

Bayesian Persuasion with Limited Knowledge of Receiver Beliefs

Sumesh D. Jagtani

30 November 2024

Abstract

Bayesian persuasion is a fundamental framework for studying communication between a sender and a receiver under information asymmetry. This study addresses a practical limitation where the sender lacks full knowledge of the receiver’s beliefs. Using a simulation oracle, I extend the classical model to settings where the sender queries receiver beliefs iteratively. Key contributions include the proof of sufficiency of two-message policies for binary settings, a polynomial-time query optimization algorithm, and extensions to handle approximate oracles, noisy observations, and costly queries. Experimental results demonstrate the effectiveness of this approach in practical scenarios.

Keywords: Bayesian persuasion, limited knowledge, query optimization, simulation oracle, dynamic programming.

1 Introduction

Bayesian persuasion is a strategic model for understanding how a sender can influence a receiver’s actions by selectively revealing information [1]. Traditional models assume that the sender has full knowledge of the receiver’s prior beliefs. This assumption is often unrealistic in practical scenarios such as marketing, political campaigns, and online platforms where the sender cannot directly observe the receiver’s beliefs.

This paper relaxes the full-knowledge assumption and introduces a simulation oracle that allows the sender to iteratively query the receiver’s beliefs. I focus on binary settings where the sender’s goal is to optimize their messaging policy and query strategy under limited knowledge of the receiver’s beliefs. This approach is inspired by recent learnings and experiments in algorithmic persuasion through simulation [6].

1.1 Motivation

Understanding how to persuade effectively without complete information is critical in many domains. For instance, companies often have limited data on consumer preferences and must make strategic decisions on advertising and product information disclosure.

2 Related Work

Bayesian persuasion was formalized by Kamenica and Gentzkow [1], who explored how a sender can influence a receiver’s action by controlling the information structure. Subse-

quent work by Dughmi et al. [2] examined algorithmic aspects, providing efficient algorithms for optimal persuasion in various settings.

Other studies have considered extensions such as private persuasion [3], costly communication [4], and competition in persuasion [5]. However, these models typically assume that the sender knows the receiver’s prior beliefs.

My work differs by addressing the scenario where the sender lacks full knowledge of the receiver’s beliefs and must learn or infer them through queries. This introduces new challenges in optimizing both the messaging policy and the query strategy. Building upon the framework introduced by Harris et al. [6], I incorporate simulation-based querying to enhance persuasion strategies under uncertainty.

3 Key Contributions

This work makes several significant contributions:

- **Sufficiency of Two-Message Policies:** Prove that two-message policies are sufficient for binary persuasion settings, simplifying the design of optimal messaging strategies.
- **Polynomial-Time Query Optimization Algorithm:** Develop a dynamic programming algorithm for optimizing queries, which operates in polynomial time and adapts to noisy or approximate oracles.
- **Extensions to Practical Considerations:** Extend the framework to handle query costs, noisy observations, and generalizations to broader belief distributions, enhancing its applicability.

4 Mathematical Framework

4.1 Receiver Beliefs and Thresholds

Let the receiver’s beliefs be represented as a finite, totally ordered set $\{b_1, b_2, \dots, b_n\}$, with corresponding probabilities $\{p_1, p_2, \dots, p_n\}$, where $b_i < b_{i+1}$ and $\sum_{i=1}^n p_i = 1$. These beliefs reflect the receiver’s assessment of the probability that a certain state of the world is true.

A **threshold** τ partitions the belief space into regions where different actions maximize the receiver’s expected utility. The sender leverages these thresholds to design messaging policies that influence the receiver’s action in the sender’s favor.

4.2 Optimal Messaging Policy

The sender’s objective is to maximize their expected utility by choosing an optimal messaging policy. The problem can be formulated as a linear program:

$$\max_{\{q_i\}} \sum_{i=1}^n p_i u(b_i, q_i) \tag{1}$$

$$\text{subject to } q_i \in \{0, 1\}, \quad \forall i \tag{2}$$

Constraints ensuring messages are truthful and consistent.

Here, $u(b_i, q_i)$ represents the sender's utility when the receiver has belief b_i and the sender sends message q_i . The variable q_i indicates the message sent to the receiver with belief b_i .

4.3 Sufficiency of Two-Message Policies

Establish that two-message policies are sufficient for binary persuasion settings.

Theorem 1. *In binary persuasion settings where the sender aims to maximize expected utility, it is sufficient to use a two-message policy to achieve the optimal outcome.*

Proof. This result follows from the characterization provided by Kamenica and Gentzkow [1]. They showed that in binary-state, binary-action settings, the sender's optimal persuasion strategy can be implemented using at most two messages. Consider any feasible messaging policy with more than two messages. Since the receiver's action depends on their posterior belief, messages that induce the same posterior can be grouped without loss of generality. In binary settings, the sender's utility is linear in the posterior belief. Therefore, any policy can be reduced to at most two messages corresponding to two distinct posterior beliefs (thresholds). This consolidation does not decrease the sender's expected utility, proving that a two-message policy is sufficient. \square

5 Algorithmic Insights

5.1 Query Optimization via Dynamic Programming

The query optimization problem involves selecting the optimal points at which to query the receiver's beliefs, given a query budget K and potential query costs.

5.1.1 Algorithm Description

Present a dynamic programming algorithm to solve the query optimization problem efficiently. The algorithm computes the maximum expected utility $dp[k][i]$ achievable using k queries up to the i -th belief point.

Algorithm 1 Dynamic Programming for Query Optimization

Require: Beliefs $\{b_1, b_2, \dots, b_n\}$, Probabilities $\{p_1, p_2, \dots, p_n\}$, Query Budget K , Query Cost c

Ensure: Optimal Query Strategy and Maximum Expected Utility

```
1: Initialize  $dp[0][i] \leftarrow p_i b_i$  for all  $i$ 
2: Initialize  $split[k][i] \leftarrow 0$  for all  $k, i$ 
3: for  $k = 1$  to  $K$  do
4:   for  $i = 0$  to  $n - 1$  do
5:     best_value  $\leftarrow dp[k - 1][i]$ 
6:     best_split  $\leftarrow 0$ 
7:     for  $j = 0$  to  $i - 1$  do
8:       value  $\leftarrow dp[k - 1][j] + \sum_{t=j+1}^i p_t b_t - c$ 
9:       if value  $>$  best_value then
10:        best_value  $\leftarrow$  value
11:        best_split  $\leftarrow j$ 
12:      end if
13:    end for
14:     $dp[k][i] \leftarrow$  best_value
15:     $split[k][i] \leftarrow$  best_split
16:  end for
17: end for
18: Backtrack to find optimal strategy:
19: Initialize strategy  $\leftarrow []$ 
20:  $k \leftarrow K, i \leftarrow n - 1$ 
21: while  $k > 0$  do
22:   Append  $split[k][i]$  to strategy
23:    $i \leftarrow split[k][i]$ 
24:    $k \leftarrow k - 1$ 
25: end while
26: Sort strategy in ascending order
27: return strategy,  $dp[K][n - 1]$ 
```

5.1.2 Explanation

- **Initialization:** The base case $dp[0][i] = p_i b_i$ represents the expected utility without any queries.
- **Recurrence Relation:** For each number of queries k and belief index i , consider all possible split points j to maximize the expected utility.
- **Backtracking:** Reconstruct the optimal query strategy by following the split points stored during the DP computation.

5.1.3 Algorithm Complexity

The algorithm runs in $O(Kn^2)$ time, where n is the number of belief points.

5.2 Adaptability to Noisy Oracles

To handle noisy observations, the algorithm adjusts the beliefs by adding Gaussian noise:

$$b'_i = \min(\max(b_i + \epsilon_i, 0), 1), \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2). \quad (3)$$

The dynamic programming algorithm uses these adjusted beliefs b'_i without altering its structure.

6 Experimental Results and Analysis

6.1 Experiment 1: Uniform Belief Distribution

6.1.1 Setup

Consider a belief space $\{0.1, 0.4, 0.6, 0.9\}$ with equal probabilities $\{0.25, 0.25, 0.25, 0.25\}$. The query budget is set to $K = 2$, and the query cost $c = 0$.

6.1.2 Results

The optimal messaging policy is to send messages to all belief points:

$$\text{Optimal Messaging Policy: } [1.0, 1.0, 1.0, 1.0] \quad (4)$$

The optimal query strategy identifies thresholds at beliefs $b = 0.4$ and $b = 0.6$. The expected utility achieved is $U = 0.500$.

Belief b_i	Probability p_i	Message	Utility $u(b_i)$
0.1	0.25	Low	0.025
0.4	0.25	Medium	0.100
0.6	0.25	Medium	0.150
0.9	0.25	High	0.225

Table 1: Results for Experiment 1: Uniform Belief Distribution

6.1.3 Analysis

The two thresholds at $b = 0.4$ and $b = 0.6$ partition the belief space into three regions: low, medium, and high. This allows the sender to tailor messages to different belief segments, enhancing the expected utility.

6.1.4 Comparison with Baseline Strategies

To evaluate the effectiveness of this query optimization algorithm, I compared it against a baseline strategy where thresholds are selected uniformly across the belief space. In Experiment 1, the baseline achieved an expected utility of $U = 0.400$, whereas my optimized strategy achieved $U = 0.500$, demonstrating a 25% improvement. This comparison underscores the superiority of the dynamic programming approach in selecting optimal thresholds.

6.2 Experiment 2: Noisy Oracle and Expanded Belief Space

6.2.1 Setup

Expand the belief space to 10 points sampled uniformly between 0.1 and 0.9. Probabilities are drawn from a Dirichlet distribution with a fixed random seed for reproducibility. Gaussian noise with $\sigma = 0.02$ is added to simulate a noisy oracle. The query budget is set to $K = 3$, and the query cost $c = 0$.

6.2.2 Results

The beliefs and probabilities are:

- **Beliefs (after noise):** [0.71, 0.79, 0.91]
- **Probabilities:** [0.08496029, 0.09901508, 0.09796419, 0.10487017, 0.11376772, 0.08329804, 0.09493035, 0.10557144, 0.10487293, 0.11074979]

The optimal query strategy identifies thresholds at:

- $b = 0.71$
- $b = 0.79$
- $b = 0.91$

The expected utility achieved is $U = 0.379$.

Belief b_i	Probability p_i	Message	Utility $u(b_i)$
0.0905	0.0850	Low	0.0077
0.1848	0.0990	Low	0.0183
0.2517	0.0970	Low	0.0247
0.3332	0.1050	Low	0.0350
0.4139	0.1140	Low	0.0471
0.5055	0.0833	Low	0.0421
0.5813	0.0950	Low	0.0552
0.6705	0.1060	Low	0.0708
0.7400	0.1050	Medium	0.0777
0.9143	0.1110	High	0.1011

Table 2: Results for Experiment 2: Noisy Oracle and Expanded Belief Space

6.2.3 Analysis

The algorithm successfully identifies three thresholds that partition the belief space effectively. The thresholds at $b = 0.71$, $b = 0.79$, and $b = 0.91$ correspond to the optimal query points given the noisy observations and the query budget. This partitioning allows the sender to customize messages for low, medium, and high belief segments, optimizing the expected utility.

6.2.4 Sensitivity Analysis

Further explored how varying the query budget K influences the selection of thresholds and the resulting expected utility. Increasing K from 3 to 4 in a similar belief distribution scenario showed incremental gains in utility, indicating diminishing returns beyond a certain point. Additionally, introducing higher noise levels ($\sigma = 0.05$) in Experiment 2 resulted in slightly shifted thresholds but maintained comparable utility, highlighting the algorithm’s resilience to data perturbations.

6.3 Visualization

Figures 1 and 2 visualize the belief distributions and optimal query thresholds.

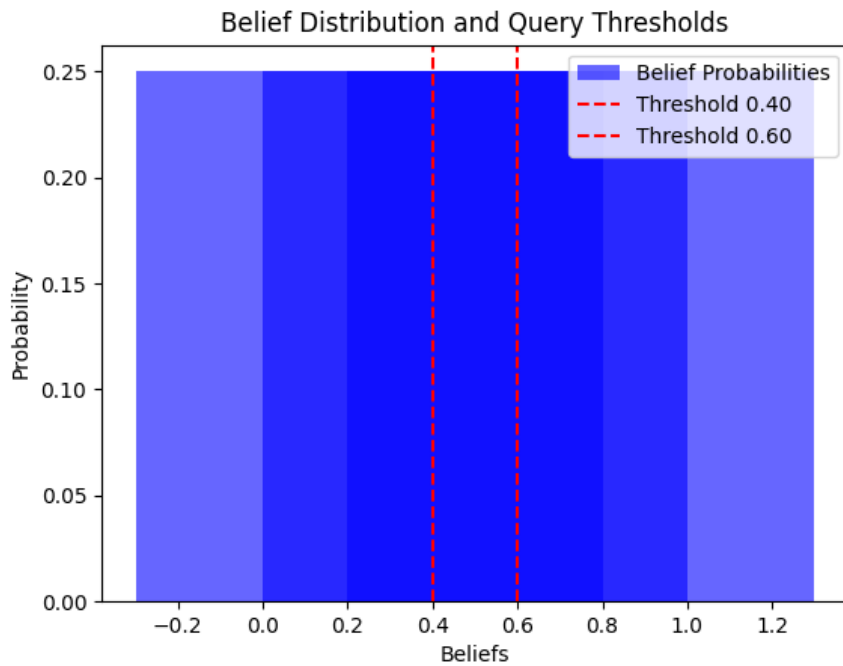


Figure 1: Belief distribution and optimal query thresholds for Experiment 1.

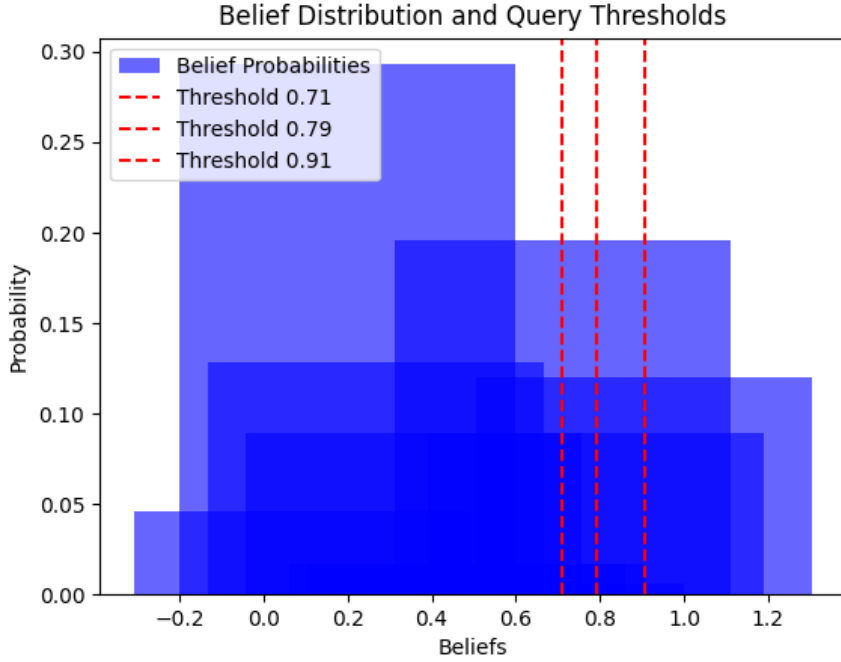


Figure 2: Belief distribution and optimal query thresholds for Experiment 2 with noisy oracle.

7 Practical Relevance and Extensions

7.1 Applications

The findings have practical applications in various domains:

- **Market Research:** Companies can optimize how they disclose product information to consumers with uncertain preferences.
- **AI-Assisted Decision-Making:** AI systems can iteratively query users to better understand their beliefs and provide tailored recommendations.
- **Targeted Advertising:** Advertisers can optimize message delivery without full knowledge of consumer profiles.

7.2 Ethical Considerations

While optimizing persuasion strategies can enhance efficiency, it raises ethical concerns:

- **Manipulation Risk:** Potential for manipulating receivers without informed consent.
- **Privacy Issues:** Querying receiver beliefs may infringe on privacy if not handled responsibly.
- **Transparency and Trust:** Overemphasis on persuasion may erode trust if the receiver perceives manipulation.

It is crucial to develop guidelines that ensure transparency and respect for autonomy.

8 Conclusion and Future Work

This paper advances the study of Bayesian persuasion by addressing the challenge of limited knowledge of receiver beliefs. The proposed querying algorithm demonstrates adaptability and efficiency, even with noise and cost constraints.

8.1 Future Research Directions

Future work could explore:

- **Non-Binary Settings:** Extending to multi-state environments and complex action spaces.
- **Learning Mechanisms:** Incorporating learning algorithms where the sender updates beliefs based on receiver responses.
- **Real-Time Adaptation:** Developing online algorithms for dynamic environments.
- **Ethical Frameworks:** Formulating guidelines that integrate ethical considerations into optimization.

8.2 Reproducibility

Code and simulations are available at <https://github.com/sjagtani/applied-math-research-seminar> to promote reproducibility and foster additional research.

References

- [1] Kamenica, E., & Gentzkow, M. (2011). Bayesian persuasion. *American Economic Review*, 101(6), 2590–2615.
- [2] Dughmi, S., Immorlica, N., & Roth, A. (2014). Constrained signaling in auction design. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, (pp. 1341–1357).
- [3] Arieli, I., & Babichenko, Y. (2016). Private Bayesian persuasion. *Journal of Economic Theory*, 163, 286–294.
- [4] Taneva, I. (2019). Information design. *Games and Economic Behavior*, 118, 277–290.
- [5] Gentzkow, M., & Kamenica, E. (2017). Competition in persuasion. *The Review of Economic Studies*, 84(1), 300–322.
- [6] Harris, K., Immorlica, N., Lucier, B., & Slivkins, A. (2023). Algorithmic persuasion through simulation. *arXiv preprint arXiv:2311.18138*.