Integrating Inference and Experimental Design for Contextual Behavioral Model Learning

Gongtao Zhou Haoran Yu [1]

School of Computer Science & Technology
Beijing Institute of Technology

AAAI'25

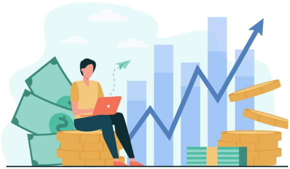
[1]Corresponding author.

Contextual Behavioral Model

- Contextual behavior model: user behavior is based on their contextual information

bidding in auctions

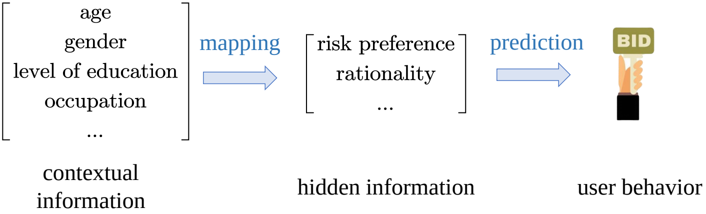
offering in bargainings



selecting in portfolios

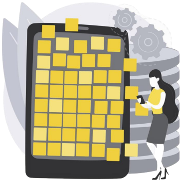
Contextual Behavioral Model

- Offline Learning: learn the relation between contextual information and user hidden information
  - Cons: require a large offline dataset



Contextual Behavioral Model

- Online learning: design sequential experiments to collect the most informative user behavioral data for learning
  - Cons: strategic environment for user can be designed
- Related work: assume linear context-valuation mappings
- Our work: explore a general setting: (i) mapping can be non-linear (ii) hidden information is multidimensional

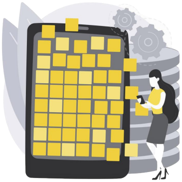


collecting dataset
(online)

training model
(online / offline)

Contextual Behavioral Model

- Online learning: design sequential experiments to collect the most informative user behavioral data for learning
  - Cons: strategic environment for user can be designed
- Related work: assume linear context-valuation mappings
- Our work: explore a general setting: (i) mapping can be non-linear (ii) hidden information is multidimensional



collecting dataset
(online)

training model
(online / offline)

Contextual Behavioral Model

- Question: how to design sequential experiments to learn an accurate contextual behavioral model?

Problem

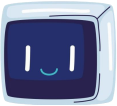
Portfolio Choice

- For simplicity, our problem begins with a single experiment involving one investor

observable:
context information $\boldsymbol{x}$

unobservable:
risk preference $r(x)$
rationality $\lambda(x)$

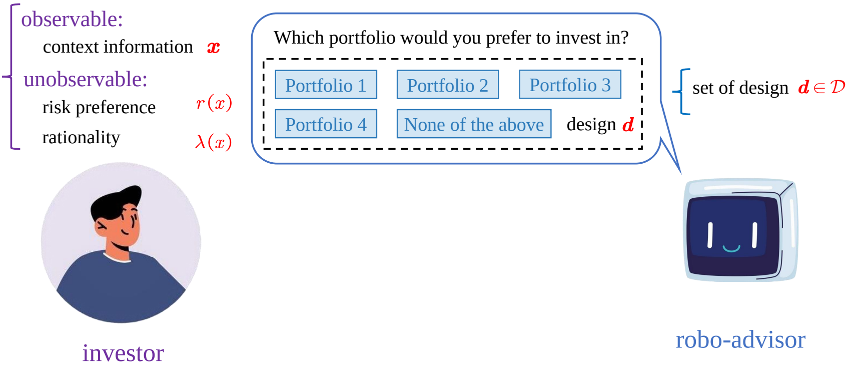
investor



robo-advisor

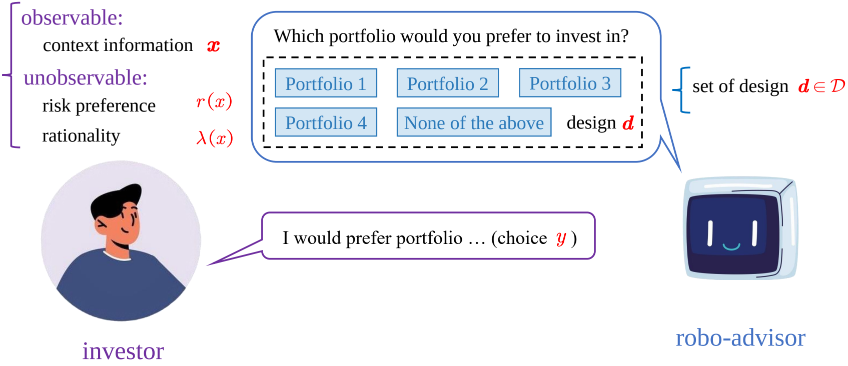
Portfolio Choice

- For simplicity, our problem begins with a single experiment involving one investor



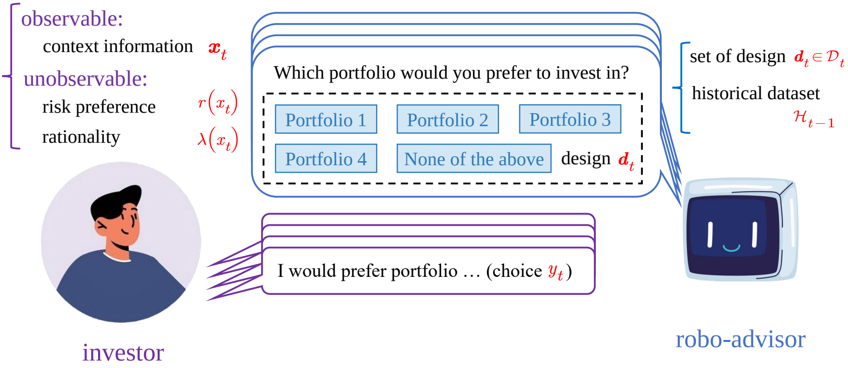
Portfolio Choice

- For simplicity, our problem begins with a single experiment involving one investor



Portfolio Choice

- Sequential experiments: in each period $t = 1, 2, \ldots, T$, robo-advisor will interact with investors
    - Investors of each period can be multiple or different
    - historical dataset $\mathcal{H}_{t-1} \triangleq \{(\mathbf{x}_1, \mathbf{d}_1, y_1), \ldots, (\mathbf{x}_{t-1}, \mathbf{d}_{t-1}, y_{t-1})\}$



Investor Strategy

- Investor Strategy: select the portfolio that aligns best with investor's preference, or opt to forgo investing
  - $r(\boldsymbol{x})$: risk preference (if $r(\boldsymbol{x}) < 0$, the investor is risk-seeking)
  - $\lambda(\boldsymbol{x})$: rationality (if $\lambda(\boldsymbol{x}) \to \infty$, the investor is fully rational)
  - $\boldsymbol{d} = \{(m_1, v_1), ..., (m_K, v_K)\}$: designed investment portfolios
    - m: return mean
    - v: return variance
    - K: the number of portfolios in the design
- Markowitz's mean-variance model

$$p(y = k|\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{v}) = \frac{\exp(\lambda(\boldsymbol{x})(m_k - r(\boldsymbol{x})v_k))}{\sum_{\tilde{k}=1}^{K} \exp(\lambda(\boldsymbol{x})(m_{\tilde{k}} - r(\boldsymbol{x})v_{\tilde{k}}))}$$

Investor Strategy

- Investor Strategy: select the portfolio that aligns best with investor's preference, or opt to forgo investing
  - $r(\boldsymbol{x})$: risk preference (if $r(\boldsymbol{x}) < 0$, the investor is risk-seeking)
  - $\lambda(\boldsymbol{x})$: rationality (if $\lambda(\boldsymbol{x}) \to \infty$, the investor is fully rational)
  - $\boldsymbol{d} = \{(m_1, v_1), ..., (m_K, v_K)\}$: designed investment portfolios
    - m: return mean
    - v: return variance
    - K: the number of portfolios in the design
- Markowitz's mean-variance model

$$p(y = k|\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{v}) = \frac{\exp(\lambda(\boldsymbol{x})(m_k - r(\boldsymbol{x})v_k))}{\sum_{\tilde{k}=1}^{K} \exp(\lambda(\boldsymbol{x})(m_{\tilde{k}} - r(\boldsymbol{x})v_{\tilde{k}}))}$$

Investor Strategy

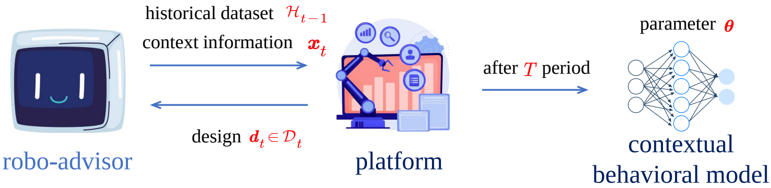
- Investor Strategy: select the portfolio that aligns best with investor's preference, or opt to forgo investing
  - $r(\boldsymbol{x})$: risk preference (if $r(\boldsymbol{x}) < 0$, the investor is risk-seeking)
  - $\lambda(\boldsymbol{x})$: rationality (if $\lambda(\boldsymbol{x}) \to \infty$, the investor is fully rational)
  - $\boldsymbol{d} = \{(m_1, v_1), ..., (m_K, v_K)\}$: designed investment portfolios
    - m: return mean
    - v: return variance
    - K: the number of portfolios in the design

- Markowitz's mean-variance model

$$p(y = k | \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{v}) = \frac{\exp(\lambda(\boldsymbol{x})(m_k - r(\boldsymbol{x})v_k))}{\sum_{\tilde{k}=1}^{K} \exp(\lambda(\boldsymbol{x})(m_{\tilde{k}} - r(\boldsymbol{x})v_{\tilde{k}}))}$$

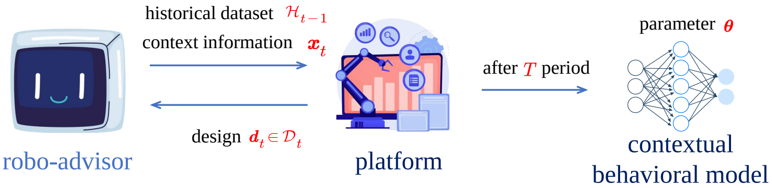
Platform Strategy

- Platform Strategy: collect experimental data to facilitate the learning of a more accurate behavioral model
  - input: contextual information $\boldsymbol{x}_t$
  - output: hidden information $\hat{r}_{\boldsymbol{\theta}}(x)$ and $\hat{\lambda}_{\boldsymbol{\theta}}(x)$



Platform Strategy

- Platform Strategy: collect experimental data to facilitate the learning of a more accurate behavioral model
  - input: contextual information $\boldsymbol{x}_t$
  - output: hidden information $\hat{r}_{\boldsymbol{\theta}}(x)$ and $\hat{\lambda}_{\boldsymbol{\theta}}(x)$



Platform's Experimental Design Problem

Given $\mathcal{H}_{t-1}$ and $\mathbf{x}_t$, how to optimize the design $\mathbf{d}_t \in \mathcal{D}_t$ for training an accurate θ-parameterized neural network?

Solution

Expected Information Gain

- Question1: how does the platform select design d?
- Solution: we select the design to maximize the Expected Information Gain (EIG)

$$\mathrm{IG}(\boldsymbol{d}, y) \triangleq H[p(\boldsymbol{\theta})] - H[p(\boldsymbol{\theta}|\boldsymbol{x}, y, \boldsymbol{d})]$$

- $p(\boldsymbol{\theta})$: prior distribution of θ
- $p(\boldsymbol{\theta}|\boldsymbol{x}, y, \boldsymbol{d})$: posterior distribution based on experiment
- $H[\cdot]$: information entropy

$$\mathrm{EIG}(\boldsymbol{d}) \triangleq \mathbb{E}[\mathrm{IG}(\boldsymbol{d}, y)]$$

- the investor choice y depends on the context $\boldsymbol{x}$ and the design $\boldsymbol{d}$, its distribution can be estimated

Expected Information Gain

- Question1: how does the platform select design d?
- Solution: we select the design to maximize the Expected Information Gain (EIG)

$$\text{IG}(\boldsymbol{d}, y) \triangleq H[p(\boldsymbol{\theta})] - H[p(\boldsymbol{\theta}|\mathbf{x}, y, \boldsymbol{d})]$$

  - $p(\boldsymbol{\theta})$: prior distribution of θ
  - $p(\boldsymbol{\theta}|\mathbf{x}, y, \boldsymbol{d})$: posterior distribution based on experiment
  - $H[\cdot]$: information entropy

$$\text{EIG}(\boldsymbol{d}) \triangleq \mathbb{E}[\text{IG}(\boldsymbol{d}, y)]$$

  - the investor choice y depends on the context $\boldsymbol{x}$ and the design $\boldsymbol{d}$, its distribution can be estimated

Expected Information Gain

- Question1: how does the platform select design d?
- Solution: we select the design to maximize the Expected Information Gain (EIG)

$$\text{IG}(\boldsymbol{d}, y) \triangleq H[p(\boldsymbol{\theta})] - H[p(\boldsymbol{\theta}|\boldsymbol{x}, y, \boldsymbol{d})]$$

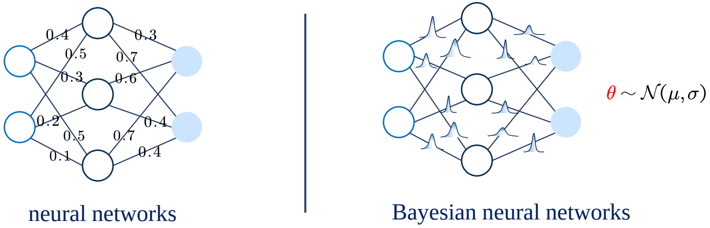
  - $p(\boldsymbol{\theta})$: prior distribution of θ
  - $p(\boldsymbol{\theta}|\boldsymbol{x}, y, \boldsymbol{d})$: posterior distribution based on experiment
  - $H[\cdot]$: information entropy

$$\text{EIG}(\boldsymbol{d}) \triangleq \mathbb{E}[\text{IG}(\boldsymbol{d}, y)]$$

  - the investor choice y depends on the context $\boldsymbol{x}$ and the design $\boldsymbol{d}$, its distribution can be estimated

Weight Uncertainty

- Question2: how to calculate EIG?
  - neural network has deterministic weights, making it impossible to compute information entropy
- Solution: consider Bayesian neural network
  - $\phi = (\boldsymbol{\mu}, \boldsymbol{\sigma})$: parameters of Bayesian neural network



Weight Uncertainty

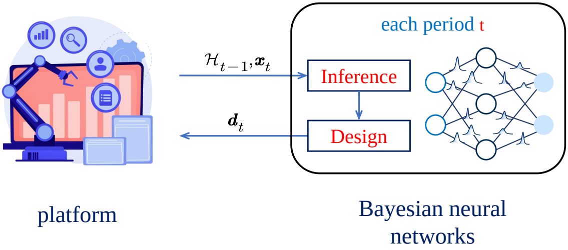
- For computational tractability, we employ variational inference to construct $q(\boldsymbol{\theta}|\phi)$ to approximate $p(\boldsymbol{\theta}|\mathcal{H}_{t-1})$
- the loss function for training Bayesian neural network:

$$\begin{aligned}\phi^* &= \arg\min_{\phi} \mathrm{KL}[q(\boldsymbol{\theta}|\phi)||p(\boldsymbol{\theta}|\mathcal{H}_{t-1})] \\ &= \arg\min_{\phi} \mathrm{KL}[q(\boldsymbol{\theta}|\phi)||p(\boldsymbol{\theta})] - \mathbb{E}_{q(\boldsymbol{\theta}|\phi)}[\log p(\mathcal{H}_{t-1}|\boldsymbol{\theta})]\end{aligned}$$

  - The first term is the KL divergence, which represents the difference between the variational distribution and the prior
  - The second term represents the deviation between the model and the collected historical data

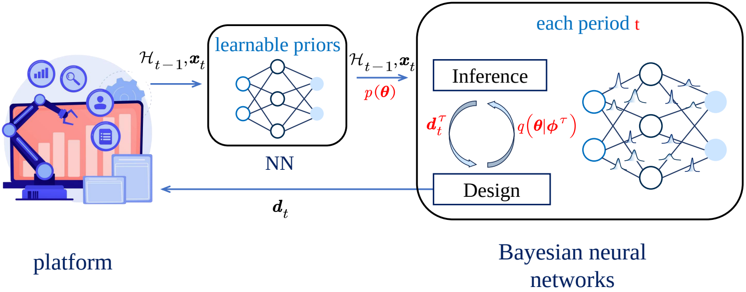
Inference-then-Design

- We present our **I**nference-then-**D**esign (**ID**) method
  - Inference: learn ϕ by training Bayesian neural network
  - Design: choose the design to maximize the estimated EIG



Improved Inference-then-Design Methods

- Integrated **I**nference-and-**D**esign with **L**earnable **P**riors
  - Learnable prior: a random initialized prior distribution $p(\boldsymbol{\theta})$ will disrupt the inference step during the early periods
  - Integrated Inference-and-Design: separating inference and design may lead to suboptimal performance, as the optimization of ϕ fails to maximize information gain



Improved Inference-then-Design Methods

- Loss function for iterative optimization:
  - ϕ^τ and $\boldsymbol{d}_t^\tau$ are the decision variables obtained in τ-th iteration
  - Optimize ϕ^τ while keeping $\boldsymbol{d}_t = \boldsymbol{d}_t^{\tau-1}$:

$$\phi^\tau = \arg\min_{\phi} \mathrm{KL}[q(\boldsymbol{\theta}|\phi)||p(\boldsymbol{\theta}|\mathcal{H}_{t-1})] - \mathbb{E}_{q(\boldsymbol{\theta}|\phi)}[\log \mathrm{KL}[p(y_t|\boldsymbol{x}_t, \boldsymbol{d}_t^{\tau-1}, \boldsymbol{\theta})||p(y_t|\boldsymbol{x}_t, \boldsymbol{d}_t^{\tau-1})]]$$

  - Optimize $\boldsymbol{d}_t^\tau$ while keeping $\phi = \phi^\tau$:

$$\boldsymbol{d}_t^\tau = \arg\max_{\boldsymbol{d}_t} \int q(\boldsymbol{\theta}|\phi^\tau) \log \mathrm{KL}[p(y_t|\boldsymbol{x}_t, \boldsymbol{d}_t, \boldsymbol{\theta})||p(y_t|\boldsymbol{x}_t, \boldsymbol{d}_t)] d\boldsymbol{\theta}$$

Experiments

Experimental Settings

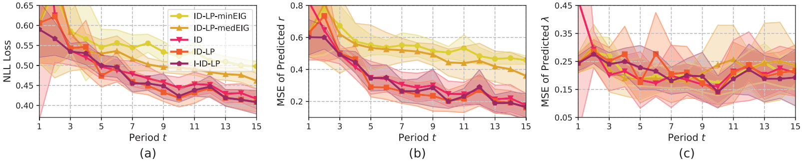
- Our methods
  - **ID**, **ID-LP**, **I-ID-LP**
  - **ID-LP-minEIG**, **ID-LP-medEIG**
- Comparison methods
  - **PreEntropy**, ***MinMaxPro***
  - **LaplaceInfer**
  - **Random**
  - ***MaxMean***, ***MaxVar***
  - ***MaxMean+Var***, ***MaxMean-Var***

Experimental Settings

- **Datasets**
  - Set of design $\mathcal{D}$
    - In each period t, the platform utilizes a new set of designs $\mathcal{D}_t$
    - The mean m and variance v of returns for portfolios in the design $\boldsymbol{d}_t$ are sampled from a uniform distribution
  - Investor information
    - Context vector $\boldsymbol{x}$ is generated by uniform distribution
    - Five distinct mapping relationships are defined between contextual information $\boldsymbol{x}$ and hidden information $r(\boldsymbol{x})$ and $\lambda(\boldsymbol{x})$

Results

- **Comparison Among ID Methods**
  - NLL Loss: the negative log-likelihood loss of predicting investor choice y in $\mathcal{H}_{test}$;
  - MSEs of Predicting $r(\boldsymbol{x})$ and $\lambda(\boldsymbol{x})$: the mean squared errors between the outputs of the neural network and the actual $r(\boldsymbol{x})$, $\lambda(\boldsymbol{x})$ of investors in $\mathcal{H}_{test}$.



Comparison of ID Methods Under Setting A (20 Random Seeds)

Results

Table: Negative Log-Likelihood Loss on Testing Dataset

Method	Setting A	Setting B	Setting C
ID	0.416 ± 0.025	0.319 ± 0.017	0.397 ± 0.010
ID-LP	0.409 ± 0.025	0.306 ± 0.018	0.390 ± 0.007
I-ID-LP	$\mathbf{0.408 \pm 0.029}$	$\mathbf{0.297 \pm 0.010}$	$\mathbf{0.387 \pm 0.007}$
ID-LP-minEIG	0.498 ± 0.012	0.427 ± 0.026	0.565 ± 0.113
ID-LP-medEIG	0.461 ± 0.019	0.331 ± 0.010	0.435 ± 0.047
PreEntropy	0.557 ± 0.111	0.366 ± 0.028	0.634 ± 0.216
MinMaxPro	0.611 ± 0.122	0.366 ± 0.030	0.705 ± 0.196
LaplaceInfer	0.462 ± 0.026	0.327 ± 0.012	0.411 ± 0.013
Random	0.459 ± 0.022	0.329 ± 0.012	0.425 ± 0.035
MaxMean	0.468 ± 0.003	0.404 ± 0.045	0.510 ± 0.012
MaxVar	0.560 ± 0.008	0.340 ± 0.008	0.624 ± 0.010
MaxMean+Var	0.468 ± 0.003	0.336 ± 0.004	0.501 ± 0.003
MaxMean−Var	0.512 ± 0.013	0.803 ± 0.424	0.562 ± 0.063

Conclusion

- We propose experimental design methods for learning general contextual behavioral models
  - Challenge: collect more informative data
  - Novelty: integrate variational inference with experimental design

THANK YOU