Auditing Black-Box Prediction Models for Data Minimization Compliance (Supplementary Material)

Bashir Rastegarpanah
Boston University
bashir@bu.edu

Krishna P. Gummadi
MPI-SWS
gummadi@mpi-sws.org

Mark Crovella
Boston University
crovella@bu.edu

A Setting

Figure 1 demonstrates our setting for auditing black-box prediction models as described in Section 3.

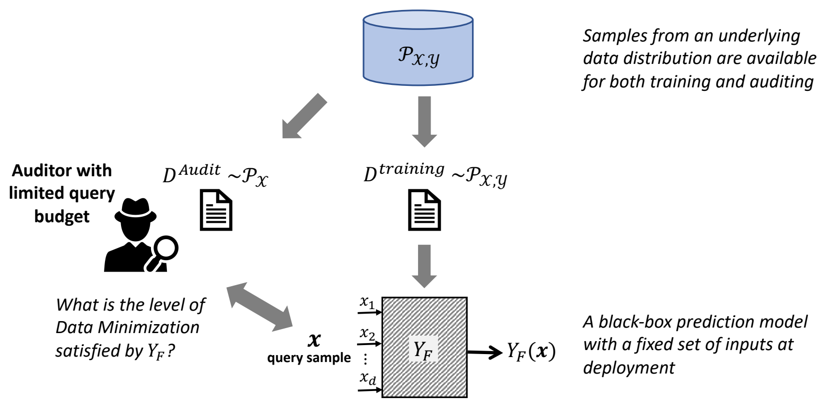
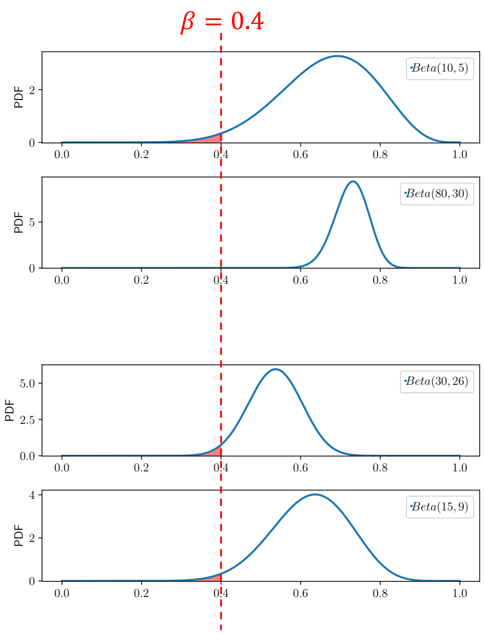


Figure 1

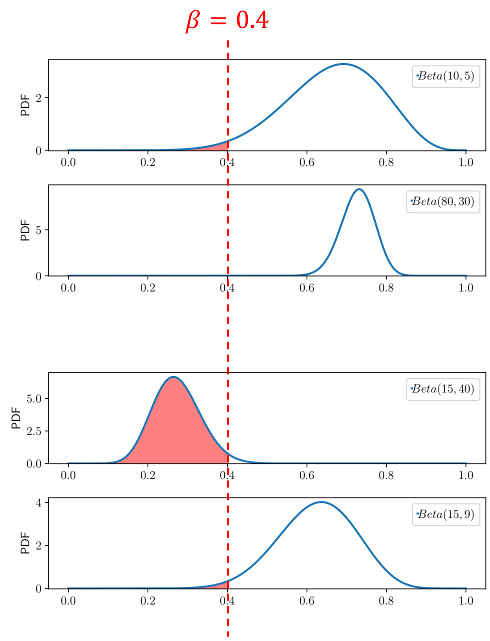
35th Conference on Neural Information Processing Systems (NeurIPS 2021).

B Probabilistic Audit

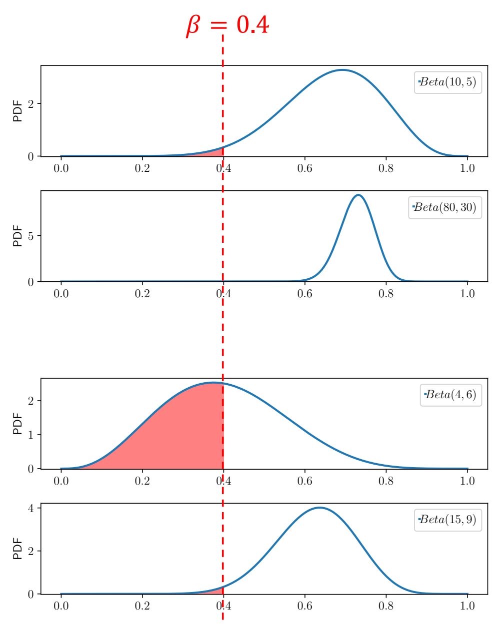
Figure 2 demonstrates examples of how the posterior distributions of model instability with respect to different simple imputations can be used to derive a probabilistic data minimization guarantee.



(a) Data minimization is satisfied.



(b) Data minimization is not satisfied.



(c) An example of a situation where a decision cannot be made based on the posterior distributions.

Figure 2: Example posterior distributions of model instability with respect to different imputations, and a probabilistic data minimization audit at level 0.4 with 90% confidence.

C Pseudocodes for Exploration Strategies

Algorithm 1: Thompson Sampling (TS)

Input: Success and failure counters

for $f_j \in F$ **do**
    for b *in* $\mathcal{X}_j$ **do**
        $\theta_j^b \sim Beta(x; S_j^b + a, F_j^b + c)$
$j^*, b^* = \text{argmin}_{j,b}\, \theta_j^b$
return (f_{j^*}, b^*)

Algorithm 2: Top-Two Thompson Sampling (TTTS)

Input: Success and failure counters

for $f_j \in F$ **do**
    for b *in* $\mathcal{X}_j$ **do**
        $\theta_j^b \sim Beta(x; S_j^b + a, F_j^b + c)$
$j^*, b^* = \text{argmin}_{j,b}\, \theta_j^b$
$K \sim Bernoulli(1/2)$
if *K=1* **then**
    return (f_{j^*}, b^*)
else
    repeat
        for $f_j \in F$ **do**
            for b *in* $\mathcal{X}_j$ **do**
                $\theta_j^b \sim Beta(x; S_j^b + a, F_j^b + c)$
        $\tilde{j}, \tilde{b} = \text{argmin}_{j,b}\, \theta_j^b$
    until $(\tilde{j}, \tilde{b}) \neq (j^*, b^*)$;
    return $(f_{\tilde{j}}, \tilde{b})$

Algorithm 3: Greedy

Input: Success and failure counters, β^*

for $f_j \in F$ **do**
    for b *in* $\mathcal{X}_j$ **do**
        $p_j^b = F_{Beta}(\beta^*; S_j^b + a, F_j^b + c)$
$j^*, b^* = \text{argmax}_{j,b}\, p_j^b$
return (f_{j^*}, b^*)

Algorithm 4: Probability Matching (PM)

Input: Success and failure counters, β^*

for $f_j \in F$ **do**
    for b *in* $\mathcal{X}_j$ **do**
        $p_j^b = F_{Beta}(\beta^*; S_j^b + a, F_j^b + c)$
Randomly choose an arm (f_j, b) with probability $\frac{p_j^b}{\sum_{(f_j,b)} p_j^b}$
return (f_j, b)