

Table of Contents

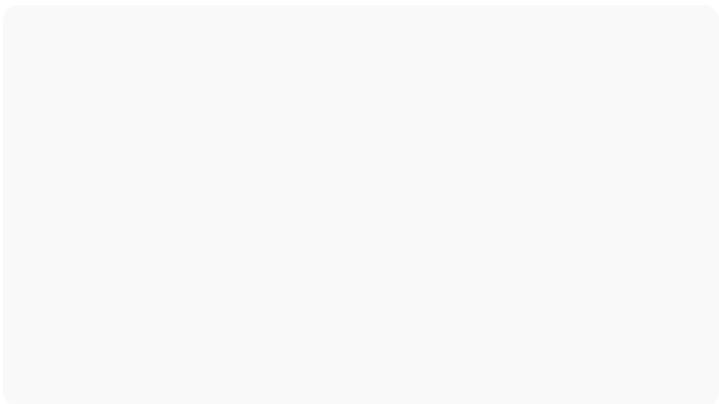
- Non-convexity of the Support-Finding Subproblem
- Neural Network Approximation for Support-Finding
- Challenges in Preference Learning



STA4830 (1).pdf
PDF

Read it and keep it in mind

I have read and processed the document you uploaded. Feel free to ask questions or let me know how you'd like to proceed!



rewrite the slide into a well-organized one

Here's a revised version of the slide, making it more structured and clear:

Challenges in Preference Learning Optimization

1. Parametric Bias Assumption

- The model assumes a specific distribution (such as a mixture of distributions), which can introduce bias.
- This can lead to inaccurate model predictions, especially when using methods like Expectation-Maximization (EM).

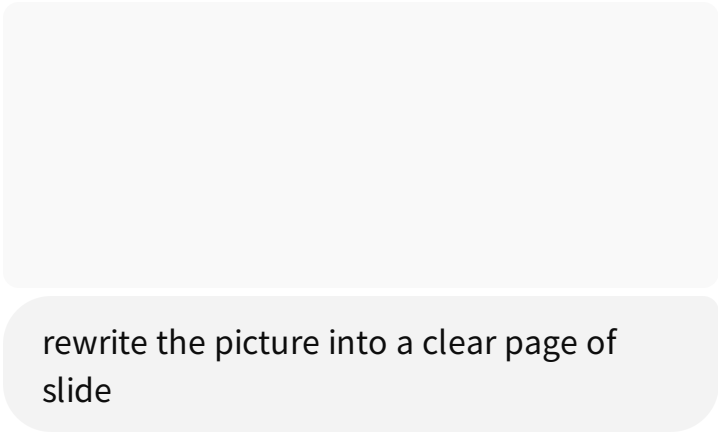
2. Projected Gradient Method

- The difficulty arises from not being able to clearly formulate the problem's constraints.
- For example, mapping weights and features to a probability space is complex and hard to describe explicitly.

3. Why Use Frank-Wolfe Algorithm?

- Due to the above challenges, we opt for the Frank-Wolfe algorithm, which addresses these issues effectively.

This organization emphasizes the challenges while providing a clear flow toward the solution. Would you like any further adjustments?



Here's a rewritten version of the content for a clearer slide format:

Non-convexity of the Support-Finding Subproblem

Optimization Problem:

At each iteration of the conditional gradient algorithm, the support-finding step reduces to the following optimization problem:

$$\min_{\omega \in \mathbb{R}^d} \langle \nabla \text{loss}(g^{(k-1)}), f(\omega) \rangle$$

- Where:
 - $f(\omega) \in \mathbb{R}^M$ is the atomic likelihood vector computed from the MNL model parameter ω .
 - Each component of $f(\omega)$ takes the form of a softmax probability.

Key Challenge - Non-convexity:

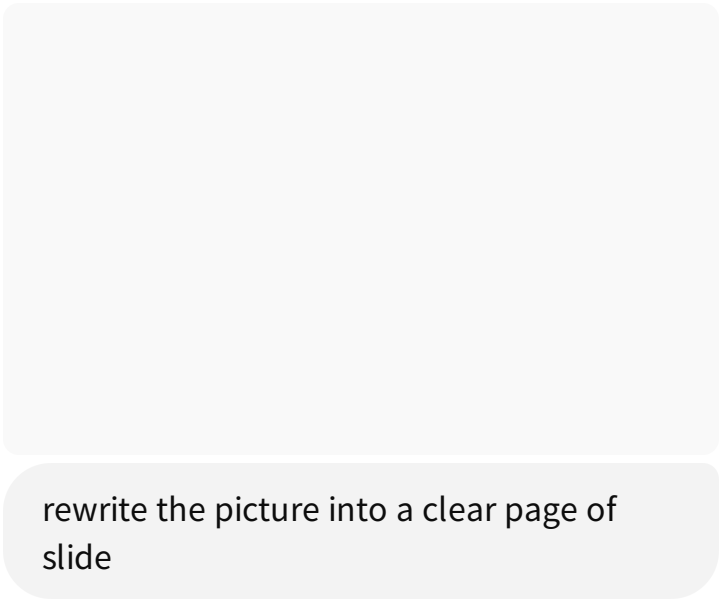
- The mapping $\omega \mapsto f(\omega)$ involves a composition of exponential and rational functions.
- This results in a highly nonlinear and non-concave structure.

- While the loss function itself is linear in f , the optimization is conducted over the nonlinear transformation of ω , making the problem overall non-convex.

Implication:

- No guarantee of global optimality.
- The problem is solved using first-order methods (e.g., BFGS), which provide only approximate descent directions.

This format organizes the content into easily digestible bullet points with a clearer structure for a slide presentation. Would you like any adjustments?



Here is a rewritten, more organized version for a clear slide format:

Neural Network Approximation for Support-Finding

Optimization Problem:

At each iteration of the conditional gradient algorithm, the support-finding step requires solving the following optimization problem:

$$\min_{\omega \in \mathbb{R}^d} \langle \nabla \text{loss}(g^{(k-1)}), f(\omega) \rangle$$

Where $f(\omega) \in \mathbb{R}^M$ is the atomic likelihood vector produced by the multinomial logit (MNL) model with parameter ω . This problem is non-convex and may be difficult to solve directly, especially in high-dimensional settings or with complex feature transformations.

Solution - Using Neural Networks:

We propose using a two-layer neural network to approximate the atomic likelihood function $f(\omega)$. The choice probabilities are

parameterized as:

$$f_{\theta}(x) = \text{softmax}(W_2 \cdot \sigma(W_1 x + b_1) + b_2)$$

- Where:
 - $\theta = \{W_1, W_2, b_1, b_2\}$ are the learnable parameters of the network.
 - $\sigma(\cdot)$ is a nonlinear activation function (e.g., ReLU or tanh).

Optimization:

Instead of solving for the optimal logit parameter ω , we optimize the neural network parameters θ to minimize the same linear objective:

$$\min_{\theta} \langle \nabla \text{loss}(g^{(k-1)}), f_{\theta}(x) \rangle$$

Advantages:

- This approach allows for greater expressiveness, capturing more complex patterns in the atomic likelihood structure beyond the linear logit formulation.
- It enables the use of modern stochastic gradient optimization techniques for efficient approximation in large-scale settings.

This slide makes the content clearer and breaks down the complex information into easy-to-digest points. Let me know if you'd like any adjustments!



rewrite the picture into a clear page of slide

Here is the content rewritten for a clearer, well-structured slide:

Challenges in Preference Learning

1. Limited Data Size

- The Expedia dataset used in our experiments contains a limited number of training examples.
- Neural networks typically require large-scale data to generalize well.
- A lack of sufficient data can result in poor generalization, even with strong regularization.

2. Under-regularized Proportions

- Although regularization was applied to the support-finding step, we didn't penalize model complexity during the proportion update step.
- This can lead to overfitting, where complex types receive large weights in the mixture even if they overfit the training data.
- Future work may include sparsity-inducing penalties or entropy regularization in the weight optimization process.

3. Inherent Linearity of Human Decision Boundaries

- Human preferences in real-world decision-making (e.g., hotel booking) often follow relatively linear structures.
- Customers tend to prefer cheaper hotels, higher star ratings, or prominent locations.
- In such scenarios, linear models may already capture most of the signal, and introducing nonlinearity could add noise and increase the risk of overfitting.

This organization simplifies the content into clearly defined points that are easy to present in a slide format. Let me know if you'd like further revisions!

RLHF是什么

RLHF stands for Reinforcement Learning with Human Feedback. It is a technique used to train machine learning models, particularly in reinforcement learning (RL), where human feedback is used to guide the learning process.

Here's how it works:

- In traditional reinforcement learning, agents learn by interacting with an environment and receiving rewards or penalties based

on their actions.

- In RLHF, the agent receives feedback not just from the environment but also from humans, who can provide additional signals, corrections, or preferences to help improve the agent's behavior.
- This human feedback can be in the form of ratings, preferences, or demonstrations.

RLHF is particularly useful in tasks where it is difficult to specify an exact reward function or where human intuition and preferences are crucial for guiding the learning process, such as in natural language processing or robotics. It has been used, for example, in training large language models like GPT to improve responses based on human preferences.