other related subgroups to improve performance estimates for these smaller subpopulations. To accomplish this task, we proposed model-based metric (MBM) estimators, a procedure to construct more accurate predictive model performance estimates for small subpopulations. MBMs offer a promising middle ground between either needing to collect additional data or being unable to draw reliable inferences about the smallest subpopulations due to extreme variance. We designed experiments that investigate trade-offs between simple and complex evaluation models and how they interact with varying sample sizes in both a semi-synthetic and real data setting. We found that in many settings with small subgroups, MBMs tend to have narrower confidence intervals than the naive subsample-based estimators, and are often more accurate as well.

The difficult task of estimating ML performance for small subpopulations raises numerous issues, in both statistical methodology, fairness implications, and their intersection. Better theoretical understanding of these model-based estimators, the use of more flexible non-linear model components, the incorporation of covariate uncertainty, and extensions to other metrics and non-classification settings all warrant further study.

The partial pooling estimators we deploy use information across groups by design, making subgroup estimates correlated with one another. This can potentially complicate the estimation of disparity between groups — an area that requires further study. Further, good theoretical coverage properties in the multi-group setting requires the development of more sophisticated estimators (Yu and Hoff, 2018). Validating and comparing evaluation models is another area requiring further examination. The underlying assumption of our approach is that more accurate approximations of the class conditional distributions will yield more accurate metric estimates. Cross-validated log likelihood may be a more appropriate for estimating certain metrics (e.g., AUC) than others (e.g., FPR).

We also adopted a simple approach for building flexible models when using continuous-valued covariates (e.g., age) — we discretize the feature and fit a coefficient for each discrete level. There are a multitude of more sophisticated approaches for incorporating non-linear conditional models, including splines, Gaussian processes, or even neural networks. We used a simple class of single- and multi-level models to ensure statistical inference was reliable and efficient, but we anticipate that the incorporation of non-linear modeling components will yield additional benefits.

Uncertainty in the covariates themselves is another potential issue. The evaluation model conditions on a set of demographic and additional covariates, and ignoring uncertainty in these measurements can be a source of model misspecification. Techniques for incorporating these uncertain measurements into the class conditional model may yield more reliable estimates of model performance.

Lastly, in this work we focused on three commonly used metrics for evaluating binary classifiers: AUC, FPR, and PPV. These metrics are purely discriminative, or rank-based; future work should also examine metrics relating to model calibration, or how well a model's predicted probabilities align with true event probabilities. Furthermore, we only applied MBMs to binary classification settings, but in principle they may also be applied for modeling other types of outcomes, e.g., categorical, continuous-valued, or survival data.

Model-based metric estimators can also be incorporated into the process of building a model card (Mitchell et al., 2019) for the predictive ML model. These estimates could help characterize the potential effects a model might have on different subpopulations, and highlight groups for which more data collection is needed.

References

Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. *The Journal of Machine Learning Research*, 18(1):2653–2688, 2017.

N John Bosomworth. Practical use of the Framingham risk score in primary prevention: Canadian perspective. *Canadian Family Physician*, 57(4):417, 2011.

Peter Bühlmann. Discussion of big Bayes stories and BayesBag. *Statistical Science*, 29(1): 91–94, 2014.

Paul-Christian Bürkner. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1):1–28, 2017. doi: 10.18637/jss.v080.i01.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1), 2017.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Ralph B D'agostino, Ramachandran S Vasan, Michael J Pencina, Philip A Wolf, Mark Cobain, Joseph M Massaro, and William B Kannel. General cardiovascular risk profile for use in primary care. *Circulation*, 117(6):743–753, 2008.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in Statistics*, pages 569–593. Springer, 1992.

Andrew Gelman, Yuri Goegebeur, Francis Tuerlinckx, and Iven Van Mechelen. Diagnostic checks for discrete data regression models using posterior predictive simulations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(2):247–268, 2000.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC press, 2013.

Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European Conference on Information Retrieval*, pages 345–359. Springer, 2005.

Blaise Hanczar, Jianping Hua, Chao Sima, John Weinstein, Michael Bittner, and Edward R Dougherty. Small-sample precision of ROC-related estimates. *Bioinformatics*, 26(6):822–830, 2010.

Matthew D Hoffman and Andrew Gelman. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1): 1593–1623, 2014.

Jonathan H Huggins and Jeffrey W Miller. Robust inference and model criticism using bagged posteriors. *arXiv preprint arXiv:1912.07104*, 2019.

Jonathan H Huggins and Jeffrey W Miller. Robust and reproducible model selection using bagged posteriors. *arXiv preprint arXiv:2007.14845*, 2020.

Disi Ji, Padhraic Smyth, and Mark Steyvers. Can I trust my fairness metric? assessing fairness with unlabeled data and bayesian inference. *arXiv preprint arXiv:2010.09851*, 2020.

Wesley O Johnson, Geoff Jones, and Ian A Gardner. Gold standards are out and bayes is in: Implementing the cure for imperfect reference tests in diagnostic accuracy studies. *Preventive Veterinary Medicine*, 167:113–127, 2019.

Alka M Kanaya, Namratha Kandula, David Herrington, Matthew J Budoff, Stephen Hulley, Eric Vittinghoff, and Kiang Liu. Mediators of atherosclerosis in south asians living in america (masala) study: objectives, methods, and cohort description. *Clinical cardiology*, 36(12):713–720, 2013.

Mark W Lipsey and David B Wilson. *Practical meta-analysis.* SAGE publications, Inc, 2001.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229, 2019.

Urs J Muehlematter, Paola Daniore, and Kerstin N Vokinger. Approval of artificial intelligence and machine learning-based medical devices in the USA and Europe (2015–20): A comparative analysis. *The Lancet Digital Health*, 2021.

National Center for Health Statistics. National health and nutrition examination survey data. Hyattsville, MD: U.S. Department of Health and Human Services, Centers for Disease Control and Prevention, 2017.

Luke Oakden-Rayner and Lyle Palmer. Docs are ROCs: A simple off-the-shelf approach for estimating average human performance in diagnostic studies. *arXiv preprint arXiv:2009.11060*, 2020.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv preprint arXiv:1806.00451*, 2018.

Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed Research International*, 2014, 2014.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling. *arXiv preprint arXiv:1507.02646*, 2015.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and Computing*, 27(5):1413–1432, 2017.

Chaoyu Yu and Peter D Hoff. Adaptive multigroup confidence intervals with constant coverage. *Biometrika*, 105(2):319–335, 2018.

Appendix A. Experimental setup details

A.1. Semi-synthetic NHANES experiment (`semi`)

We generate data using statistics from the 2017-2018 National Health and Nutrition Examination Survey, available at https://wwwn.cdc.gov.

We devise experiments in three increasingly complex data generating procedures, designed to mimic the situation where a machine learning model aims to predict cardiovascular disease risk in two categories — low and elevated. In all settings, we generate a population that mimics the NHANES demographic frequencies, as well as the statistics of common cardiovascular risk factors — defined by diabetes status, hypertensive treatment status, systolic blood pressure, total cholesterol, and HDL cholesterol.

Before generating a synthetic population, we first compute statistics within each demographic category — defined by race, age buckets, sex, and body mass index (BMI) buckets — of cardiovascular risk factors. Within each demographic bucket, we describe the continuous risk factors — log systolic blood pressure, log total cholesterol, and log HDL cholesterol — with a multivariate Gaussian distribution, and diabetes and hypertensive treatment using independent probabilities. To generate a unit, we first randomly draw the demographic categories from the weighted NHANES units, then draw continuous and discrete risk factors. For each unit, we then compute the Framingham CVD risk score using the simulated risk factor values (D'agostino et al., 2008). We construct an outcome variable by computing the Framingham risk score, and use 10% risk as a cutoff to define two classes — low and elevated risk of cardiovascular disease (Bosomworth, 2011). This process results in a dataset of demographic subpopulation, covariate values, and a binary outcome, $\{a_n, x_n, y_n\}_{n=1}^{N_{pop}}$, where $N_{pop} = 5e6$ units.

Next, we construct a machine learning model score, aimed at predicting y_n for each unit. We construct such a score in three, increasingly complex ways.

- *No structure* (`none`): $S_n \sim \mathcal{N}(\mu_{y_n}, \sigma^2)$ — the class conditional in this relationship is independent of all variables except for the true class label, y. In this relationship, the AUC is identical across all subpopulations, though the false positive rate and positive predictive value can vary from subpopulation to subpopulation (due to differences in prevalence).

```
beta.y = c(-3.5, -2.0)
S.mu = beta.y[factor(popdf$frame.class)]
S.sd = 1.25
```

- *Additive structure* (`simple`): $S_n \sim \beta_a + \beta_r + \beta_g + \beta_b + \beta_y + \sigma\epsilon$ — the class conditional mean has additive structure, based on demographic information. This results in variation in AUC, FPR, and PPV between demographic subpopulations. We set values of β with the following R code snippet:

```
Nr = length(unique(popdf$race))
Na = length(unique(popdf$age_bin))
Nb = length(unique(popdf$bmi_bin))
beta.gender = c(-.2, .2)
beta.race   = seq(from=-.2, to=.2, length=Nr))
```

```
beta.age    = seq(from=-.2, to=.2, length=Na))
beta.bmi    = seq(from=-.2, to=.2, length=Nb))
beta.y      = c(-3, -2.5)
S.mu = beta.gender[factor(popdf$gender)] +
       beta.race  [factor(popdf$race)] +
       beta.age   [factor(popdf$age_bin)] +
       beta.bmi   [factor(popdf$bmi_bin)] +
       beta.y     [factor(popdf$frame.class)]
S.sd = .5
```

- *Interactive structure* (`interactions`): $S_n \sim \beta_{a,r} + \beta_{g,b} + \bar{f} + f_n + \sigma \cdot \epsilon$, where the age and race buckets have strong interactions, as well as the gender and BMI buckets. Additionally, we add in the true (simulated) framingham score f_n, shrunken toward its global mean to mimic the compressed output of a typical machine learning classification score. Thsi results in variation in AUC, FPR, and PPV between demographic subpopulations as well as a more difficult modeling problem.

```
# interact race and age
beta.race = seq(from=-sqrt(.2), to=sqrt(.2), length=Nr)
beta.age  = seq(from=-sqrt(.2), to=sqrt(.2), length=Na)
beta.age.race = beta.age %o% beta.race
age.race.idx  = cbind(factor(popdf$age_bin), factor(popdf$race))
# interact gender and bmi
beta.gender = seq(from=-sqrt(.2), to=sqrt(.2), length=Ng)
beta.bmi    = seq(from=-sqrt(.2), to=sqrt(.2), length=Nb)
beta.gender.bmi = beta.gender %o% beta.bmi
gender.bmi.idx = cbind(factor(popdf$gender), factor(popdf$bmi_bin))
# mean value
S.mu = beta.age.race[age.race.idx] +
       beta.gender.bmi[gender.bmi.idx] +
       .7*popdf$frame.score + .3*mean(popdf$frame.score)
S.sd = .5
```

Additionally, we vary the type of class conditional noise between two settings — homoscedastic and heteroscedastic (denoted with the suffix `-hetero`). In the homoscedastic setting, the conditional variance σ^2 is fixed and shared between all simulated units. In the heteroscedastic setting, the conditional variance is a sigmoidal function of the conditional mean, decreasing as the mean increases. This makes the evaluation modeling problem much more complex, requiring us to use distributional models that allow the residual noise variance to vary with the observed covariates in order for the model to be properly specified. The heteroskedsatic error is generated with the following R code snippet:

```
# heteroscedasticity
s.fac = 1 / (1 + exp(S.mu))
s.fac = s.fac / max(s.fac)
S.sd  = s.fac*S.sd + .5*S.sd
```

In addition to the *structure* and *noise heteroscedasticity*, we also examine *subsample sizes*. That is, how accurate are estimates when only $N = 1{,}000$ units are observed, vs $N = 5{,}000$ units, vs $N = 10{,}000$.

A.2. Evaluation models

For both experiments, we use a set of increasingly complex evaluation models to generate metric approximations. The complexity incorporates additional covariates, interactions between demographic buckets, and heterscedastic noise terms. We fit each model using the R package `brms` (Bürkner, 2017), drawing 4,000 posterior samples for each. Additionally, for the fixed effects models we simulate 1,000 draws from 100 bootstrapped posteriors to generate confidence intervals. The random effects models are computationally much more expensive, so we solely rely on approximations of the leave-one-out log-likelihood (Vehtari et al., 2015) and the truncated importance-weighted bootstrap replicates we describe in Section 3.5.

Here we present each model in `brms` formula notation (using the `bf` function for heteroscedastic models).

Semi-synthetic experiment evaluation models (`semi`) The subpopulation covariates we use (derived from NHANES) are `gender`, `race`, and `age_bin`. For additional covariates, we incorporate BMI, and (log) diastolic and systolic blood pressure, total cholesterol, and HDL cholesterol. We examine three fixed effects models, and two random effects models:

- fixed a: only demographic and outcome information

```
fixed.a = "S ~ gender + race + age_bin + bmi_bin + Y"
```

- fixed b: demographic, outcome, and additional covariate information

```
fixed.b = "S ~ gender + race + age_bin + bmi_bin + Y + ln.diabp + ln.
    ppbp + ln.tc + ln.hdl"
```

- fixed c: marginal and all pairwise interactions between demographic variables, and additional covariate information

```
fixed.c = "S ~ (gender+race+age_bin+bmi_bin+Y)^2 + ln.diabp + ln.ppbp +
     ln.tc + ln.hdl"
```

- fixed d: same as fixed c, but with heteroscedastic noise based on demographics

```
fixed.d = bf("S ~ (gender+race+age_bin+bmi_bin+Y)^2 + ln.diabp + ln.
    ppbp + ln.tc + ln.hdl",
             "sigma ~ gender + race + age_bin + bmi_bin + Y")
```

- random effects a: random effects based on all marginal and pairwise interctions between demographic variables, plus population level covariates

```
rand.a = "S ~ (1 | (gender + race + age_bin + bmi_bin + Y)^2) + ln.
    diabp + ln.ppbp + ln.tc + ln.hdl"
```

- random effects b: same as random effects a, but with heteroscedastic noise term based on demographic information

```
rand.b = bf("S ~ (1 | (gender + race + age_bin + bmi_bin + Y)^2) + ln.
    diabp + ln.ppbp + ln.tc + ln.hdl",
            "sigma ~ (1 | (gender + race + age_bin + bmi_bin + Y)^2)")
```

Diabetes rehospitalization experiment evaluation models (`dm`) The subpopulation covariates we used are `gender`, `race`, and `age_bin`. For additional covariates, we incorporate `number_inpatient` (number of previous inpatient admissions), `number_diagnoses` (total number of diagnoses), `number_emergency` (number of previous emergency room visits), `CHF` (history of congestive heart failure) and `number_outpatient` (number of previous outpatient visits). Similar to before, we examine three fixed effects models, and two random effects models:

- fixed a: only demographic and outcome information

```
fixed.a = "S ~ 1 + gender + race + age_bin + Y"
```

- fixed b: demographic, outcome, and additional covariate information

```
static.cov = " + number_inpatient + number_diagnoses + number_emergency
    + CHF + number_outpatient"
fixed.b = "S ~ 1 + gender + race + age_bin + Y" + static.cov
```

- fixed c: marginal and all pairwise interactions between demographic variables, and additional covariate information

```
fixed.c = "S ~ (gender + race + age_bin + Y)^2" + static.cov
```

- fixed d: same as fixed c, but with heteroscedastic noise based on demographics

```
fixed.d = bf("S ~ (gender + race + age_bin + Y)^2" + static.cov,
             "sigma ~ gender + race + age_bin + Y"),
```

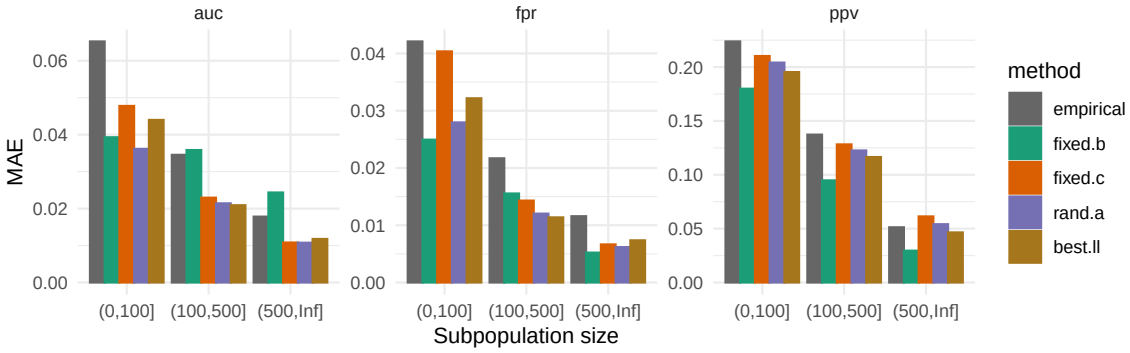
- random effects a: random effects based on all marginal and pairwise interctions between demographic variables, plus population level covariates

```
rand.a = "S ~ (1 | (gender + race + age_bin + Y)^2)" + static.cov
```

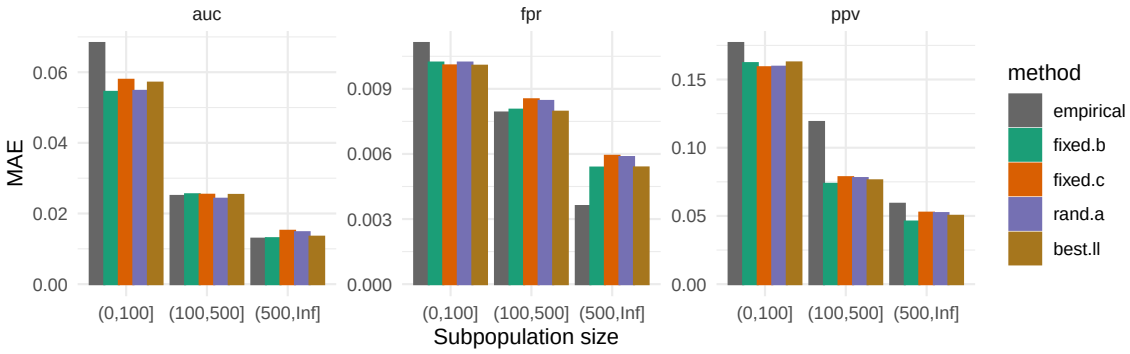
- random effects b: same as random effects a, but with heteroscedastic noise term based on demographic information

```
rand.b = bf("S ~ (1 | (gender + race + age_bin + Y)^2)" + static.cov,
            "sigma ~ (1 | (gender + race + age_bin + Y)^2)")
```

Appendix B. Additional Empirical Results

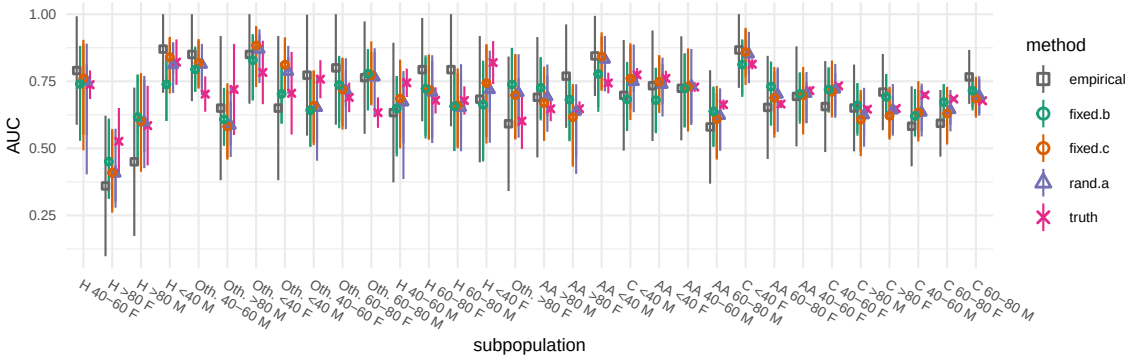


(*a*) `semi` experiment (interactions-hetero)

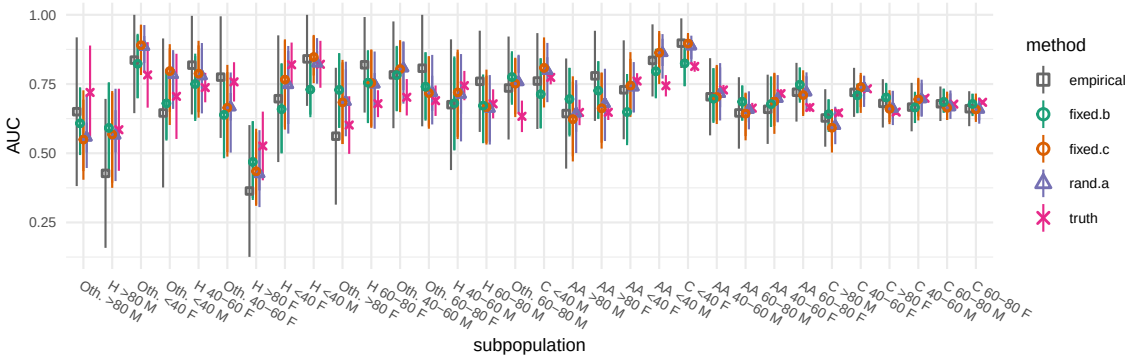


(*b*) `dm` experiment

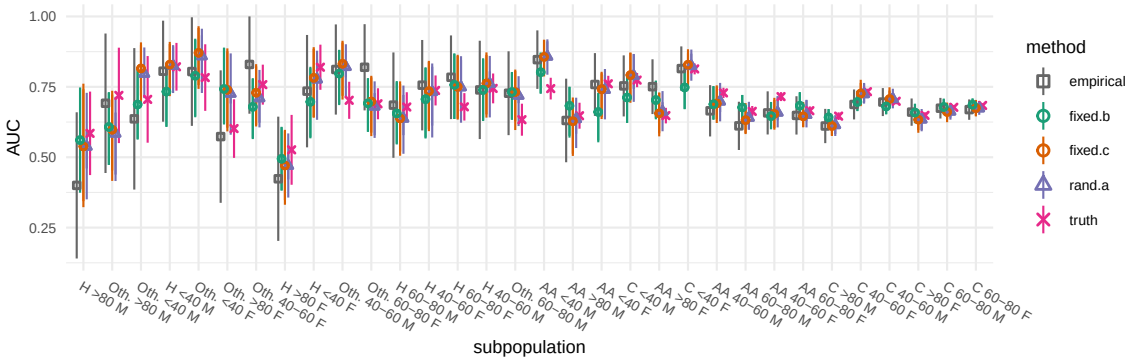
Figure 4: Mean absolute error of AUC, FPR, and PPV estimates by subpopulation size, split into three bins, $(0, 100]$, $(100, 500]$, and > 500. For both the semi-synthetic and real re-hospitalization data, we observe that smaller subpopulations tend to see bigger improvements, while larger subpopulations see similar performance, averaged over the data sampling process.



(*a*) `dm`, $N = 1{,}000$

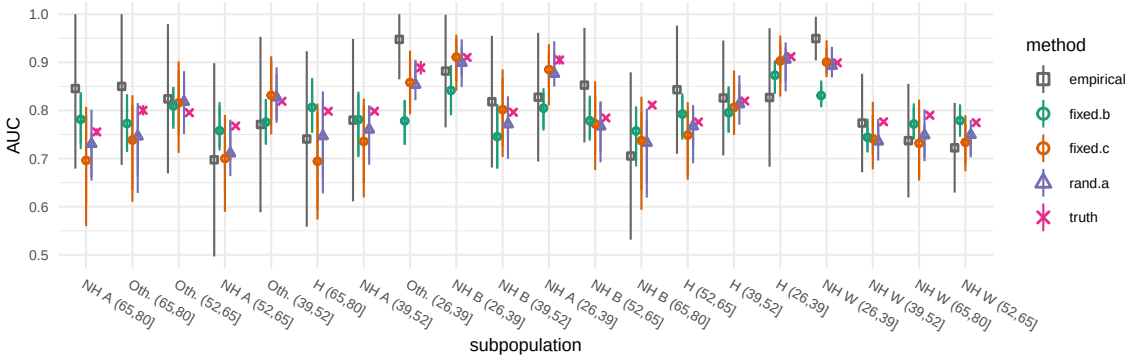


(*b*) `dm`, $N = 2{,}000$

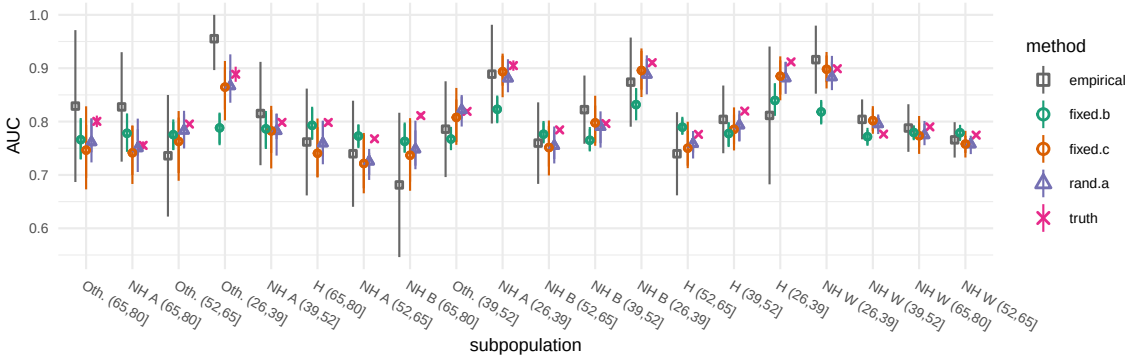


(*c*) `dm`, $N = 5{,}000$

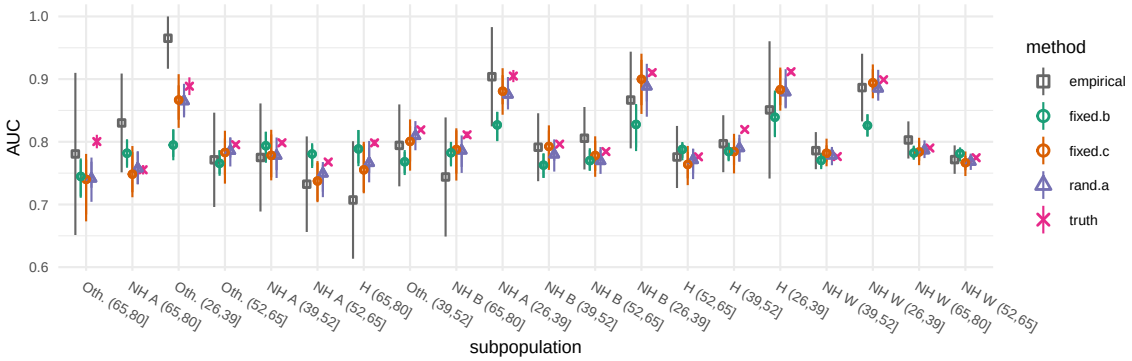
Figure 5: Comparison of AUC estimates by subpopulation for the `dm` experiment for sample sizes (a) $N = 1{,}00$, (b)$N = 5{,}00$, and (c) $N = 10{,}00$. Subpopulations are sorted by size (smallest to largest).



(*a*) `semi` *interactions-hetero*, $N = 1{,}000$

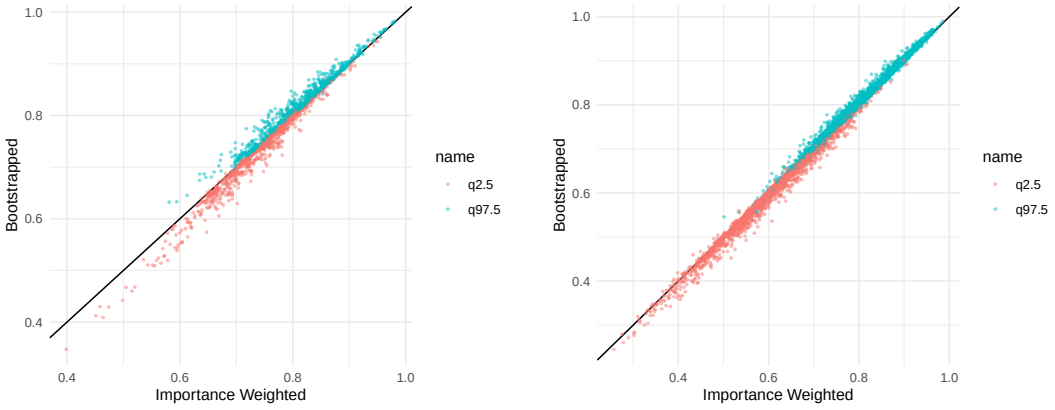


(*b*) `semi` *interactions-hetero*, $N = 5{,}000$



(*c*) `semi` *interactions-hetero*, $N = 10{,}000$

Figure 6: Comparison of AUC estimates by subpopulation for the `semi` experiment for sample sizes for (a) $N = 1{,}00$, (b)$N = 5{,}00$, and (c) $N = 10{,}00$. Subpopulations are sorted by size (smallest to largest).



(*a*) `semi` (*b*) `dm`

Figure 7: Comparison of bootstrap-estimated (slow) quantiles and approximate bootstrap-estimated (faster) quantiles, using re-weighted posterior samples. Despite potential instability of importance weighted estimators, we see tight agreement between the two confidence interval estimators. Note we only compare models `fixed.a`, `fixed.b`, and `fixed.c`, as the multi-level models too expensive to sample from the posterior for each bootstrap data set.

	AUC			FPR			PPV		
method	50%	95%	range	50%	95%	range	50%	95%	range
empirical	0.43	0.92	0.99	0.52	0.83	0.87	0.32	0.77	0.96
fixed.a-boot	0.17	0.46	0.54	0.24	0.54	0.69	0.17	0.51	0.66
fixed.a-iw	0.17	0.43	0.49	0.21	0.45	0.57	0.16	0.50	0.67
fixed.b-boot	0.24	0.54	0.63	0.27	0.65	0.77	0.23	0.53	0.66
fixed.b-iw	0.23	0.51	0.59	0.23	0.56	0.67	0.23	0.51	0.65
fixed.c-boot	0.40	0.87	0.96	0.37	0.82	0.92	0.18	0.53	0.64
fixed.c-iw	0.28	0.72	0.89	0.22	0.65	0.81	0.16	0.46	0.59
fixed.d-iw	0.28	0.76	0.88	0.32	0.68	0.87	0.17	0.51	0.67
rand.a-iw	0.27	0.72	0.87	0.23	0.64	0.81	0.17	0.50	0.61
rand.b-iw	0.28	0.75	0.88	0.34	0.73	0.86	0.18	0.54	0.67

(*a*) `semi` coverage (*interactions-hetero*)

	AUC			FPR			PPV		
method	50%	95%	range	50%	95%	range	50%	95%	range
empirical	0.55	0.95	0.99	0.50	0.58	0.59	0.40	0.53	0.86
fixed.a-boot	0.07	0.22	0.29	0.08	0.18	0.24	0.25	0.54	0.62
fixed.a-iw	0.05	0.16	0.21	0.06	0.13	0.17	0.23	0.52	0.61
fixed.b-boot	0.41	0.87	0.95	0.30	0.53	0.64	0.39	0.72	0.84
fixed.b-iw	0.41	0.86	0.95	0.28	0.50	0.60	0.37	0.71	0.82
fixed.c-boot	0.45	0.89	0.96	0.31	0.52	0.65	0.40	0.71	0.81
fixed.c-iw	0.40	0.83	0.93	0.29	0.49	0.58	0.38	0.68	0.80
fixed.d-iw	0.41	0.86	0.94	0.24	0.51	0.64	0.30	0.64	0.79
rand.a-iw	0.41	0.86	0.94	0.28	0.49	0.58	0.38	0.69	0.79
rand.b-iw	0.42	0.88	0.95	0.25	0.53	0.66	0.32	0.66	0.81

(*b*) `dm` coverage

Figure 8: Empirical coverage of 50% and 95% confidence intervals for AUC, FPR, and PPV, as well as the total range over all (exact or approximate) bootstrap samples. The empirical estimator is reported first. We compare the importance weighted (iw) and full bootstrap (boot) intervals for evaluation models `fixed.a`, `fixed.b`, and `fixed.c`. For evaluation models `fixed.d`, `rand.a`, and `rand.b` we only compute the more efficient importance weighted estimate. We find the `empirical` estimator using the bootstrap to be well-calibrated for AUC, but overconfident for FPR and PPV (i.e. coverage is smaller than it should be). We find that MBMs are also overconfident, but only slightly more or equally so, while also producing more accurate estimates on average. Coverage results for the importance weighted intervals are generally close to results from the full bootstrap procedure.