Learning through Adaptive Queries: a Directional Derivative Approach

Doyoung Heo [*1] **Hwankyu Sung** [*2] **Sanghwa Kim** [2] **Seungki Min** [3]

Abstract

In the task of Bayesian active learning, a learner aims to identify unknown model parameters by asking a series of queries in an adaptive manner. We consider tasks with binary responses and suggest a simple adaptive query selection rule, called the Directional Uncertainty Reduction Maximization (DURM) algorithm. The algorithm utilizes *directional derivative* of the likelihood function as a proxy for the informativeness of an observation along a particular direction in the parameter space. In each query selection step, it identifies the currently most uncertain direction based on Laplace's approximation and then selects the query with the largest directional derivative along that direction. By doing so, it collects a new observation whose information gain is best aligned with the direction that the information is most needed. We investigate the algorithm's behavior and prove its optimality theoretically for two canonical response models, the logistic model, and the MIRT model. Our numerical experiments confirm these theoretical findings using synthetic data, and further demonstrate that DURM maintains competitive predictive accuracy while providing significant advantages in computational efficiency and posterior concentration on real-world data.

1. Introduction

In modern online services, the platforms actively interact with their users in order to collect user feedback and provide highly personalized services. Feedback from newly joined users is particularly valuable for successful onboarding, and platforms often take aggressive actions. For example, YouTube has recently stopped showing recommended videos to those who do not have a watch history (Android-Police, 2024), and many other platforms require new users to complete a survey to start using their services.

We tackle the decision-making problem that these online platforms face: How to learn about a new user as quickly as possible. More formally, the platform is given a set of queries and sequentially decides which query to ask to receive a response from the user, aiming to identify the user's latent feature with the minimum number of queries. We adopt the Bayesian approach, recognizing that these platforms typically have enough experiences to construct a reasonable prior belief about the new users even before actually interacting with them. Starting from the prior, the platform can sequentially update the belief whenever it receives a user response.

This problem can be framed as a Bayesian active learning task (Settles, 2012), in which the learner has a prior belief, updates its belief with new information, and selects queries to speed up the learning. Many Bayesian active learning algorithms evaluate the informativeness of a query through entropy-based acquisition functions such as Expected Information Gain (EIG; (MacKay, 1992a;b)) and Bayesian Active Learning by Disagreement (BALD; (Houlsby et al., 2011)). These acquisition functions measure the reduction in the entropy of the posterior. However, as pointed out by (Zhao et al., 2021a;b) and (Gontier et al., 2022), computing posterior entropy is often intractable, making these algorithms computationally inefficient and their theoretical analysis challenging.

We propose a simple, computationally efficient heuristic for this problem. Our suggested algorithm, called Directional Uncertainty Reduction Maximization (DURM), utilizes *directional derivative* of the (log-)likelihood, focusing not only on the magnitude but also on the direction of informativeness of a data point. By selecting the query that has the largest directional derivative along the most uncertain direction in the parameter space, our algorithm makes the posterior concentrate quickly. For active learning tasks with binary responses, we observe that the notion of directional derivative is particularly meaningful and trivially easy to

[*]Equal contribution [1]Department of Electrical Engineering, KAIST, Daejeon, South Korea [2]Graduate School of AI, KAIST, Daejeon, South Korea [3]Seoul National University Business School, Seoul, South Korea. Correspondence to: Seungki Min <mskyt@snu.ac.kr>.

Accepted at ICML 2026 Workshop on Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning.

compute; see the discussion in Section 3.

Another clear contribution of our work is the theoretical analysis. We carefully analyze our algorithm in two stylized setups that are motivated by real-world applications. The analysis shows that our algorithm achieves the optimal *posterior concentration rate* specifically within the logistic and MIRT response models. Numerical experiments confirm these theoretical findings and highlight that DURM offers a clear advantage in computational speed and posterior concentration, while maintaining competitive predictive accuracy compared to computationally heavier baselines.

Related work. The problem we tackle can be considered a Bayesian active learning task (see (Settles, 2012) for a broader review). BALD (Houlsby et al., 2011; Gal et al., 2017), EIG (MacKay, 1992a;b), Mean STD (Kampffmeyer et al., 2016; Kendall et al., 2015), and the Var Ratio Test (Freeman, 1965) would be the work closest to ours, all of which are included as benchmark algorithms in our numerical experiments. Our proposed DURM algorithm shares an algorithmic principle similar to that of EIG and BALD, as it also aims to reduce posterior entropy in a greedy manner. However, they incorporate expensive acquisition functions, which may not be desirable in online user preference/ability learning tasks.

Existing Bayesian active learning algorithms have struggled to provide strong theoretical guarantees on convergence (Zhao et al., 2021a;b) or convergence rate (Gontier et al., 2022). (Zhao et al., 2021a;b) address these issues by reworking the Expected Loss Reduction used in Bayesian classification and proposing the weighted Mean Objective Cost of Uncertainty, which offers a convergence guarantee. Our work goes beyond, not only by providing convergence results, but also by proving the (asymptotic) optimality of our algorithm; see Section 4.

Another key feature of our algorithm is the use of directional derivatives. Although gradient information has frequently been used in the active learning literature (e.g., (Donmez & Carbonell, 2008; You et al., 2014)), to the best of our knowledge, our work is the first to combine directional derivatives with posterior distribution. Through this novel approach to query informativeness, our algorithm not only benefits from favorable theoretical analysis, but also achieves optimal posterior concentration rates strictly under our analyzed response models.

Our work is closely related to the bandit literature. Online learning of customer preferences has long been studied and formulated in the bandit literature (Agrawal et al., 2017; Faury et al., 2020), which motivated our model choice. Our theoretical analysis also relies on proof techniques developed in that literature, although bandit algorithms typically aim to minimize cumulative regret while our algorithm aims to reduce posterior entropy. Specifically, we refer to (Bassamboo et al., 2023) for convergence and asymptotic analysis of the adaptive query rule, and (Faury et al., 2020) for the time-decomposition of the negative Hessian matrix. Our analysis builds a connection between online learning and experimental design in the sense that our notion of optimality resembles the D-optimality in the experimental design literature and our suggested algorithm pursues E-optimality in a greedy fashion. Recently, a stream of work in the bandit literature also adopts concepts developed in the experimental design literature (Jamieson & Jain, 2018), but connections to Bayesian active learning have not been made yet.

2. Problem and Preliminaries

2.1. Problem Setup

We consider an online platform who is an active learner trying to identify a newly joined user's preference/ability by sequentially asking queries to the user. Let $\theta \in \Theta \subseteq \mathbb{R}^{d_\theta}$ be the parameter variable describing the user's preference/ability, and θ^* be its ground truth value that the learner aims to learn. We assume that the parameter set Θ is continuous. Let $(x_t)_{t\in\mathbb{N}}$ be the sequence of queries selected by the online platform. It takes values in a predefined query set $\mathcal{X} \subseteq \mathbb{R}^{d_x}$, which can be either discrete or continuous.[1]

Response models. The user's response to each query is either positive or negative, randomly determined according to a predefined response model. We describe the response model with the *likelihood function* $f : \mathcal{X} \times \Theta \to [0, 1]$, where $f(x, \theta)$ represents the likelihood of positive response when the query x is asked to the user with parameter θ. To be precise, let $y_t \in \{0, 1\}$ be the user's response to the query x_t at time step t. Then, $y_t \sim \text{Bern}(f(x_t, \theta^*))$, and y_t's are conditionally independent given the queries x_t's. We assume that this likelihood function $f(x, \theta)$ is twice-differentiable with respect to θ. The learner knows the function $f(\cdot, \cdot)$.

Adaptive query selection. In each time step, the online platform decides which query to ask and then receives the user's response immediately. The platform should make decisions based only on the observations obtained previously. More formally, let $\mathcal{D}_{t-1} := \left\{(x_s, y_s)\right\}_{s=1}^{t-1}$ be the collection of query-response pairs chosen/realized prior to time t, which is the information that the platform can utilize to select the tth query. A query selection algorithm π is expressed as a sequence of mappings, $(\pi_t)_{t\in\mathbb{N}}$, where each map $\pi_t : (\mathcal{X} \times \{0, 1\})^{t-1} \to \mathcal{X}$ specifies the rule to select the next query given the past observations, i.e.,

[1] In real-world environments, the query set can be effectively continuous if there are an enormous amount of queries available or if any query can be synthesized using generative models.

$x_t = \pi_t(\mathcal{D}_{t-1})$.

Bayesian framework. We adopt the Bayesian framework so that the online platform has a prior belief about the distribution of the user parameter θ^*. We denote the prior distribution by $p(\theta)$, a continuous probability distribution supported on Θ. According to the Bayes' rule, the posterior distribution can be derived as follows:

$$p(\theta|\mathcal{D}_{t-1}) \propto p(\theta) \times \prod_{s=1}^{t-1} f(x_s,\theta)^{y_s}\big(1-f(x_s,\theta)\big)^{1-y_s},$$

where $\mathcal{D}_{t-1} := \big\{(x_s,y_s)\big\}_{s=1}^{t-1}$. As a possibly simplest point estimator, we consider the *maximum a posteriori* (MAP) estimator, whose resulting estimate at time t is denoted by $\hat{\theta}_t$:

$$\hat{\theta}_t \in \arg\max_{\theta\in\Theta} \log p\big(\theta|\mathcal{D}_{t-1}\big). \tag{1}$$

Laplace's approximation. The posterior distribution is typically intractable. The Laplace's approximation suggests a simple heuristic in such a case. (Long, 2022), (Long et al., 2013) The posterior distribution is approximated with a multivariate Gaussian distribution $\mathcal{N}(\hat{\theta}_t, \Lambda_t^{-1})$, i.e.,

$$p(\theta|\mathcal{D}_{t-1}) \approx \frac{\det(\Lambda_t)^{1/2}}{(2\pi)^{d_\theta/2}} \exp\left(-\frac{1}{2}(\theta-\hat{\theta}_t)^\top \Lambda_t (\theta-\hat{\theta}_t)^\top\right),$$

where $\Lambda_t \in \mathbb{R}^{d_\theta\times d_\theta}$ is the *negative Hessian* matrix of the posterior log-likelihood:

$$\Lambda_t := -\nabla_\theta^2 \log p(\hat{\theta}_t|\mathcal{D}_{t-1}). \tag{2}$$

Our suggested algorithm's procedure and its underlying motivation heavily rely on this approximation.

2.2. Examples of response model

In our theoretical analysis and numerical experiments, we specifically consider two response models – (1) the logistic model, and (2) the MIRT model. We here introduce and motivate them.

2.2.1. Logistic response model

We first introduce a response model that mirrors the classic logistic regression model. Given a d-dimensional parameter space and a $(d+1)$-dimensional query space (i.e., $\Theta \subseteq \mathbb{R}^d$, $\mathcal{X} \subseteq \mathbb{R}^{d+1}$), the likelihood function is given by

$$f\big(\mathbf{x},\boldsymbol{\theta}\big) = \phi\left(x_{d+1} + \sum_{j=1}^{d} x_j \cdot \theta_j\right), \tag{3}$$

where $\phi(z) := 1/(1+e^{-z})$ is the logistic function, and x_i (respectively, θ_i) represents the ith component of the vector $\mathbf{x}$ (respectively, $\boldsymbol{\theta}$). The bold letters are used just to clarify that they are vectors.

The next example motivates this response model.

Example 1 (Customer preference learning) Imagine an online fashion retail store in which thousands of fashion items are being sold. For a newly joined customer, the store tries to identify the customer's preference through a series of "this-or-that" questions: in each question, the store displays two items and lets the customer pick more preferred one.

Let us adopt the following random utility model which is commonly used as a choice model. The value that a customer perceives for item i is given by $u_i = \boldsymbol{\theta}^\top \mathbf{v}_i - p_i + \varepsilon_i$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ is the customer's latent preference parameter vector, $\mathbf{v}_i \in \mathbb{R}^d$ is the item's known feature vector, $p_i \in \mathbb{R}_+$ is the item's price, and ε_i is the random perturbation drawn from the standard Gumbel distribution. When an item pair (i,k) is presented to the customer, the probability that the customer prefers item i over item k can be represented as

$$\begin{aligned} p(u_i > u_k) &= \phi\left((\boldsymbol{\theta}^\top \mathbf{v}_i - p_i) - (\boldsymbol{\theta}^\top \mathbf{v}_k - p_k)\right) \\ &= \phi\left(\boldsymbol{\theta}^\top(\mathbf{v}_i - \mathbf{v}_k) + (p_k - p_i)\right). \end{aligned}$$

This situation can be formulated with the above logistic model (3) with a query set $\mathcal{X} := \{(\mathbf{v}_i - \mathbf{v}_k, p_k - p_i)\}_{i\neq k}$.

2.2.2. MIRT response model

Following the multi-dimensional item response theory (MIRT; (Bock & Gibbons, 2021)), we introduce the MIRT response model whose likelihood function is given by

$$f(\mathbf{x},\boldsymbol{\theta}) = \prod_{j=1}^{d} \phi\big(x_j + \theta_j\big), \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} \in \mathbb{R}^d$, and $\phi(z) := 1/(1+e^{-z})$ is the logistic function. We motivate this model through the following example.

Example 2 (Ability testing) Imagine an online math education platform who conducts level tests for newly joined student users. A student's mathematical ability is represented with a d-dimension vector $\boldsymbol{\theta} \in \mathbb{R}^d$, where each component $\theta_j \in \mathbb{R}$ represents the skill level in a particular aspect such as geometry, algebra, probability, etc. The platform owns a pool of math questions, each of which is also represented as a d-dimensional vector $\mathbf{y} \in \mathbb{R}^d$. Here, each component $y_j \in \mathbb{R}$ represents the difficulty of the question in each aspect.

We assume that the difficulty vector serves as a component-wise "soft threshold". Given a student with ability vector $\boldsymbol{\theta}$ and a question with difficulty vector $\mathbf{y}$, the probability that the student correctly figures out the jth aspect of the question is given by $\phi(\theta_j - y_j)$. Further assuming that each ability/difficulty dimension contributes to the final outcome independently, the student gives the correct answer with

probability $\prod_{j=1}^{d} \phi(\theta_j - y_j)$. One can alternatively interpret this model as follows: the student solves the question correctly if and only if his/her randomly perturbed ability level exceeds the question's difficulty level in all aspects (i.e., $\theta_j + \varepsilon_j > y_j$ for all j where ε_j's are drawn from the standard Gumbel distribution independently).

This situation can be formulated with the above MIRT response model (4) with a query set $\mathcal{X} := \{-\mathbf{y} : \mathbf{y} \in \mathcal{Y}\}$, where $\mathcal{Y}$ is the set of difficulty vectors of all available questions.

3. Algorithm

We now propose *Directional Uncertainty Reduction Maximization* (DURM) algorithm, which offers a simple adaptive query selection rule that effectively reduces the learner's uncertainty in parameter estimation. The main idea is straightforward: in each query selection step, the algorithm (1) finds the *most uncertain direction* in the parameter space, and then (2) selects the query that *maximizes the reduction* in the uncertainty along this direction. This procedure is implemented in Algorithm 1 below.

Algorithm 1 Directional Uncertainty Reduction Maximization (DURM) Algorithm

Initialize: $\mathcal{D}_0 \leftarrow \emptyset$
for $t = 0, 1, 2, \ldots$ **do**
  Compute the MAP estimate along with the negative Hessian matrix:

$$\hat{\theta}_t \leftarrow \arg\max_{\theta \in \Theta} p(\theta|\mathcal{D}_{t-1}), \;\; \Lambda_t \leftarrow -\nabla_\theta^2 \log p(\hat{\theta}_t|\mathcal{D}_{t-1}).$$

  Find the query at which the likelihood gradient becomes most aligned with the eigenvector of Λ_t corresponding to the smallest eigenvalue: $u_t \leftarrow v_{\min}(\Lambda_t)$,

$$x_t \leftarrow \arg\max_{x \in \mathcal{X}} \left| \nabla_\theta f(x, \hat{\theta}_t)^\top u_t \right|. \tag{5}$$

  Observe $y_t \sim \text{Bernoulli}(f(x_t, \theta^*))$ and update $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(x_t, y_t)\}$.
end for

Most uncertain direction. To describe more formally, consider the moment at which the active learner is determining the tth query given the past query-response pairs, $\mathcal{D}_{t-1} := \{(x_s, y_s)\}_{s=1}^{t-1}$. Let $\mathcal{L}_t(\theta) := \log p(\theta|\mathcal{D}_{t-1})$ be the log-likelihood function of the posterior distribution. The second order Taylor expansion at the peak $\hat{\theta}_t$ (which is the MAP estimate) gives

$$\mathcal{L}_t(\hat{\theta}_t + u) \approx \mathcal{L}_t(\hat{\theta}_t) + \frac{1}{2} u^\top \nabla^2 \mathcal{L}_t(\hat{\theta}_t) u. \tag{6}$$

With the negative Hessian matrix $\Lambda_t := -\nabla^2 \mathcal{L}_t(\hat{\theta}_t)$ which is a positive semi-definite matrix, the direction $u_t \in \mathbb{R}^{d_\theta}$ that (locally) maximizes the log-likelihood value around $\hat{\theta}_t$ can be found by[2]

$$\begin{aligned} u_t &:= \arg\max_{u \in \mathbb{R}^{d_\theta} : \|u\|_2 = 1} \left\{ \mathcal{L}_t(\hat{\theta}_t) + \frac{1}{2} u^\top \nabla^2 \mathcal{L}_t(\hat{\theta}_t) u \right\} \\ &= \arg\min_{u \in \mathbb{R}^{d_\theta} : \|u\|_2 = 1} \left\{ u^\top \Lambda_t u \right\} = v_{\min}(\Lambda_t). \end{aligned} \tag{7}$$

Here, $v_{\min}(\Lambda_t)$ indicates the normalized eigenvector associated with the smallest eigenvalue of the matrix Λ_t.

We refer to the direction u_t (7) as the *most uncertain direction* at time step t. It is the direction along which the learner is least confident in terms of parameter estimation. Let $\epsilon_t := \hat{\theta}_t - \theta^*$ be the estimation error vector. The approximation (6) is simply the Laplace's approximation saying that $\epsilon_t | \mathcal{D}_{t-1} \sim \mathcal{N}(0, \Lambda_t^{-1})$. The expected amount of estimation error along a particular direction $u \in \mathbb{R}^{d_\theta}$ can be represented as $\text{Var}(\epsilon_t^\top u | \mathcal{D}_{t-1}) \approx u^\top \Lambda_t^{-1} u$, which is maximized at the direction u_t.

Query selection. Given the most uncertain direction u_t, the algorithm aims to select the query that is most informative to reduce the uncertainty along this direction. More specifically, consider two parameter vectors, $\hat{\theta}_t + \delta u_t$ and $\hat{\theta}_t - \delta u_t$, for some small real number $\delta \in \mathbb{R}_+$. This pair would be the one that is most difficult for the learner to determine which one among two is closer to the true parameter θ^* based on his current knowledge. The amount of *disagreement* between the two parameter vectors for the same query x can be measured by the gap between their corresponding likelihood values, which admits the following approximation:

$$|f(x, \hat{\theta}_t + \delta u_t) - f(x, \hat{\theta}_t - \delta u_t)| \approx 2\delta |\nabla_\theta f(x, \hat{\theta}_t)^\top u_t|.$$

Although this disagreement compares the likelihoods of positive response, it remains identical even if we compare the likelihoods of negative response, because $|\nabla_\theta f^\top u| = |\nabla_\theta (1 - f)^\top u|$.

Among all possible queries, the query that maximizes this disagreement is chosen:

$$x_t = \arg\max_{x \in \mathcal{X}} \left| \nabla_\theta f(x, \hat{\theta}_t)^\top u_t \right|. \tag{8}$$

[2] We implicitly assume that the direction u_t is a feasible direction at $\hat{\theta}_t$. That is, there exists a sequence $(\theta_k)_{k \in \mathbb{N}} \subset \Theta$ with $\theta_k \to \hat{\theta}_t$ and $(\theta_k - \hat{\theta}_t)/\|\theta_k - \hat{\theta}_t\| \to u_t$ as $k \to \infty$. If $\hat{\theta}_t$ is not an interior point of Θ, however, the direction u_t may not be feasible. In such a case, we can consider the set of all feasible directions (often referred to as tangent cone) and add it as a constraint in the minimization (7).

The chosen query maximally differentiates two parameter candidates, $\hat{\theta}_t + \delta u_t$ and $\hat{\theta}_t - \delta u_t$.

One can alternatively motivate the objective (8) as the *expected length of the directional derivative* of the log-likelihood. For a given query x, the learner would expect that the positive response occurs with probability $f(x, \hat{\theta}_t)$. Letting $\mathcal{L}(\theta, x, y) := \log\left(f(x,\theta)^y (1 - f(x,\theta))^{1-y}\right)$ be the log-likelihood value corresponding to a query-response pair (x, y), the expected amount of disagreement in the log-likelihood along the direction u_t is given by

$$\mathbb{E}\left[\left|\nabla_\theta \mathcal{L}(\theta, x_t, y_t)^\top u_t\right| \,|\mathcal{D}_{t-1}, x_t = x\right] \approx 2\left|\nabla_\theta f(x,\theta)^\top u_t\right|$$

Interpreting the directional derivative of the log-likelihood as the directional informativeness of an observation, we view the query chosen by the DURM algorithm as the one that maximizes the expected amount of information toward the most uncertain direction.

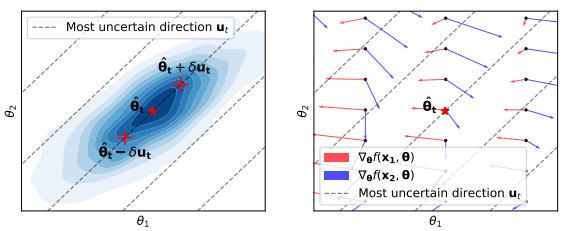


Figure 1: Illustration of DURM's query selection rule. (Left) Given the posterior distribution of the parameter, the algorithm finds the most uncertain direction $\mathbf{u}_t$ at the current estimate $\hat{\boldsymbol{\theta}}_t$, visualized with dashed lines. (Right) The gradient of the likelihood, visualized with colored arrows, quantifies each query's multi-dimensional informativeness. The algorithm chooses the query whose gradient at the current estimate is most aligned with the direction $\mathbf{u}_t$.

4. Analysis

In this section, we provide theoretical analysis of our suggested algorithm for the two response models introduced in Section 2. All the proofs are deferred to the Appendix. We begin by introducing the performance metric of our major interest.

Definition 4.1. Given an algorithm π, we define the *posterior concentration rate* of π up to time t as

$$H_t^\pi := \frac{(\det \Lambda_t)^{\frac{1}{d_\theta}}}{t},$$

where Λ_t is the negative Hessian of posterior log-likelihood evaluated at the MAP estimate, i.e., $\Lambda_t := -\nabla_\theta^2 \log p(\hat{\theta}_t | \mathcal{D}_{t-1})$ where $\hat{\theta}_t := \arg\max_{\theta \in \Theta} \log p(\theta | \mathcal{D}_{t-1})$.

The posterior concentration rate H_t^π quantifies how quickly the total amount of estimation uncertainty reduces over time. Recall that the Laplace's approximation postulates that $\theta^* | \mathcal{D}_{t-1} \sim \mathcal{N}(\hat{\theta}_t, \Lambda_t^{-1})$. Following the relationship between the (differential) entropy of a Gaussian and the log-determinant of its covariance matrix, the entropy of the posterior distribution can be approximated as $\mathbb{H}(\theta^* | \mathcal{D}_{t-1}) \approx \frac{d_\theta}{2}\log(2\pi e) + \frac{1}{2}\log\det(\Lambda_t^{-1}) = \Theta(-\log(t \cdot H_t))$ (Clarke, 1999). Therefore, if H_t is large, the entropy decays quickly, suggesting that the algorithm quickly identifies the true parameter.

Presumably, the posterior concentration rate H_t^π converges to some constant (which is shown, in fact, for our algorithm suggested in specific setups; see theorems below). The asymptotic normality of the MAP estimator (Lehmann & Casella, 2006) implies that $(\theta^* - \hat{\theta}_t)/\sqrt{t} \sim \mathcal{N}(0, \mathcal{I}^{-1})$, where $\mathcal{I}$ is the Fisher information matrix. Consequently, we can argue that $\Lambda_t \approx t \cdot \mathcal{I}^{-1}$, and therefore the posterior concentration rate H_t^π converges to $(\det \mathcal{I})^{-1/d_\theta}$. Maximizing the posterior concentration rate is effectively equivalent to minimizing the determinant of the Fisher information matrix, which is also known as D-optimality in the optimal experimental design domain.

4.1. Logistic response model

In this section, we investigate the logistic response model introduced in Section 2.2.1. Motivated from Section 2.2.1, we consider a *cylindrical and continuous query set* $\mathcal{X} := \{(x^{\text{dir}}, x^{\text{offset}}) \in \mathbb{R}^d \times \mathbb{R} : \|x^{\text{dir}}\|_2 \le \alpha\}$. Recall that, given a parameter $\theta \in \Theta = \mathbb{R}^d$ and a query $x = (x^{\text{dir}}, x^{\text{offset}}) \in \mathcal{X}$, the likelihood is given by $f(x, \theta) = \phi\left({x^{\text{dir}}}^\top \theta + x^{\text{offset}}\right)$, $\phi(z) := 1/(1 + e^{-z})$.

Next propositions characterize the query chosen by the DURM algorithm.

Proposition 4.2. *The query $x_t = (x_t^{dir}, x_t^{offset})$ chosen by the DURM algorithm at time t satisfies that*

$$x_t^{dir} = \alpha \cdot u_t, \qquad f(x_t, \hat{\theta}_t) = \frac{1}{2}.$$

This result shows that the query selection procedure (5) yields the direction component x_t^{dir} that is most aligned with the most uncertain direction u_t, and yields the offset component x_t^{offset} at which the positive outcome and the negative outcome are equally likely with respect to the current MAP estimate. This choice of query is particularly natural considering the situation of Section 2.2.1 – when the platform is uncertain about the customer's preference on the feature u_t, the "this-or-that" question to be asked consists of two items that maximally disagree on the feature u_t and their prices are balanced so that the two items are equally likely to be chosen by the customer.

Next proposition further shows that this query is myopically optimal in terms of reduction in the log-determinant of

negative Hessian matrix which is understood as a proxy for the total amount of uncertainty.

Proposition 4.3. *Let $\Lambda(\theta; \mathcal{D}) := -\nabla^2_\theta \log p(\theta|\mathcal{D})$ be the negative Hessian of the posterior likelihood evaluated at θ given the observation set $\mathcal{D}$. Then, the query x_t chosen by the DURM algorithm at time t satisfies*

$$x_t \in \arg\max_{x \in \mathcal{X}} \log\det \Lambda\big(\hat{\theta}_t; \mathcal{D}_{t-1} \cup \{(x, y)\}\big),$$

regardless of $y \in \{0, 1\}$. The query minimizes the log-determinant of the negative Hessian updated with the next observation but evaluated at the current MAP estimate $\hat{\theta}_t$.

This result implies that if the MAP estimate does not change too much (i.e., $\hat{\theta}_{t+1} \approx \hat{\theta}_t$), then the query indeed minimizes the log-determinant of the next negative Hessian (evaluated at the next MAP estimate $\hat{\theta}_{t+1}$). We then observe that there is a time-invariant structure in the negative Hessian being preserved under our algorithm if a Gaussian prior is adopted:

Proposition 4.4. *Suppose that a Gaussian prior is adopted, i.e., $\theta^* \sim \mathcal{N}(\theta_0, \Lambda_0^{-1})$ so that $\hat{\theta}_1 = \theta_0$ and $-\nabla^2_\theta \log p(\hat{\theta}_1) = \Lambda_0$. Let $\mathcal{V}_0 = (v_{0,1}, \ldots, v_{0,d})$ be the (ordered) set of orthonormal eigenvectors of Λ_0, and $(\lambda_{0,1}, \ldots, \lambda_{0,d})$ be their corresponding eigenvalues. Under the DURM algorithm,[3] the negative Hessian admits the following eigendecomposition:*

$$\Lambda_t = \sum_{j=1}^{d} \lambda_{t,j} v_{0,j} v_{0,j}^\top, \lambda_{t,j} = \lambda_{0,j} + \alpha^2 \sum_{s<t: j_s = j} g(x_s, \hat{\theta}_t),$$

where $g(x, \theta) := f(x, \theta)(1 - f(x, \theta))$ and $j_s \in \arg\min_{j \in [d]} \lambda_{s,j}$. The query $x_t = (x_t^{dir}, x_t^{offset})$ chosen by the algorithm satisfies that $x_t^{dir} = \alpha v_{0,j_t}$.

Based on this result, next Theorem establishes a performance guarantee on the posterior concentrate rate that the DURM algorithm achieves.

Theorem 4.5. *Suppose that a Gaussian prior is adopted. Under the DURM algorithm, the posterior concentrates at an asymptotic rate of $\alpha^2/(4d)$:*

$$\lim_{t \to \infty} H_t^{\pi_{DURM}} = \frac{\alpha^2}{4d}, \quad \textit{almost surely}.$$

Finally, we conclude this section by showing that DURM achieves the theoretical upper bound of the posterior concentration rate, highlighting that the algorithm is indeed optimal under the linear logistic response model.

Theorem 4.6. *The asymptotic posterior concentration rate cannot exceed $\alpha^2/(4d)$ under any algorithm: for any π,*

$$\limsup_{t \to \infty} H_t^\pi \le \frac{\alpha^2}{4d}, \quad \textit{almost surely}.$$

[3] To be precise, we consider the algorithm that utilizes the eigendecomposition (4.4) when determining $v_{\min}(\Lambda_t)$.

4.2. MIRT response model

We next investigate the MIRT response model, introduced in Section 2.2.2. Specifically, we consider the parameter set $\Theta = \mathbb{R}^d$ and the query set $\mathcal{X} = \bar{\mathbb{R}}^d$ where $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ is the extended real line. Recall that the likelihood is given by $f(x, \theta) = \prod_{j=1}^d \phi(x_j + \theta_j)$. This choice of query set allows the active learner to ignore particular axes – i.e., choosing $x_j = \infty$ gives $\phi(x_j + \theta_j) = 1$ regardless of the value of θ_j, effectively removing the jth component from our consideration.

Proposition 4.7. *Let x_t be the query chosen by the DURM algorithm at step t. It satisfies*

$$\left(\frac{d}{d+1}\right)^d \le f(x_t, \hat{\theta}_t) \le \frac{1}{2}.$$

Note that $(d/(d+1))^d \ge 1/e \approx 37\%$. In the context of Section 2.2.2, this result shows that the online education platform will ask a question such that the student will give the correct answer with probability between 37% and 50%. The question will not be too trivial nor too difficult.

Analogously to Proposition 4.4, next Proposition identifies a time-invariant structure in the negative Hessian matrix when an component-wise independent Gaussian prior is adopted.

Proposition 4.8. *Suppose that a Gaussian prior with a diagonal covariance matrix is adopted, i.e., $\theta^* \sim \mathcal{N}(\theta_0, \Lambda_0^{-1})$ with $\Lambda_0 = \mathrm{diag}((\lambda_{0,j})_{j=1}^d)$ and query space uses extended real number, Under the DURM algorithm, the negative Hessian admits the following eigen decomposition:*

$$\Lambda_t = \mathrm{diag}\left((\lambda_{t,j})_{j=1}^d\right), \lambda_{t,j} = \lambda_{0,j} + \sum_{s<t: j_s = j} g(x_s, \hat{\theta}_t),$$

on $g(x, \theta) := f(x, \theta)(1 - f(x, \theta))$ and $j_s \in \arg\min_{j \in [d]} \lambda_{s,j}$.

We next present the results analogous to Theorem 4.5 and Theorem 4.6.

Theorem 4.9. *Suppose that a Gaussian prior with a diagonal covariance matrix is adopted. Under the DURM algorithm, the posterior concentrates at an asymptotic rate of $1/(4d)$: $\lim_{t \to \infty} H_t^{\pi_{DURM}} = 1/(4d)$, almost surely.*

Theorem 4.10. *The asymptotic posterior concentration rate cannot exceed $1/(4d)$ under any algorithm that is strongly consistent: i.e., for any π that guarantees $\hat{\theta}_t \overset{a.s.}{\to} \theta^*$, we have $\limsup_{t \to \infty} H_t^\pi \le 1/(4d)$, almost surely.*

The proofs of above two Theorems basically mirror the ones used to prove the results for the logistic model. Combining above two results, we conclude that the DURM algorithm achieves the best possible performance for the MIRT response model.

5. Numerical Experiments

In this section, we provide simulation results obtained with synthetic data and real-world data, for two response models. All experiments were conducted in an environment with Intel(R) Core(TM) i7-14700K, 3.40 GHz, 32GB RAM.

5.1. Synthetic data

We consider the configurations described in Section 4.1 for the linear response model and Section 4.2 for the logistic response model, with the choice of $d = 4$, $\alpha = 10$, $\theta^* = 0$, and a Gaussian prior with mean vector $(0.9, 0.3, 0.3, 0.1)$ and covariance matrix whose eigenvalues are $7.1, 5.5, 4.83, 4.1$.

Figure 2 reports the performance of the DURM algorithm measured across 250 runs of simulation. In both logistic and MIRT models, we observe that the log-determinant of negative Hessian, $\log\det\Lambda_t$, sharply matches its theoretical prediction, confirming Theorem 4.5 and Theorem 4.9. Furthermore, the estimation error, $\|\hat{\theta}_t - \theta^*\|_2^2$, quickly decays and also sharply matches its theoretical limit following the Cramer-Rao bound.

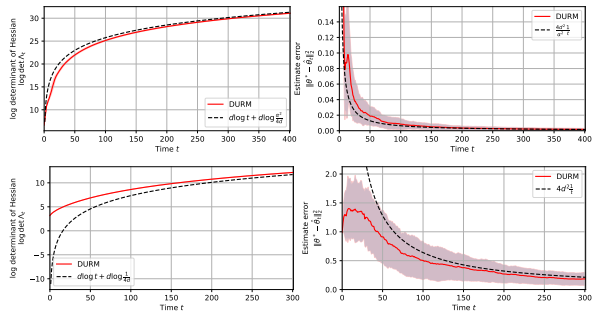


Figure 2: Simulation results obtained with synthetic data for logistic response model (upper) and MIRT response model (lower).

We also tested various experimental configurations and observed consistent results. See Appendix B.1 in the Supplementary Materials for details.

5.2. Real-world data

We used a part of *i-scream edu dataset* (IscreamEDU, 2020), provided by an online math education platform. The dataset for seventh grade is used, containing 364,038 samples of interactions between 388 students and 1,322 questions.

With the choice of $d = 4$, the linear model and the MIRT model are fitted separately to the data so that 388 student feature vectors and 1,322 question feature vectors are estimated. Using the sample mean/covariance of the estimated student feature vectors, a Gaussian prior is obtained and used for all algorithms. The algorithms are tested using randomly chosen 38 students as if these students had just joined the platform. For each student being tested, only the questions that his/her actual responses are available in the dataset are allowed to ask and the same question cannot be asked again. We let the algorithms ask 200 different adaptive questions and observe the actual responses.

As performance metrics, we have measured (1) the log-determinant of the negative Hessian as a measure of the posterior concentration level (higher is better), (2) the estimation error $\|\hat{\theta}_t - \theta^*\|$, (3) the accuracy of online prediction over 200 questions, and (4) the computation time required to select 200 questions. When measuring the prediction accuracy, the MAP estimate in each period is used to predict the outcome of the next query with a pre-tuned threshold.

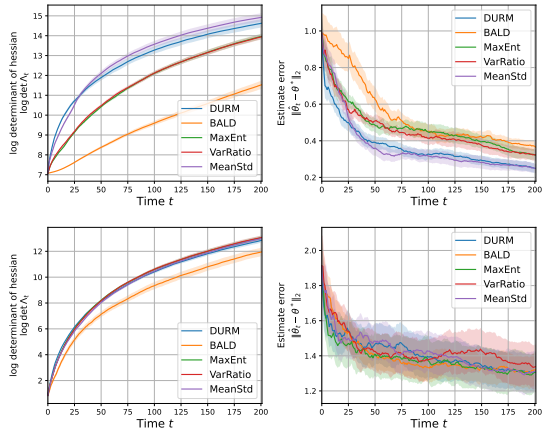


Figure 3: Simulation results obtained with *i-scream edu 7-th grade dataset* for logistic response model (left) and MIRT response model (right).

Table 1: Accuracy and Runtime comparison of algorithms (Logistic vs. MIRT response model).

Algorithm	Logistic		MIRT	
	Acc (%)	Time (s)	Acc (%)	Time (s)
DURM	72.9±0.8	**4.4±0.7**	71.1±0.1	**21.1±1.9**
BALD	72.5±0.9	33.4±5.3	70.4±0.1	170.6±66.8
MaxEnt	73.4±0.8	28.3±2.4	71.1±0.1	146.8±6.3
VarRatio	73.4±0.8	28.2±2.2	71.1±0.1	146.9±6.1
MeanStd	**73.0±0.8**	29.8±3.1	**72.5±0.1**	148.4±6.5

Table 1 and Figure 3 show the performance of DURM (ours) compared to baseline methods. While DURM does not always yield the strictly highest predictive accuracy, it remains highly competitive. More importantly, DURM demonstrates a clear advantage in computational efficiency, executing significantly faster than entropy-based algorithms. Furthermore, Figure 3 illustrates its strong capability in rapidly increasing the posterior concentration (measured by the log-determinant of the negative Hessian).

We additionally considered a setup with a flattened prior whose covariance matrix is scaled by 0.01. This setup pos-

tulates the situation in which the learner has a very weak belief about the users, or the user pool is very diverse.

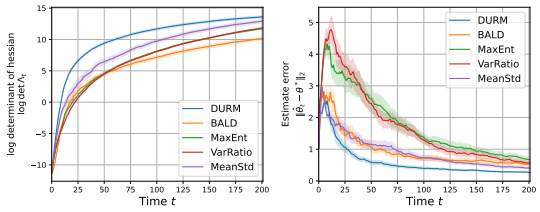


Figure 4: Simulation results obtained with *i-scream edu 7-th grade dataset* for logistic response model with the flattened prior.

As shown in Figure 4, the DURM algorithm maintains robust posterior concentration and distinct computational efficiency particularly in this setup. The sampling-based calculations of posterior entropy become noisy when the prior is flat and, therefore, the acquisition functions used in the other algorithms become difficult to evaluate accurately. In contrast, the DURM algorithm relies on the Laplace's approximation, which is relatively less sensitive to the sampling-based approximation, resulting in a better performance.

Moreover, we tested the 1st, 3rd, and 9th grades datasets and also the non-Gaussian prior settings. And the DURM algorithm achieved the near-best performance (see Appendix B.2 in the Supplementary Materials).

6. Conclusion

In this paper, we have suggested an effective algorithm for adaptive testing of user preference/ability, and provided theoretical analyses and numerical experiments that highlight its effectiveness. The algorithm provides a novel perspective on the use of directional derivatives to active learning tasks. However, there are several limitations in our work:

First, our algorithm heavily relies on a local approximation of the posterior distribution. Although it leads to computational efficiency as well as theoretical tractability and our experiments show that it is a sufficiently good approximation in our application, its expressiveness may fall short compared to more advanced posterior approximation techniques such as variational Bayesian method. Once the most uncertain direction can be properly defined and identified, we believe that the idea of directional derivatives will remain effective.

Second, our analysis is yet limited to specific response models and priors. We gently anticipate that our algorithm will achieve optimality as long as the E-optimal design coincides with the D-optimal design under the assumed model and the posterior concentration is sufficiently orthogonal. Unfortunately, we find it extremely challenging to establish explicit conditions for this argument, highlighting the need for further investigation.

Impact Statement

This paper contributes to the advancement of Bayesian active learning by proposing an efficient adaptive query selection method based on directional derivatives. Our work primarily aims to improve computational efficiency and theoretical guarantees in active learning. We do not foresee any specific ethical concerns or societal risks arising from this research.

References

Agrawal, S., Avadhanula, V., Goyal, V., and Zeevi, A. J. Thompson sampling for the mnl-bandit. In *Annual Conference Computational Learning Theory*, 2017.

AndroidPolice. Youtube stops showing recommendations when you're not logged in, 2024. AndroidPolice Article.

Bassamboo, A., Deep, V., Juneja, S., and Zeevi, A. Learning to ask the right questions: A multi-armed bandits approach. *Available at SSRN*, 01 2023.

Bock, R. D. and Gibbons, R. D. *Item response theory*. John Wiley & Sons, 2021.

Clarke, B. Asymptotic normality of the posterior in relative entropy. *IEEE Transactions on Information Theory*, 45 (1):165–176, 1999.

Donmez, P. and Carbonell, J. Optimizing estimated loss reduction for active sampling in rank learning. In *International Conference on Machine Learning*, pp. 248–255, 01 2008.

Faury, L., Abeille, M., Calauzènes, C., and Fercoq, O. Improved optimistic algorithms for logistic bandits. In *International Conference on Machine Learning*, pp. 3052–3060. PMLR, 2020.

Freeman, L. G. *Elementary Applied Statistics for students in Behavioral Science*. 1965.

Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1183–1192, 2017.

Gontier, C., Surace, S. C., Delvendahl, I., Müller, M., and Pfister, J.-P. Efficient sampling-based bayesian active learning for synaptic characterization. *PLOS Computational Biology*, 19, 2022.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

IscreamEDU, 2020. AIHub Dataset.

Jamieson, K. G. and Jain, L. A bandit approach to sequential experimental design with false discovery control. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Kampffmeyer, M., Salberg, A.-B., and Jenssen, R. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1–9, 2016.

Kendall, A., Badrinarayanan, V., and Cipolla, R. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint*, 2015.

Lehmann, E. L. and Casella, G. *Theory of point estimation*. Springer Science & Business Media, 2006.

Long, Q. Multimodal information gain in bayesian design of experiments. *Computational Statistics*, 37(2):865–885, 2022.

Long, Q., Scavino, M., Tempone, R., and Wang, S. Fast estimation of expected information gains for bayesian experimental designs based on laplace approximations. *Computer Methods in Applied Mechanics and Engineering*, 259:24–39, 2013.

MacKay, D. J. C. Information-based objective functions for active data selection. *Neural Computation*, 4:590–604, 1992a.

MacKay, D. J. C. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992b.

Settles, B. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6, 06 2012.

You, X., Wang, R., and Tao, D. Diverse expected gradient active learning for relative attributes. *IEEE Transactions on Image Processing*, 23:3203–3217, 2014.

Zhao, G., Dougherty, E. R., Yoon, B.-J., Alexander, F. J., and Qian, X. Bayesian active learning by soft mean objective cost of uncertainty. In *International Conference on Artificial Intelligence and Statistics*, 2021a.

Zhao, G., Dougherty, E. R., Yoon, B.-J., Alexander, F. J., and Qian, X. Uncertainty-aware active learning for optimal bayesian classifier. In *International Conference on Learning Representations*, 2021b.

A. Omitted Proofs

A.1. Proofs for Section 4.1

Proof of Proposition 4.2.

Proof. As the gradient of f is given as $\nabla_\theta f(x, \hat{\theta}_t) = x^{\text{dir}} f(x, \hat{\theta}_t)\big(1 - f(x, \hat{\theta}_t)\big)$ with the linear model, the objective in (5) is given as

$$\left|\nabla_\theta f(x, \hat{\theta}_t)^\top v_{\min}(\Lambda_t)\right| = v_{\min}(\Lambda_t)^\top x^{\text{dir}} f(x, \hat{\theta}_t)\big(1 - f(x, \hat{\theta}_t)\big). \tag{9}$$

Here, note that $x^{\text{dir},\top} v_{\min}(\Lambda_t) \leq \alpha$ and $f(x, \hat{\theta}_t)\big(1 - f(x, \hat{\theta}_t)\big) \leq \frac{1}{4}$, hence $\left|\nabla_\theta f(x, \hat{\theta}_t)^\top v_{\min}(\Lambda_t)\right| \leq \frac{1}{4}\alpha$. As $x_t = \big(\alpha v_{\min}(\Lambda_t), -\alpha v_{\min}(\Lambda_t)^\top \hat{\theta}_t\big)$ achieves this maximum value, the DURM algorithm selects the query such that $x_t^{\text{dir}} = \alpha v_{\min}(\Lambda_t)$ and $f(x_t, \hat{\theta}_t) = \frac{1}{2}$. □

Proof of Proposition 4.3.

Proof. Since the log-likelihood is given by

$$\log p(\theta \mid x_t, y_t) = y_t \log f(x_t, \theta) + (1 - y_t)\log(1 - f(x_t, \theta)), \tag{10}$$

the gradient and Hessian of log-likelihood is given as

$$\nabla_\theta \log p(\theta \mid x_t, y_t) = x_t^{\text{dir}}\big(y_t - f(x_t, \theta)\big); \tag{11}$$

$$\nabla_\theta^2 \log p(\theta \mid x_t, y_t) = -x_t^{\text{dir}} x_t^{\text{dir}\top}\big(f(x_t, \theta)(1 - f(x_t, \theta))\big). \tag{12}$$

The negative Hessian matrix of the log-likelihood is

$$\begin{aligned}
\Lambda(\theta; \mathcal{D}_t) &= -\nabla_\theta^2 \log p(\theta \mid \mathcal{D}_t) && (13)\\
&= -\nabla_\theta^2\big(\log p(\theta \mid \mathcal{D}_{t-1}) + \log p(\theta \mid x_t, y_t)\big) && (14)\\
&= \Lambda(\theta; \mathcal{D}_{t-1}) + x_t^{\text{dir}} x_t^{\text{dir}\top} g(x_t, \theta), && (15)
\end{aligned}$$

where $g(x, \theta) = f(x, \theta)\big(1 - f(x, \theta)\big)$.

Let's denote $\Lambda_t = \Lambda(\hat{\theta}_t; \mathcal{D}_{t-1})$, then, by the matrix determinant lemma,

$$\det \Lambda(\hat{\theta}_t; \mathcal{D}_{t-1} \cup \{(x, y)\}) = \big(1 + g(x_t, \theta) x_t^{\text{dir}\top} \Lambda_t^{-1} x_t^{\text{dir}}\big) \det \Lambda_t. \tag{16}$$

As $\det \Lambda_t$ is fixed, by Proposition 4.2, x_t chosen by the DURM algorithm maximizes value of $\log\det \Lambda(\hat{\theta}_t; \mathcal{D}_{t-1} \cup \{(x_t, y_t)\})$. □

Proof of Proposition 4.4.

Proof. Suppose that regardless of θ, the negative Hessian matrix $\Lambda(\theta; \mathcal{D}_{t-1})$ has the same orthonormal eigenvectors. Let $\mathcal{V}_t = (v_{t,1}, \ldots, v_{t,d})$ be the ordered set of orthonormal eigenvectors of $\Lambda(\theta; \mathcal{D}_{t-1})$. Hence, the negative Hessian matrix at time t can be decomposed as

$$\Lambda(\theta; \mathcal{D}_{t-1}) = \sum_{j=1}^{d} \lambda(\theta)_{t,j} v_{t,j} \tag{17}$$

where $\lambda(\theta)_{t,j}$ denotes the corresponding eigenvalue of eigenvector $v_{t,j}$ evaluated at θ.

Since $\Lambda_{t+1} = \Lambda(\hat{\theta}_{t+1}; \mathcal{D}_t)$, also $v_{\min}(\Lambda_t)$ is in $\mathcal{V}_t$, we can denote as $v_{\min}(\Lambda_t) = v_{t,j_t}$, i.e., $j_t \in \arg\min_{j \in [d]} \lambda_{t,j}(\hat{\theta}_t)$. By Proposition 4.2, $x_t^{\text{dir}} = \alpha v_{t,j_t}$, hence we get $\Lambda(\theta; \mathcal{D}_t) = \Lambda(\theta; \mathcal{D}_{t-1}) + \alpha^2 g(x_t, \theta) v_{t,j_t} v_{t,j_t}^\top$ from (15). Therefore, $\mathcal{V}_t$ provides a proper eigen decomposition of $\Lambda(\theta; \mathcal{D}_t)$ again, regardless of θ.

By choosing $v_{\min}(\Lambda_t)$ in this manner, the set of eigenvectors of the Λ_t, $\mathcal{V}_t$, is identical to $\mathcal{V}_0$ for all steps t, i.e., the set of eigenvectors does not change over time. As the initial Hessian $\Lambda(\theta)$ is constant by the condition, $\lambda_{0,j}$ is also a constant. Therefore, Λ_t can be expressed as $\Lambda_t = \sum_{j=1}^{d} \lambda_{t,j} v_{0,j} v_{0,j}^\top$, where $\lambda_{t,j} = \lambda_{0,j} + \alpha^2 \sum_{s<t:j_s=j} g(x_s, \hat{\theta}_t)$. This is because $\lambda_{t,j}$ is always evaluated at $\hat{\theta}_t$. □

Proposition A.1. *Suppose that a Gaussian prior is adopted. Under the DURM algorithm, the MAP estimator $\hat{\theta}_t$ converges almost surely to the true parameter θ^*:*

$$\lim_{t\to\infty} \hat{\theta}_t = \theta^*, \quad \textit{almost surely}.$$

This Proposition A.1 is needed to show Proposition A.2.

Proof of Proposition A.1.

Proof. For any algorithm, the sequence $\{y_s - f(x_s, \theta^*)\}_{s=1}^{t}$ is a martingale difference sequence (MDS) as

$$\mathbb{E}\big[y_s - f(x_s, \theta^*) \mid \mathcal{F}_{s-1}\big] = \big(1 - f(x_s, \theta^*)\big) f(x_s, \theta^*) - f(x_s, \theta^*)\big(1 - f(x_s, \theta^*)\big) = 0, \tag{18}$$

where $\mathcal{F}_t = \sigma\big(\{x_s, y_s\}_{s=1}^{t}\big)$.

As $x_t \perp y_t$, $\{\big(x_t^{\text{dir}}(y_t - f(x_t, \theta^*))\big)_{s=1}^{t}\}$ is also an MDS.

Note that each term is bounded. Therefore, The strong law of large numbers for martingale difference sequences holds. By Proposition 4.4, the queries are given as $x_s^{\text{dir}} = \alpha v_{0,j_s}$. So, it can be written as follows:

$$\frac{1}{t} \sum_{j=1}^{d} \sum_{s \le t: j_s = j} \alpha v_{0,j_s} \big(y_s - f(x_s, \theta^*)\big) \overset{a.s.}{\to} \mathbf{0}, \tag{19}$$

where $\mathbf{0} = [0, \ldots, 0]^\top$.

Now, note that the MAP estimator $\hat{\theta}_{t+1}$ is the estimator that maximizes log-likelihood. Since $\theta \in \mathbb{R}^d$ and differentiable everywhere, MAP is the solution of equation $\nabla_\theta \log p(\hat{\theta}_{t+1} \mid \mathcal{D}_t) = 0$.

for all s, and by (11), we get

$$\nabla_\theta \log p(\hat{\theta}_{t+1} \mid \mathcal{D}_t) = -\Lambda(\hat{\theta}_{t+1} - \theta_0) + \sum_{j=1}^{d} \sum_{s \le t: j_s = j} \alpha v_{0,j} (y_s - f(x_s, \hat{\theta}_{t+1})) = \mathbf{0}. \tag{20}$$

By multiplying (20) by $\frac{1}{t}$ and subtracting, we obtain the following:

$$\frac{1}{t} \sum_{j=1}^{d} \sum_{s \le t: j_s = j} \Big(\alpha(f(x_s, \hat{\theta}_{t+1}) - f(x_s, \theta^*)) + \Lambda^\top(\hat{\theta}_{t+1} - \theta_0)\Big) \overset{a.s.}{\to} \mathbf{0}. \tag{21}$$

Taking the inner product with any eigenvector of the initial Hessian $\{v_{0,1}, \cdots, v_{0,d}\}$, say $v_{0,j}$, we get

$$\frac{1}{t} \sum_{s \le t: j_s = j} \Big(\alpha(f(x_s, \hat{\theta}_{t+1}) - f(x_s, \theta^*)) + \lambda_{0,j} v_{0,j}^\top(\hat{\theta}_{t+1} - \theta^*) + \lambda_{0,j} v_{0,j}^\top(\theta^* - \theta_0)\Big) \overset{a.s.}{\to} \mathbf{0}. \tag{22}$$

Since $x_s^{\text{dir}} \parallel v_{j_s}$ and Λ_0 is positive semi-definite, $\lambda_{0,j} \ge 0$ for all j. This implies that $v_j^\top(\hat{\theta}_{t+1} - \theta^*)$ and $f(x_s, \hat{\theta}_{t+1}) - f(x_s, \theta^*)$ have the same sign. Therefore, (22) implies that $\forall j$, $v_j^\top(\hat{\theta}_{t+1} - \theta^*) \overset{a.s.}{\to} 0$. □

Proposition A.2. *Suppose that a Gaussian prior is adopted. Under the DURM algorithm, each eigenvalue converges to $\frac{\alpha^2}{4d}$:*

$$\lim_{t\to\infty} \frac{\lambda_{t,j}}{t} = \frac{\alpha^2}{4d}, \quad \forall j \in [d], \quad \textit{almost surely}. \tag{23}$$

This Proposition A.2 is also needed to show Theorem 4.5.

Proof of Proposition A.2.

Proof. First, we need to show that

$$\lim_{t\to\infty} \frac{\lambda_{t,j}}{t} - \frac{\alpha^2}{4}\frac{N_j(t)}{t} = 0 \tag{24}$$

where $N_j(t) = |\{s \mid j_s = j,\ 1 \le s < t\}|$, i.e., the number of direction $v_{0,j}$ is selected up to t.

Since we have $\nabla_\theta g(x,\theta) = x^{\text{dir}} f(x,\theta)\big(1 - f(x,\theta)\big)\big(2f(x,\theta) - 1\big)$, we get $\|\nabla_\theta g(x,\theta)\|_2 \le \|x^{\text{dir}}\|_2 \le \alpha$. Therefore, $g(x,\theta)$ is Lipschitz continuous in θ.

With Lipschitz continuity of g and Proposition A.1,

$$\frac{1}{t}\sum_{s<t:j_s=j} \Big| g(x_s,\hat{\theta}_t) - g(x_s,\theta^*)\Big| \le \frac{1}{t}\sum_{s<t:j_s=j} L\|\hat{\theta}_t - \theta^*\|_2 \overset{a.s.}{\to} 0. \tag{25}$$

In addition, $g(x_t,\theta^*) \overset{a.s.}{\to} \frac{1}{4}$ since f is continuous in θ. Thus, $\frac{\lambda_{t,j}}{t} = \frac{1}{t}\alpha^2 \sum_{s<t:j_s=j} g(x_s,\hat{\theta}_t)$ for all j, which implies

$$\frac{\lambda_{t,j}}{t} - \frac{\alpha^2}{4}\frac{N_j(t)}{t} \overset{a.s.}{\to} 0. \tag{26}$$

Now, we want to show that

$$\forall i,j \in [d], \quad \lim_{t\to\infty} \frac{N_i(t) - N_j(t)}{t} = 0 \quad \text{almost surely.} \tag{27}$$

From the previous result, we can say that for all $\epsilon > 0$ there exist $T_{1,i}$ and $T_{1,j}$ such that

$$\forall t > T_{1,j}, \quad \frac{4}{\alpha^2}\frac{\lambda_{t,j}}{t} \in \left[\frac{N_j(t)}{t} - \epsilon, \frac{N_j(t)}{t} + \epsilon\right], \tag{28}$$

and,

$$\forall t > T_{1,i}, \quad \frac{4}{\alpha^2}\frac{\lambda_{t,i}}{t} \in \left[\frac{N_i(t)}{t} - \epsilon, \frac{N_i(t)}{t} + \epsilon\right]. \tag{29}$$

Assume without loss of generality that $N_i > N_j$. Then

$$\frac{N_i(t) - N_j(t)}{t} > 2\epsilon \implies \lambda_{t,i} > \lambda_{t,j}. \tag{30}$$

Now, let $T_1 = \max\left\{T_{1,j}, T_{1,i}\right\}$, and for $t > T_1$,

$$\frac{4}{\alpha^2}\frac{\lambda_{t,j}}{t} \le \frac{N_j(t)}{t} + \epsilon < \frac{N_i(t)}{t} - \epsilon \le \frac{4}{\alpha^2}\frac{\lambda_{t,i}}{t}. \tag{31}$$

When $\lambda_{t,i} > \lambda_{t,j}$, we know $\lambda_{t,i}$ cannot be the minimum eigenvalue, so clearly $i \ne j_t$. Next, when $\frac{N_i(t)-N_j(t)}{t} < 2\epsilon$, if $i = j_t$ then $N_i(t+1) = N_i(t) + 1$ and $N_j(t+1) = N_j(t)$. Thus, at the next time $t+1$ we get $\frac{N_i(t)-N_j(t)}{t} < 2\epsilon + \frac{1}{t+1}$. Eventually, this term is bounded by $2\epsilon + \frac{1}{t+1}$ for all time t after.

$$\exists T_2 > \max(T_1, \frac{N_i(T_1) - N_j(T_1)}{2\epsilon}), \quad \forall t > T_2, \quad \frac{N_i(t) - N_j(t)}{t} \le 2\epsilon + \frac{1}{T_2}. \tag{32}$$

This holds for all pairs i,j, so the limit at the (27) goes to zero.

Finally, as $\sum_{j=1}^d N_j(t) = t$, it is clear that $\lim_{t\to\infty} \frac{dN_i(t) - \sum_{j=1}^d N_j(t)}{t} = 0$, which gives $\lim_{t\to\infty} \frac{N_i(t)}{t} = \frac{1}{d}$. In conclusion, $\frac{\lambda_{t,j}}{t} \overset{a.s.}{\to} \frac{\alpha^2}{4d}$. □

Proof of Theorem 4.5.

Proof. Theorem 4.5 is implied directly by Proposition A.2. By Proposition A.2,

$$\lim_{t\to\infty} H_t^{\pi_{\text{DURM}}} = \lim_{t\to\infty} \frac{\left(\prod_{j=1}^d \lambda_{t,j}\right)^{1/d}}{t} = \lim_{t\to\infty} \left(\prod_{j=1}^d \frac{\lambda_{t,j}}{t}\right)^{1/d} = \frac{\alpha^2}{4d}. \tag{33}$$

□

Proof of Theorem 4.6.

Proof. From (12), we know that each $\nabla_\theta^2 \log p(\theta \mid x_t, y_t) \succeq 0$, so for time t, Λ_t is a positive semi-definite matrix, $\lambda_{t,j} \geq 0$. From the AM-GM inequality,

$$(\det \Lambda_t)^{1/d} \leq \left(\prod_{j=1}^d \lambda_{t,j}\right)^{1/d} \leq \frac{1}{d}\left(\sum_{j=1}^d \lambda_{t,j}\right) \leq \frac{1}{d} \operatorname{tr} \Lambda_t. \tag{34}$$

The trace of the Hessian matrix is $\operatorname{tr} \Lambda_t = \operatorname{tr} \Lambda(\hat{\theta}_t; \mathcal{D}_t) + \sum_{s=1}^{t-1} \|x_s^{\text{dir}}\|^2 g(x_s, \hat{\theta}_t)$, and the last term is bounded by $\frac{\alpha^2}{4} t$. Thus, we conclude $\limsup_{t\to\infty} \frac{(\det \Lambda_t)^{1/d}}{t} \leq \limsup_{t\to\infty} \frac{1}{d} \frac{\operatorname{tr} \Lambda_t}{t} \leq \frac{\alpha^2}{4d}$, almost surely. □

A.2. Proofs for Section 4.2

Proof of Proposition 4.7.

Proof. Let's denote the objective in the procedure (5) as $C(x, \hat{\theta}_t)$. For the sake of simplicity, denote $\boldsymbol{\phi}_i := \phi(x_i + \hat{\theta}_{t,i})$ and $v := v_{\min}(\Lambda_t)$. Thus, $C(\boldsymbol{\phi}) = C(x, \hat{\theta}_t) = v^\top(1-\phi)\left(\prod_{i=1}^{d} \boldsymbol{\phi}_i\right)$ for $\boldsymbol{\phi} = (\boldsymbol{\phi}_i)_{i=1}^d \in [0,1]^d$. Then, (5) can be written as $\arg\max_{\boldsymbol{\phi}\in[0,1]^d} C(\boldsymbol{\phi})$.

Under the KKT conditions, we get:

$$\nabla_{\boldsymbol{\phi}} C(\boldsymbol{\phi}) - \mu_i = \left(-v_i \prod_{k=1}^{d} \phi_k + \frac{1}{\boldsymbol{\phi}_i} v^\top(\mathbf{1} - \boldsymbol{\phi}) \prod_{k=1}^{d} \boldsymbol{\phi}_k\right) - \mu_i = 0, \quad \forall i \in [d] \tag{35}$$

$$\mu^\top(\mathbf{1} - \boldsymbol{\phi}) = 0 \tag{36}$$

where $\mathbf{1} = (1, ..., 1)$.

Define two sets as $J = \{j \in [d] : \phi_j < 1\}$ and $NJ = \{j \in [d] : \phi_j = 1\}$. Note that μ_j=0 for $j \in J$ by the condition in (36), then the condition in (35) is reduced to $-v_j + \frac{v^\top(1-\boldsymbol{\phi})}{\phi_j} = 0$, i.e., $v_j\phi_j = v^\top(1-\boldsymbol{\phi})$ for $j \in J$. With this result, we get

$$\begin{aligned} v^\top(\mathbf{1} - \boldsymbol{\phi}) &= \sum_{j=1}^{d} v_j(1-\phi_j) && (37)\\ &= \sum_{j\in J} v_j(1-\phi_j) + \sum_{j\in NJ} v_j(1-\phi_j) && (38)\\ &= \sum_{j\in J}\left[v_j - v^\top(\mathbf{1}-\boldsymbol{\phi})\right] + 0 && (39)\\ &= \sum_{j\in J} v_j + |J| v^\top(\mathbf{1}-\boldsymbol{\phi}), && (40) \end{aligned}$$

hence $v^\top(\mathbf{1} - \boldsymbol{\phi}) = \frac{1}{|J|+1}\sum_{j\in J} v_j$. Recall that $v_j\phi_j = v^\top(1-\boldsymbol{\phi})$ for $j \in J$. Therefore, we get

$$\phi_i = \begin{cases} \frac{1}{v_j}\frac{1}{|J|+1}\sum_{j\in J} v_j & i \in J, \\ 1 & i \in NJ \end{cases}. \tag{41}$$

Now, $f(x_t, \hat{\theta}_t) = \prod_{i=1}^{d} \phi_i$ can be expressed as:

$$\begin{aligned} f(x_t, \hat{\theta}_t) &= \left(\prod_{j\in J} \phi_j\right)\left(\prod_{j\in NJ} \phi_j\right) && (42)\\ &= \frac{1}{\prod_{j\in J} w_j}\left(\frac{1}{|J|+1}\right)^{|J|}. && (43) \end{aligned}$$

where $w_i = \frac{v_i}{\sum_{j\in J} v_j}$ for $i \in J$. By the AM-GM inequality, $\prod_{j\in J} w_j \le \left(\frac{1}{|J|}\right)^{|J|}$, i.e., $\frac{1}{\prod_{j\in J} w_j} \ge |J|^{|J|}$. Next, since $\frac{v_i}{\sum_{j\in J} v_j} = \frac{1}{\phi_i}\frac{1}{|J|+1}$ and $\phi_i < 1$, we get $\frac{1}{|J|+1} < w_i$. In that $\sum_{i\in J} w_i = 1$ and $\frac{1}{|J|+1} < w_i$, $\prod_{i\in J} w_i$ is lower bounded by the product of weight distribution $\left(\frac{1}{|J|+1}, \cdots, \frac{1}{|J|+1}, \frac{2}{|J|+1}\right)$, i.e., $\frac{2}{(|J|+1)^{|J|}} \le \prod_{i\in J} w_i$. By setting $|J| = d$, we get the desired result. □

Proof of Proposition 4.8.

Proof. Since the log-likelihood is given by

$$\log p(\theta \mid x_t, y_t) = y_t \log f(x_t, \theta) + (1 - y_t) \log(1 - f(x_t, \theta)) \tag{44}$$

the gradient and Hessian of log-likelihood is given as

$$\nabla_\theta \log p(\theta \mid x_t, y_t) = (1 - \boldsymbol{\phi})\big(y_t - \frac{f(x_t, \theta)}{1 - f(x_t, \theta)}(1 - y_t)\big) \tag{45}$$

$$\nabla_\theta^2 \log p(\theta \mid x_t, y_t) = y_t \mathbf{D} + (1 - y_t)\left(-\frac{f(x_t, \theta)}{1 - f(x_t, \theta)}\mathbf{D} + \frac{f(x_t, \theta)}{(1 - f(x_t, \theta))^2}(1 - \boldsymbol{\phi})(1 - \boldsymbol{\phi})^\top\right) \tag{46}$$

where $\mathbf{D} = \operatorname{diag}\left(\left(\boldsymbol{\phi}_j(1 - \boldsymbol{\phi}_j)\right)_{j=1}^d\right)$.

First, we will prove that if the Hessian matrix at time t is diagonal, the DURM algorithm selects a problem such that only one element of vector $\boldsymbol{\phi}$ is $\frac{1}{2}$, and the others are 1. We will prove that every Hessian matrices are diagonal later.

Suppose that Λ_t is a diagonal matrix, then its eigenvectors are orthonormal basis vectors. We can say $v_{t,\min} = \mathbf{e}_{j_t}$,i.e., $j_t \in \arg\min_{j \in [d]} \lambda_{t,j}$, where $\mathbf{e}_j$ is the j-th basis vector.

Then the value $w_i = \frac{v_i}{\sum_{j \in J} v_j}$ in the proof of Proposition 4.7 should satisfy $\frac{1}{|J|+1} \leq w_i$ for $i \in J$. However, one can show that $w_{j_t} = 1$ and $w_j = 1$ for other indices $j \neq j_t$, regardless of J. Consequently, $J = \{j_t\}$, which means:

$$\boldsymbol{\phi}_i = \begin{cases} \frac{1}{2} & \text{if} \quad i = j_t \\ 1 & \text{else} \end{cases} \tag{47}$$

Now, we move to the next step. The conclusion above implies:

$$[(1 - \boldsymbol{\phi})(1 - \boldsymbol{\phi})^\top]_{i,j} = \begin{cases} (1 - \boldsymbol{\phi}_{j_t})^2 & \text{if} \quad i = j = j_t \\ 0 & \text{else} \end{cases} \tag{48}$$

$$\mathbf{D}_{i,j} = \begin{cases} \boldsymbol{\phi}_{j_t}(1 - \boldsymbol{\phi}_{j_t}) & \text{if} \quad i = j = j_t \\ 0 & \text{else} \end{cases} \tag{49}$$

i.e., every elements except the (j_t, j_t)-element are all zero. As $f(x_t, \theta) = \prod_{i=1}^d \phi(x_{t,i} + \theta_i) = \phi(x_{t,j_t} + \theta_{j_t})$, the Hessian matrix becomes:

$$\left[\nabla_\theta^2 \log p(\theta \mid x_t, y_t)\right]_{i,j} = \begin{cases} g(x_t, \theta) & \text{if} \quad i = j = j_t \\ 0 & \text{else} \end{cases} \tag{50}$$

where $g(x, \theta) = f(x, \theta)(1 - f(x, \theta))$.

Note that the Hessian matrix at the next step is still diagonal if the initial Hessian matrix is diagonal. Therefore, all Hessian matrices up to step t are also diagonal, so $\Lambda_t = \operatorname{diag}\left((\lambda_{t,j})_{j \in [d]}\right)$, and $\lambda_{t,j} = \lambda_{0,j} + \sum_{s<t: j_s = j} g(x_s, \hat{\theta}_t)$. □

Proposition A.3. *Suppose that a Gaussian prior with a diagonal covariance matrix is adopted. Under the DURM algorithm, the MAP estimator $\hat{\theta}_t$ converges almost surely to the true parameter θ^*:*

$$\lim_{t \to \infty} \hat{\theta}_t = \theta^*, \quad \textit{almost surely}.$$

Proof. This is very similar to Proposition A.1. Applying the setting of Proposition 4.8 gives the same equation as (22). □

Proposition A.4. *Suppose that a Gaussian prior with a diagonal covariance matrix is adopted. Under the DURM algorithm, each eigenvalue converges to $\frac{1}{4d}$:*

$$\lim_{t \to \infty} \frac{\lambda_{t,j}}{t} = \frac{1}{4d}, \quad \forall j \in [d], \quad \textit{almost surely}. \tag{51}$$

Proof. This is also very similar to Proposition A.2. The only difference is that $g(x, \theta)$ is Lipschitz continuous with respect to θ because $\nabla_\theta g(x, \theta) = f(x, \theta)(1 - \boldsymbol{\phi})(1 - 2f(x, \theta))$ is bounded below by $\|\nabla_\theta g(x, \theta)\|_2 \leq 1$ as each term is less than 1. □

Proof of Theorem 4.9.

Proof. By Proposition A.4,

$$\lim_{t\to\infty} H_t^{\pi_{\text{DURM}}} = \lim_{t\to\infty} \frac{(\prod_{j=1}^d \lambda_{t,j})^{1/d}}{t} = \lim_{t\to\infty} \left(\prod_{j=1}^d \frac{\lambda_{t,j}}{t}\right)^{1/d} = \frac{1}{4d}. \tag{52}$$

□

Proof of Theorem 4.10.

Proof. For any function $h(x, \theta)$ with variables x and θ, we introduce a simple notation as follows:

$$h_{s,t} = h(x_s, \hat{\theta}_t), \quad h_s^* = h(x_s, \theta^*). \tag{53}$$

For example, $f_{s,t}$ stands for $f(x_s, \hat{\theta}_t)$, and $\phi_{s,t,i}$ stands for the i-th element of vector $\phi_{s,t}$.

With this notation, the Hessian matrix is expressed as:

$$\Lambda_t = \Lambda(\hat{\theta}_t) + \sum_{s=1}^t \left(y_s \mathbf{D}_{s,t} + (1 - y_s) \left(\frac{f_{s,t}}{(1 - f_{s,t})^2} (1 - \phi_{s,t})(1 - \phi_{s,t})^\top - \frac{f_{s,t}}{1 - f_{s,t}} \mathbf{D}_{s,t} \right) \right), \tag{54}$$

where $\mathbf{D}_{s,t} = \operatorname{diag}\left(\left(\boldsymbol{\phi}_{s,t,j}(1 - \boldsymbol{\phi}_{s,t,j}) \right)_{j=1}^d \right)$.

And the trace of the Hessian matrix becomes:

$$\operatorname{tr} \Lambda_t = \operatorname{tr} \Lambda(\hat{\theta}_t) + \sum_{s=1}^t X_{s,t}(y_s) \tag{55}$$

where:

$$X_{s,t}(y) = (1 - y) \left(\frac{f_{s,t}}{(1 - f_{s,t})^2} \sum_{i=1}^d (1 - \phi_{s,t,i})^2 - \frac{f_{s,t}}{1 - f_{s,t}} \sum_{i=1}^d \phi_{s,t,i}(1 - \phi_{s,t,i}) \right) \tag{56}$$

$$+ y \sum_{i=1}^d \phi_{s,t,i}(1 - \phi_{s,t,i}). \tag{57}$$

First, we want to show $X_{s,t}(y)$ is Lipschitz continuous in θ to show that $\frac{1}{t} \sum_{s=1}^t |X_{s,t} - X_s^*| \overset{a.s.}{\to} 0$.

Let's denote:

$$A(x, \theta) = \frac{f}{(1 - f)^2} \sum_{i=1}^d (1 - \phi_i)^2, \tag{58}$$

$$B(x, \theta) = \frac{f}{1 - f} \sum_{i=1}^d \phi_i(1 - \phi_i), \tag{59}$$

$$C(x, \theta) = \sum_{i=1}^d \phi_i(1 - \phi_i). \tag{60}$$

Since $\frac{d}{d\theta_j}\phi_j = \phi_j(1-\phi_j)$, we get $\nabla_\theta f = (1-\phi)f$,

$$\nabla_\theta A(x,\theta) = 2\frac{f^2(1-\phi)}{(1-f)^3}\sum_{i=1}^{d}(1-\phi_i)^2 - 2\frac{f}{(1-f)^2}\phi(1-\phi_i)^2 + (1-\phi)A(\phi), \tag{61}$$

$$\nabla_\theta B(x,\theta) = -\frac{f^2(1-\phi)}{(1-f)^2}\sum_{i=1}^{d}\phi_i(1-\phi_i) + \frac{f}{1-f}\phi_i(1-\phi_i)(2\phi_i-1) + (1-\phi)B(\phi), \tag{62}$$

$$\left[\nabla_\theta C(x,\theta)\right]_i = \phi_i(1-\phi_i)(2\phi_i-1). \tag{63}$$

Consider the functions A, B, and C. Since every terms in numerators are bounded, the only possibility that these are unbounded is the case in which f is close to 1. Note that $\frac{1-\phi_i}{1-f} < 1$ since $\phi_i \le 1$ and $f = \prod_{i=1}^{d}\phi_i$. Therefore, they are bounded. Moreover, with the same logic and the boundedness of A and B, their gradients are also bounded. Consequently, $\nabla_\theta X_{s,t}(y) = (1-y)\left(\nabla_\theta A_{s,t} + \nabla_\theta B_{s,t}\right) + y\left(\nabla_\theta C_{s,t}\right)$ is bounded.

Now, as $X_{s,t}(y)$ is Lipschitz continuous in $\hat{\theta}_t$ with some constant L,

$$\frac{1}{t}\sum_{s=1}^{t}\left|X_{s,t}(y_s) - X_s^*(y_s)\right| \le \frac{1}{t}\sum_{s=1}^{t}L\|\hat{\theta}_t - \theta^*\|_2 \tag{64}$$

$$= L\|\hat{\theta}_t - \theta^*\|_2 \overset{a.s.}{\to} 0. \tag{65}$$

Second, let's define $Y_s^* := \mathbb{E}[X_s^* \mid \mathcal{F}_{s-1}]$, then $X_s^* - Y_s^*$ is a martingale difference sequence. Since each X_s^* and Y_s^* is bounded, by the law of large numbers for martingale,

$$\frac{1}{t}\sum_{s=1}^{t}(X_s^* - Y_s^*) \overset{a.s.}{\to} 0. \tag{66}$$

By the definition of X_s^*, we can calculate it, by replacing $f_{s,t}$ with f_s^* and $\phi_{s,t}$ with ϕ_s^* in $X_{s,t}$. Moreover, in that $\mathbb{E}[y_s] = f_s^*$, we finally get

$$Y_s^* = \frac{f_s^*}{1-f_s^*}\sum_{i=1}^{d}(1-\phi_{s,i}^*)^2 \tag{67}$$

$$= \frac{\prod_{i=1}^{d}\phi_{s,i}^*}{1-\prod_{i=1}^{d}\phi_{s,i}^*}\sum_{i=1}^{d}(1-\phi_{s,i}^*)^2. \tag{68}$$

Let's denote $\prod_{i=1}^{d}\phi_{s,i}^* = w$. Then, the candidate that maximizes $\sum_{i=1}^{d}(1-\phi_{s,i}^*)^2$ is either $[w,1,...,1]$ or $[w^{1/d},...,w^{1/d}]$. Thus, we find that the maximum value of (68) is either $\frac{w}{1-w}(1-w)^2$ or $\frac{w}{1-w}d(1-w^{\frac{1}{d}})^2$, whichever is larger for $0 \le w \le 1$. We know that

$$\max_{0\le w\le 1}\frac{w}{1-w}d(1-w^{\frac{1}{d}})^2 = \max_{0\le u\le 1}\frac{u^d}{1-u^d}d(1-u)^2 \le \max_{0\le u\le 1}\frac{u}{1-u}(1-u)^2 \tag{69}$$

Therefore, the maximum value is $\frac{w}{1-w}(1-w)^2 = w(1-w)$ achieved by $[w,1,...,1]$. Taking $w = \frac{1}{2}$ gives $\max_{x_s\in\mathcal{X}} Y_s^* = \frac{1}{4}$.

Now Y_s^* is bounded by its maximum value $\frac{1}{4}$ when $\phi_s^* = [\frac{1}{2},1,...,1]$, WLOG. In conclusion,

$$\limsup_{t\to\infty}\frac{(\det\Lambda_t)^{1/d}}{t} \le \limsup_{t\to\infty}\frac{1}{d}\frac{\operatorname{tr}\Lambda_t}{t} \le \frac{1}{4d}, \quad \text{almost surely.} \tag{70}$$

□

B. Additional Experiments

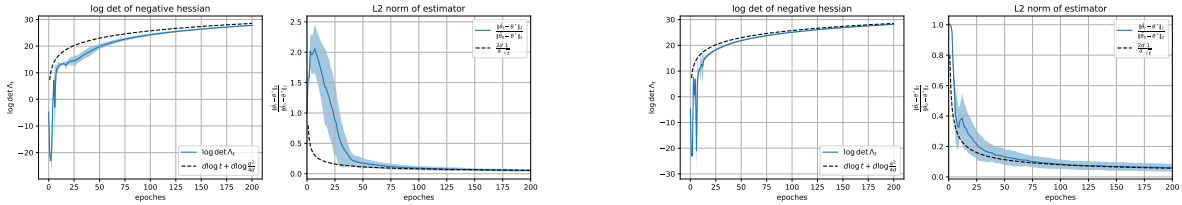
B.1. Additional Experiments for Synthetic Data

Figure 5 presents the Logistic model synthetic data experiment using non-Gaussian priors, such as the Gaussian Mixture Model and Multivariate t-distribution as the initial priors. We conducted 50 iterations of the experiment. The initialization conditions are described below. The Gaussian Mixture Model and Multivariate t-distribution priors follow equations (71) and (72), respectively.

$$p(\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\theta; \mu_k, \Sigma_k) \tag{71}$$

$$p(\theta) = \frac{\Gamma[(\nu+d)/2]}{\Gamma(\nu/2)\nu^{d/2}\pi^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \left[1 + \frac{1}{\nu}(\theta - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\theta - \boldsymbol{\mu})\right]^{-(\nu+d)/2} \tag{72}$$

- Gausssian Mixture Model Prior : Figure 5 (a)
  - $d = 4$, $K = 3$, $(\pi_1, \pi_2, \pi_3) = (\frac{1}{6}, \frac{1}{2}, \frac{1}{3})$, $(\Sigma_1, \Sigma_2, \Sigma_3) = (\frac{1}{2}I_d, I_d, \frac{1}{4}I_d)$, $\mu_1 = (0,0,0,0)$, $\mu_2 = (-1,-1,-1,-1)$, $\mu_3 = (1,1,1,1)$
  - $\theta^* = (0.9, 0.3, 0.3, 0.1)$, $\hat{\theta}_0 = (1.0, 0.0, 0.6, 1.0)$, $(\lambda_{0,1}, \lambda_{0,2}, \lambda_{0,3}, \lambda_{0,4}) = (0.9, 0.1, 0.3, 0.3)$
- Multi-variate T-distribution model Prior : Figure 5 (b)
  - $d = 4$, $\nu = 5$, $\mu = (1.0, 0.0, 0.6, 1.0)$, $\lambda(\Sigma^{-1}) = (0.9, 0.1, 0.3, 0.3)$
  - $\theta^* = (0.9, 0.3, 0.3, 0.1)$, $\hat{\theta}_0 = (1.0, 0.0, 0.6, 1.0)$, $(\lambda_{0,1}, \lambda_{0,2}, \lambda_{0,3}, \lambda_{0,4}) = (1.5, 0.17, 0.5, 0.5)$

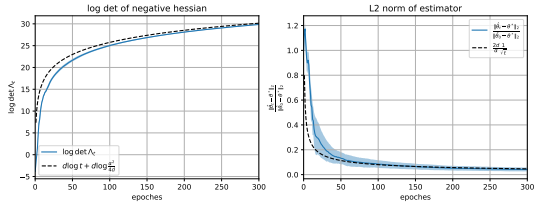


((a)) Logistic model, Gaussian Mixture Model Prior

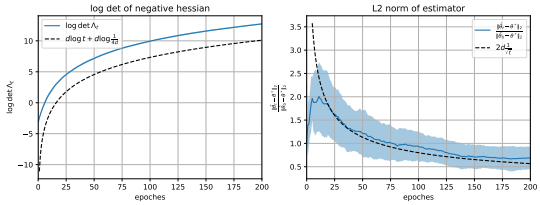
((b)) Logistic model, Multivariate t-distribution Prior

Figure 5: Non Gaussian Prior applied synthetic data experiments

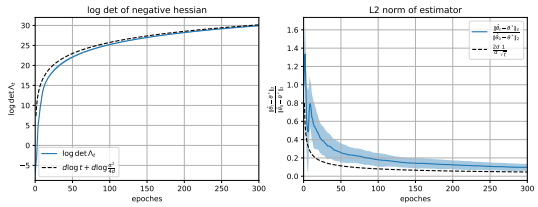
Figure 6 presents additional experiments using synthetic data based on the Logistic and MIRT models ($d = 4$), with various choices of θ^* and Gaussian priors characterized by the mean vector $\hat{\theta}_0$ and $(\lambda 0, 1, \ldots, \lambda_{0,d})$. For each combination of θ, $\hat{\theta}_0$, and $(\lambda 0, 1, \ldots, \lambda_{0,d})$, we conducted 50 iterations of experiments.



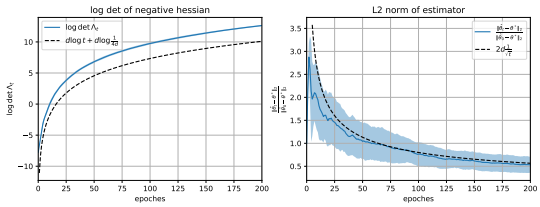
((a)) Logistic model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (5.24, 0.07, 0.2, 0.33)$
$\hat{\theta}_0 = (0.49, 0.26, 0.09, 0.21)^\top, \theta^* = (0.88, 0.12, 0.84, 0.74)^\top,$



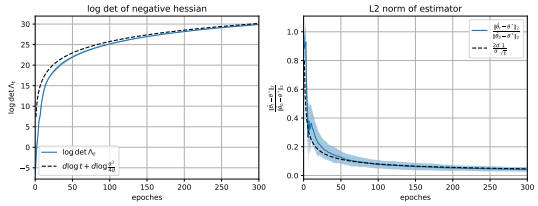
((b)) MIRT model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (0.27, 0.49, 0.66, 0.49)$
$\hat{\theta}_0 = (0.79, 0.62, 0.88, 0.46)^\top, \theta^* = (0.21, 0.24, 0.63, 0.47)^\top$



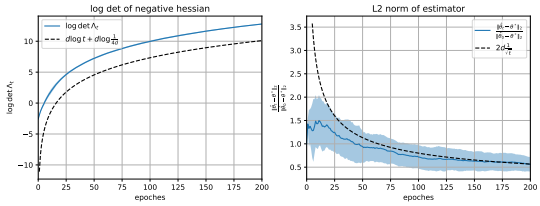
((c)) Logistic model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (5.17, 0.97, 0.01, 0.02)$
$\hat{\theta}_0 = (0.87, 0.21, 0.83, 0.49)^\top, \theta^* = (0.75, 0.66, 0.91, 0.41)^\top,$



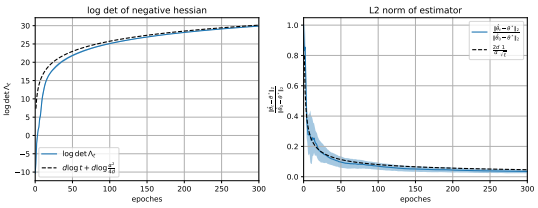
((d)) MIRT model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (0.43, 0.06, 0.33, 0.05)$
$\hat{\theta}_0 = (0.08, 0.43, 0.07, 0.45)^\top, \theta^* = (0.36, 0.26, 0.92, 0.9)^\top$



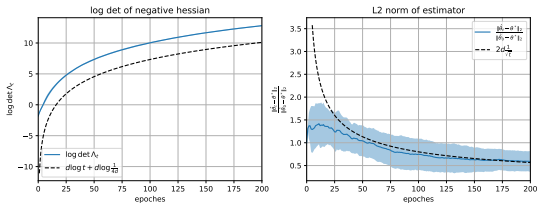
((e)) Logistic model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (7.17, 0.27, 0.12, 0.01)$
$\hat{\theta}_0 = (0.18, 0.66, 0.7, 0.82)^\top, \theta^* = (0.72, 0.54, 0.96, 0.04)^\top,$



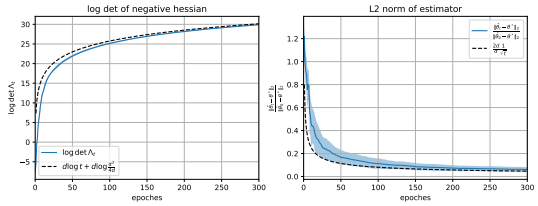
((f)) MIRT model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (0.59, 0.26, 0.74, 0.66)$
$\hat{\theta}_0 = (0.13, 0.8, 0.3, 0.27)^\top, \theta^* = (0.14, 0.96, 0.93, 0.93)^\top$



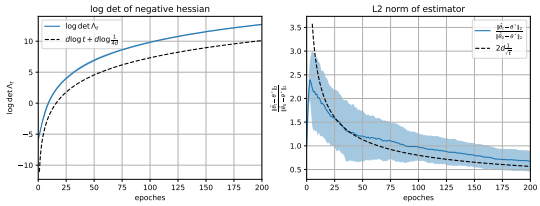
((g)) Logistic model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (3.99, 0.0, 0.18, 0.21)$
$\hat{\theta}_0 = (0.11, 0.99, 0.56, 0.92)^\top, \theta^* = (0.93, 0.16, 0.19, 0.59)^\top,$



((h)) MIRT model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (0.66, 0.79, 0.43, 0.72)$
$\hat{\theta}_0 = (0.52, 0.94, 0.54, 0.15)^\top, \theta^* = (0.8, 0.93, 0.6, 0.98)^\top$



((i)) Logistic model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (3.8, 0.68, 0.0, 0.06)$
$\hat{\theta}_0 = (0.72, 0.33, 0.89, 0.78)^\top, \theta^* = (0.23, 0.44, 0.55, 0.36)^\top,$



((j)) MIRT model, $(\lambda_{0,1}, .., \lambda_{0,4}) = (0.2, 0.2, 0.15, 0.45)$
$\hat{\theta}_0 = (0.74, 0.41, 0.51, 0.12)^\top, \theta^* = (0.13, 0.59, 0.9, 0.56)^\top$

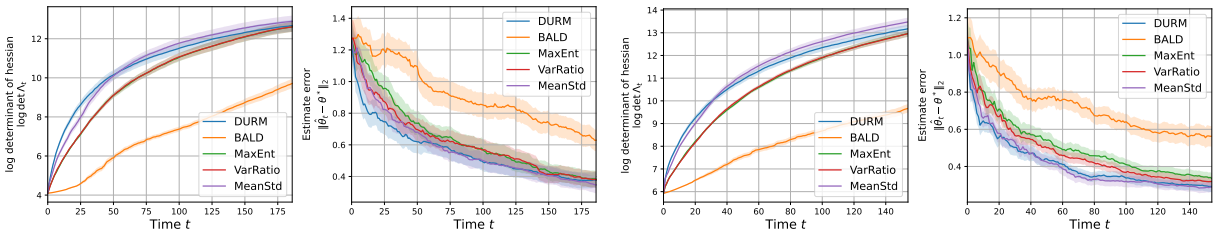
Figure 6: Simulation results obtained with synthetic data for logistic response model and MIRT response model with randomly chosen parameters

B.2. Additional Experiments for Realworld Data

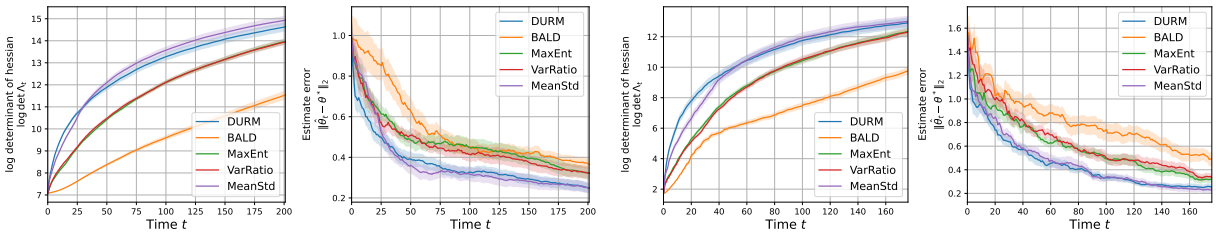
B.2.1. Various Dataset Experiments

In Figure 7, we simulated experiments on different subsets of a real-world dataset using the logistic model, with settings similar to those in Figure 3. As described in Section 5.2, a random sample of 38 students was drawn from a total of 388, representing 10% of the population. Similarly, for students in grades 1, 5, and 9, we conducted the experiment by randomly sampling 10% of the total student population in each grade. During the experiment, the prior distribution was assumed to be Gaussian, with the mean vector and covariance matrix of the students' parameter vectors used to define the distribution. The response to each problem was deterministic, as labeled in the student-problem interaction table. A detailed description of the 7th-grade dataset is provided in the main paper.

- *1st grade dataset* contains 116,348 samples of interactions between 355 students and 896 questions.
  The algorithms are tested using randomly chosen 35 students as if these students had just joined the platform.
- *5th grade dataset* contains 124,678 samples of interactions between 343 students and 1,206 questions.
  The algorithms are tested using randomly chosen 34 students as if these students had just joined the platform.
- *9th grade dataset* contains 59,580 samples of interactions between 203 students and 443 questions.
  The algorithms are tested using randomly chosen 20 students as if these students had just joined the platform.



((a)) Grade 1 ((b)) Grade 5

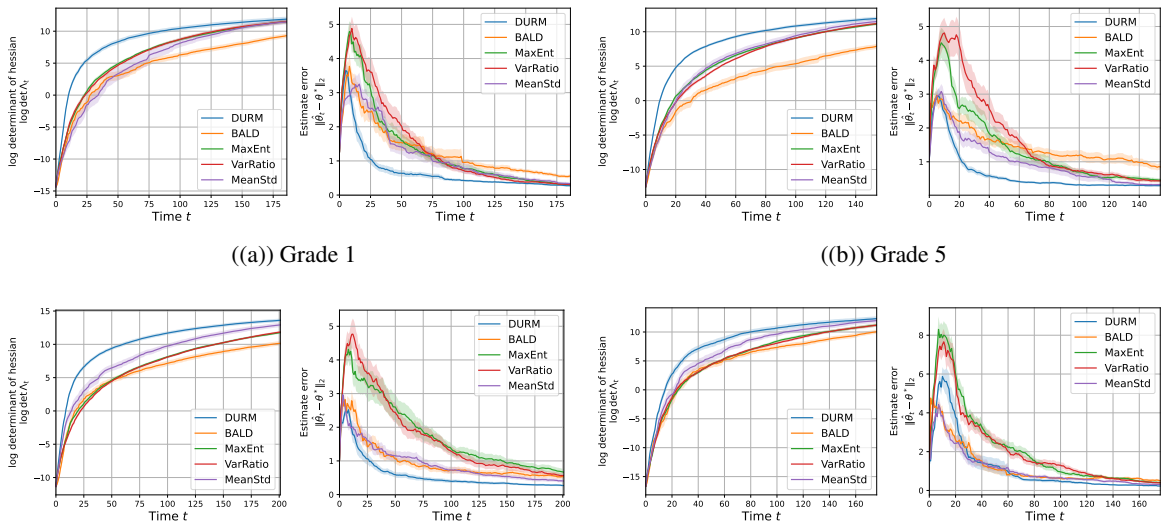


((c)) Grade 7 ((d)) Grade 9

Figure 7: Simulation results obtained with *i-scream edu 1st, 5th, 7th, 9th grade dataset* (logistic response model)

On Figure 7, initial prior distribution follows under descriptions.

- *1st grade* : $\theta^* \sim \mathcal{N}(\hat{\theta}_0, \Lambda_0^{-1})$ such that $\hat{\theta}_0 = (-0.41, 0.09, -0.33, -0.35)$, $\lambda(\Lambda_0^{-1}) = (0.94, 0.22, 0.27, 0.3)$
- *5th grade* : $\theta^* \sim \mathcal{N}(\hat{\theta}_0, \Lambda_0^{-1})$ such that $\hat{\theta}_0 = (-0.13, 0.38, -0.26, 0.13)$, $\lambda(\Lambda_0^{-1}) = (1.17, 0.21, 0.14, 0.08)$
- *7th grade* : $\theta^* \sim \mathcal{N}(\hat{\theta}_0, \Lambda_0^{-1})$ such that $\hat{\theta}_0 = (-0.12, -0.01, 0.0, -0.11)$, $\lambda(\Lambda_0^{-1}) = (0.65, 0.2, 0.06, 0.1)$
- *9th grade* : $\theta^* \sim \mathcal{N}(\hat{\theta}_0, \Lambda_0^{-1})$ such that $\hat{\theta}_0 = (0.05, -0.01, -0.11, -0.13)$, $\lambda(\Lambda_0^{-1}) = (3.01, 0.31, 0.41, 0.46)$



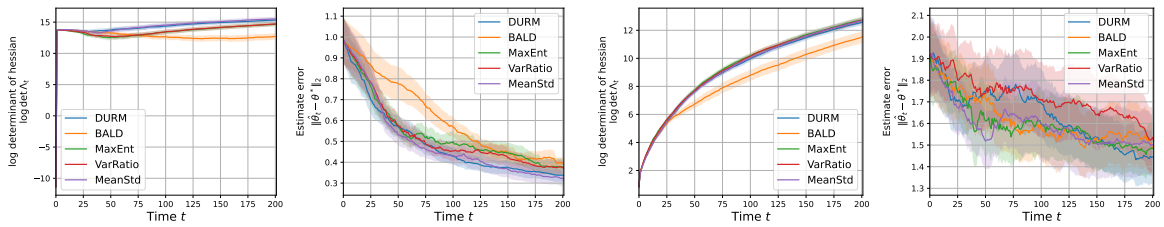
((a)) Grade 1 ((b)) Grade 5

((c)) Grade 7 ((d)) Grade 9

Figure 8: Simulation results from *i-scream edu 1st, 5th, 7th, 9th grade dataset* for logistic response model with the flattened prior

In Figure 8, we simulated the scenario of a spread covariance matrix, as described in the paper, representing the case where the prior is flat. The experiment used the same settings as in Figure 7, but with the covariance matrix eigenvalues scaled by a factor of 100 to make the evaluation more challenging. The initial prior settings for the experiments in Figure 8 are described below.

- *1st grade* : $\theta^* \sim \mathcal{N}(\hat{\theta}_0, \Lambda_0^{-1})$ such that $\hat{\theta}_0 = (-0.41, 0.09, -0.33, -0.35)$, $\lambda(\Lambda_0^{-1}) = (94, 22, 27, 30)$
- *5th grade* : $\theta^* \sim \mathcal{N}(\hat{\theta}_0, \Lambda_0^{-1})$ such that $\hat{\theta}_0 = (-0.13, 0.38, -0.26, 0.13)$, $\lambda(\Lambda_0^{-1}) = (117, 21, 14, 8)$
- *7th grade* : $\theta^* \sim \mathcal{N}(\hat{\theta}_0, \Lambda_0^{-1})$ such that $\hat{\theta}_0 = (-0.12, -0.01, 0.0, -0.11)$, $\lambda(\Lambda_0^{-1}) = (65, 20, 6, 10)$
- *9th grade* : $\theta^* \sim \mathcal{N}(\hat{\theta}_0, \Lambda_0^{-1})$ such that $\hat{\theta}_0 = (0.05, -0.01, -0.11, -0.13)$, $\lambda(\Lambda_0^{-1}) = (301, 31, 41, 46)$



((a)) Logistic model ((b)) MIRT model

Figure 9: Simulation results obtained with *i-scream edu 7-th grade dataset* for logistic response model and MIRT model with the multivariate-t distribution prior.

In Figure 9, we used the 7th-grade dataset, as in Figures 7 and 8, but with a multivariate t-distribution as the initial prior instead of a Gaussian prior. The prior distribution follows the equations in (72), and we fitted the parameters of the multivariate t-distribution to the students' parameter vector set ($\nu = 5$, $d = 4$). The specifics for both the Logistic and MIRT models are described below.

- Logistic Response model : $\mu = (-0.11, 0.0, 0.01, -0.11)$, $\lambda(\mathbf{\Sigma}) = (0.39, 0.12, 0.04, 0.06)$
- MIRT model : $\mu = (1.38, 1.42, 1.43, 1.38)$, $\lambda(\mathbf{\Sigma}) = (1.22, 0.41, 0.31, 0.36)$