École polytechnique fédérale de Lausanne

Master thesis spring 2024

Master in Applied Mathematics

APPLICATION OF THE INFORMATION-THEORETIC CLUSTERING TO THE DISCRETE CHOICE

Author:
Alexandre Monti

Supervisors:
Pavel Ilinov
Evangelos Paschalidis
Michel Bierlaire

Abstract

This thesis investigates the application of the Deterministic Information Bottleneck (DIB) algorithm to discrete choice models, aiming to establish a link between these two methodologies to enhance model selection and evaluation. Discrete choice models are essential for understanding decision-making processes, yet they often face challenges in model selection due to high-dimensional data and complex preference structures. The DIB algorithm, which optimizes the trade-off between relevance and compression of information, offers a promising approach to address these issues.

In this study, we demonstrate that the DIB algorithm can identify the best-fitting discrete choice model by maximizing log-likelihood and minimizing the Akaike Information Criterion (AIC). By integrating DIB with discrete choice theory, we develop a novel framework for model selection that leverages information-theoretic principles to improve predictive performance and interpretability.

To validate our approach, we apply the proposed method to different datasets. The empirical results show that models selected using the DIB algorithm consistently outperform other possible models in terms of log-likelihood and AIC. These findings support our conjecture that DIB can serve as a powerful tool for model selection in discrete choice analysis, providing more accurate and reliable insights into consumer behavior.

Overall, this research contributes to the field of discrete choice modelling by introducing an information-theoretic perspective, which enhances both the theoretical understanding and practical application of these models.

Contents

1 Introduction

1.1 Basics of utility theory

Utility theory provides a structured approach to understanding how individuals make decisions when faced with multiple alternatives. It is grounded in the concept that individuals seek to maximize their utility, a representation of the choice procedure they follow. Utility theory employs utility functions to assign numerical values to potential outcomes, reflecting individuals' preferences and guiding their decision-making process. These functions encapsulate preferences and attitudes, thereby rationalizing the selection of the most preferred option among various choices.

A random utility model is based on a utility function for an alternative i which is defined as the sum of two components, a deterministic part and a random component:

$$U_i = V_i + \epsilon_i$$

V_i captures the systematic aspects of an individual's preferences or the intrinsic value associated with a particular choice. It encompasses factors such as the desirability of an option, its tangible attributes such as cost, and any other deterministic influences that contribute to the individual's overall satisfaction or utility.

The term ϵ_i represents the random component or stochastic element in the utility function. This term accounts for the unobservable factors that influence decision-making, which the analyst cannot capture or predict. It introduces variability into the utility function, reflecting the uncertainty and unpredictability in modeling human choice behavior.

1.1.1 Logit model

The logit model [5] is widely recognized in discrete choice analysis for its straightforward mathematical framework, clear interpretation of parameters, and adaptability to different choice scenarios. With its parameters offering practical insights into the significance of various attributes affecting decision-making, the model aids in informed choices.

Definition: Let C be a choice set. Assume that the random components of the alternatives ϵ_i are i.i.d. EV(η, μ). The logit model is derived as

$$P(i \mid C) = \frac{e^{\mu V_i}}{\sum_{j \in C} e^{\mu V_j}}$$

In general, $V_i = \sum_k \beta_k z_{ik}$ denoting a linear combination of the vector of attributes z_i for alternative i but the utility function is not necessarily linear in parameters. The β_k coefficients serve as model parameters that necessitate estimation using data. While η is typically a scale parameter in the extreme value distribution, setting it to 0 simplifies the model, assuming identical and independent distribution of the random components. The decision-making process involves a fixed number of alternatives in the set C, where the decision-maker selects the alternative with the highest utility.

1.1.2 Nested logit model

In many decision-making scenarios, the simple logit model proves insufficient due to its property of independence from irrelevant alternatives (IIA), which emerges from the underlying theory of the logit model. This property implies that the relative odds of choosing between any two alternatives are unaffected by the presence of other alternatives. To illustrate this, consider a scenario with three transportation options: car, red bus, and blue bus. If a person initially chooses between a car and a red bus, the introduction of a blue bus should theoretically not influence their decision if the red and blue buses are similar. However, the logit model fails to account for this correlation, leading to potentially inaccurate predictions.

To address this limitation, we turn to the nested logit model [5]. This model groups similar alternatives into nests, allowing for correlated unobserved factors within each nest. For instance, in our example, we can create two nests: "private transport" for the car and "public transport" for the red and blue buses. By structuring the model in this way, we account for the similarity between the red and blue buses, as they are more likely to be substitutes for each other than for the car. Thus, the nested logit model provides a more realistic framework for modeling choices when alternatives share similarities, overcoming the constraints of the simple logit model.

Definition: Let C be the choice set and $C_1, ..., C_M$ a partition of C. Again, assume that the random components of the alternatives ϵ_i are i.i.d. EV(η, μ). The nested logit model is derived as

$$P(i \mid C) = \sum_{i=1}^{M} P(i \mid m, C) P(m \mid C)$$

In the nested model, each alternative i belongs to exactly one nest m which leads to

$$P(i \mid C) = P(i \mid m, C) P(m \mid C)$$

Property: Each nest m is associated with a scale parameter μ_m and an expected maximum utility given by

$$\tilde{V}_m = \frac{1}{\mu_m} \ln \left[\sum_{i \in C_m} e^{\mu_m V_i} \right]$$

Comments:

1. The ratio μ/μ_m must be estimated from data and needs to be in $[0, 1]$.
2. Going down the nests, μ's must increase and variance must decrease.

With this definition, we can compute the general formulas for the probability of an alternative within its nest and the probability of a particular nest:

$$P(i \mid m) = \frac{e^{\mu_m V_i}}{\sum_{j \in C_m} e^{\mu_m V_j}}$$
$$P(m \mid C) = \frac{e^{\mu \tilde{V}_m}}{\sum_{p=1}^{M} e^{\mu \tilde{V}_p}}$$

In general, μ can be normalized to 1 which simplifies the computations of the above probabilities.

1.1.3 Cross-nested logit model

We might consider the following question: what if an alternative could be part of more than one nest? For instance, imagine we have five alternatives: car, bus, train, walking, and cycling, and we wish to establish two nests: motorized and private. In this scenario, car, bus, and train would be in the "motorized" nest, while car, walking, and cycling would be in the "private" nest. However, this situation does not adhere to the structure of a nested logit model, as the alternative "car" is included in both nests. Consequently, we must create a new type of model capable of accommodating this requirement.

Definition: Let C be the choice set and $C_1, ..., C_M$ groups of alternatives which may overlap. Each group is associated with a scale parameter μ_m with the constraint that $\mu/\mu_m \in [0, 1]$ and for each alternative i in nest m we define a degree of membership $\alpha_{im} \in [0, 1]$ with the following constraints:

- Each alternative must belong to at least one nest: for all j, there exists m such that $\alpha_{jm} > 0$
- $\sum_m \alpha_{jm} = 1$ for all j

The cross-nested logit model [5] is derived as

$$P(i \mid C) = \sum_{i=1}^{M} P(i \mid m, C) P(m \mid C)$$

where

$$P(i \mid m, C) = \frac{\alpha_{im}^{\mu_m/\mu} e^{\mu_m V_i}}{\sum_{j \in C} \alpha_{jm}^{\mu_m/\mu} e^{\mu_m V_j}} \quad \text{and} \quad P(m \mid C) = \frac{\left(\sum_{j \in C} \alpha_{jm}^{\mu_m/\mu} e^{\mu_m V_j}\right)^{\frac{\mu}{\mu_m}}}{\sum_{l=1}^{M} \left(\sum_{j \in C} \alpha_{jl}^{\mu_l/\mu} e^{\mu_l V_j}\right)^{\frac{\mu}{\mu_l}}}$$

We can easily see that the cross-nested model is a generalization of the nested model. Indeed, we can define a nested model from this definition by setting $\alpha_{im} = \mathbb{1}_{C_m}(i)$ where $\mathbb{1}_{C_m}(i) = 1$ if $i \in C_m$ and 0 otherwise.

The nest structure in nested and cross-nested logit models must be specified and assumed before modeling. However, in practice, determining the most suitable nest structure can be a challenging task, primarily because we lack access to the true model. Various factors, such as the context, the characteristics of the alternatives, and the available data, can influence the selection of the nest structure.

Moreover, the absence of clear guidelines or methodologies for nest identification further complicates the process.

The identification issue surrounding nest structures serves as a compelling motivation for our study. By investigating methods for effectively identifying nests in nested and cross-nested logit models, we aim to enhance the applicability and robustness of these models in decision-making contexts. Through empirical analysis and theoretical insights, we seek to contribute to the ongoing research on discrete choice modelling and provide guidance for researchers and practitioners grappling with the complexities of choice behavior analysis.

1.2 Basics of information theory

This chapter explores fundamental concepts [4] such as entropy, mutual information, and Kullback-Leibler divergence, essential for understanding the following of this work on Information Bottleneck framework.

Definition: Let X be a discrete random variable taking values in A_X with distribution p. The entropy of X is given by

$$H(X) = \sum_{x \in A_X} p(x) \cdot \log\left(\frac{1}{p(x)}\right)$$

with the convention that if $p(x) = 0$, then $0 \cdot \log\left(\frac{1}{0}\right) = 0$. When we take base 2 logarithm, the unit for the entropy is the bit.

The entropy captures the average information content (or uncertainty) associated with the outcomes of a random variable. The entropy is maximum when the distribution is uniform and is zero if an outcome happens with probability 1.

Example: To illustrate the different notions discussed in this chapter, we will take the same example for each concept. Let's consider a movie recommendation system that suggests movies to users based on their past viewing history and ratings. In this context, we can explore the uncertainty involved in predicting a user's movie rating. Denoting by R the ratings and $p(r)$ the probability distribution of the ratings, we delve into how uncertainty manifests in this scenario.

$$H(R) = \sum_{r} p(r) \cdot \log(1/p(r))$$

If the rating system is on a scale of 1 to 5, and all ratings are equally likely, the entropy would be highest because there is maximum uncertainty about what rating a user might give:

$$H(R) = \sum_{1}^{5} \frac{1}{5} \cdot \log(5) = \log(5)$$

On the other hand, if we know that one user always gives 5, the entropy would be minimal because there is no uncertainty about the rating:

$$H(R) = 1 \cdot \log(1) = 0$$

Definition: Let X and Y be two discrete random variables taking values in A_X and A_Y respectively and with joint distribution p. The joint entropy of X and Y is given by

$$H(X, Y) = \sum_{x \in A_X, y \in A_Y} p(x, y) \cdot \log\left(\frac{1}{p(x, y)}\right)$$

One property of the joint entropy is that $H(X) + H(Y) \geq H(X, Y)$ with equality if and only if $X \perp\!\!\!\perp Y$. The intuition behind the joint entropy is closely similar to the one for the entropy. It gives the uncertainty associated with the outcomes of a random vector.

Example: Continuing with our movie recommendation system example, let's introduce another variable, D, representing the time of day when the user watches the movie. Now, $H(R, D)$ can be seen as the joint uncertainty of a user's movie rating and the time of day when they watch the movie. For instance, if a user tends to rate movies highly in the evening but has varying preferences during other times of the day, the joint entropy captures the combined uncertainty of predicting both the user's rating and the time of day when they watch the movie.

Definition: The conditional entropy of X given Y is

$$H(X \mid Y) = \sum_{y \in A_Y} p(y) \left[\sum_{x \in A_X} p(x \mid y) \log\left(\frac{1}{p(x \mid y)}\right)\right] = \sum_{x \in A_X, y \in A_Y} p(x, y) \log\left(\frac{1}{p(x \mid y)}\right)$$

The conditional entropy quantifies the average uncertainty about x when y is known.

Example: Again with our movie recommendation system, we define G as the genre of the movie. Suppose that one user gives only good rates to thrillers and bad rates to any other genre. In this case, $H(R \mid G)$ will be low as there is little uncertainty about the grade that this user will give depending on the genre. On the other hand, if a user likes many genres and gives good rates to them, $H(R \mid G)$ will be high because knowing the genre does not reduce the uncertainty about the rate that the user will give.

We can link these different definitions together by the following property:

$$H(X, Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y)$$

Furthermore, $H(X \mid Y) \leq H(X)$.

We could ask ourselves: is there a way to measure how much information I can learn about one variable by observing the other? This is exactly the question that mutual information answers.

Definition: The mutual information between X and Y is given by

$$I(X; Y) = H(X) - H(X \mid Y)$$

With the properties about conditional and joint entropy from above, we know that $I(X;Y) \geq 0$ and $I(X;Y) = I(Y;X)$. Indeed,

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X) + H(Y) - H(X) - H(Y \mid X) \\ &= H(Y) - H(Y \mid X) = I(Y;X) \end{aligned}$$

Example: In our example, $I(R;G)$ indicates how much knowing the genre of a movie reduces the uncertainty of predicting a user's rating.

Remark: We can generalize all of the above to continuous random variables easily by replacing sums with integrals. In the continuous case, the entropy can be infinitely large and positive or negative.

Another interesting property of the mutual information is that we can directly link it to the Kullback-Leibler divergence. We first recall what the Kullback-Leibler divergence is:

Definition: Let p and q be two discrete probability distributions defined on the same sample space X. The Kullback-Leibler divergence (or relative entropy) from q to p is defined by

$$D_{KL}(p \mid\mid q) = \sum_{x \in X} p(x) \log\left(\frac{p(x)}{q(x)}\right)$$

The Kullback-Leibler divergence can be seen as a representation of the distance between two distributions p and q.

Now, if we go back to the definition of mutual information, we have that

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= \sum_{x \in A_X} p(x) \cdot \log\left(\frac{1}{p(x)}\right) - \sum_{x \in A_X, y \in A_Y} p(x,y) \log\left(\frac{1}{p(x \mid y)}\right) \\ &= \sum_{x \in A_X, y \in A_Y} p(x,y) \cdot \log\left(\frac{1}{p(x)}\right) - \sum_{x \in A_X, y \in A_Y} p(x,y) \log\left(\frac{1}{p(x \mid y)}\right) \\ &= \sum_{x \in A_X, y \in A_Y} p(x,y) \cdot \left[\log\left(\frac{1}{p(x)}\right) - \log\left(\frac{1}{p(x \mid y)}\right)\right] \\ &= \sum_{x \in A_X, y \in A_Y} p(x,y) \cdot \log\left(\frac{p(x \mid y)}{p(x)}\right) \\ &= \sum_{x \in A_X, y \in A_Y} p(x,y) \cdot \log\left(\frac{p(x,y)}{p(x) \cdot p(y)}\right) \\ &= D_{KL}(p(x,y) \mid\mid p(x) \cdot p(y)) \end{aligned}$$

Thus, we have that $I(X,Y) = D_{KL}(p(x,y) \mid\mid p(x) \cdot p(y))$. Informally, the mutual information quantifies how much the actual relationship between X and Y deviates from a scenario where X and Y are completely independent. Indeed, if $D_{KL}(p(x,y) \mid\mid p(x) \cdot p(y)) = 0$, it means that $p(x,y) = p(x) \cdot p(y)$ which is equivalent to $X \perp\!\!\!\perp Y$.

Example: Again with the same example, $D_{KL}(p(r,g) \mid\mid p(r) \cdot p(g))$ measures how much the joint distribution of rates and genres is close to the scenario where rates and genres are independent. If $D_{KL} = 0$, it means that rates and genres are independent so knowing the genre of a movie will not reduce the uncertainty of predicting a user's rating, i.e. $I(R;G) = 0$. On the other hand, if D_{KL} is high, it means that rates and genres are dependent and thus, knowing the genre of a movie will reduce the uncertainty of predicting a user's rating, i.e. $I(R;G) > 0$.

With these few definitions, we are now ready to delve into the information-theoretic clustering methods that will be the main topic of this work.

1.3 Information Bottleneck method (IB)

Developed by Naftali Tishby, Fernando C. Pereira, and William Bialek in 1999 [1], the Information Bottleneck method is a powerful framework in machine learning and information theory that aims to extract important information from data while compressing any irrelevant or duplicated elements. More precisely, the goal of this method is to find a compressed representation of the input data X while preserving the relevant information that X gives about the output data Y. In this setting, we represent the compressed representation of X by T. The three variables T, X and Y are random variables.

To find this compressed representation T, we need to solve the following optimization problem:

Given the joint distribution $p(x,y)$, the optimized encoding distribution $q(t|x)$ corresponds to

$$\begin{aligned}\min_{q(t|x)} L[q(t|x)] &= I(X;T) - \beta I(T;Y) \\ &= \sum_{x,t} q(x,t) \cdot \log\left(\frac{q(x,t)}{p(x)q(t)}\right) - \beta \sum_{t,y} q(t,y) \cdot \log\left(\frac{q(t,y)}{q(t)p(y)}\right) \\ &= \sum_{x,t} q(t|x)p(x) \cdot \log\left(\frac{q(t|x)}{q(t)}\right) - \beta \sum_{t,y} q(y|t)q(t) \cdot \log\left(\frac{q(y|t)}{p(y)}\right)\end{aligned}$$

with the additional Markov constraint $T \longleftrightarrow X \longleftrightarrow Y$. This constraint ensures that T can only have information about Y through X, i.e. $T \perp\!\!\!\perp Y \mid X$. In terms of probability distributions, this can be written as $p(t,y \mid x) = p(t \mid x)p(y \mid x)$.

The idea behind this optimization problem is that the first term pushes for compression, while the second term emphasizes the importance of keeping relevant information. In this context, β serves as the Lagrange multiplier associated with the constraint of retaining meaningful information.

Example: In our movie rating system, we want to compress the user's viewing history X into a simpler representation T that still retains the important information for predicting the rating Y. A simple way of compress X would be to only keep the genre of the movie. In this case, we would have $|T| = |G|$, i.e. each genre would be a cluster. By doing that, we compress the information contained in X but we can still make some prediction about Y.

Let's go back to the theory behind the Information Bottleneck. We denote by p the fixed distributions and by q the distributions that we can change or choose. By making variational calculus, we find that the original optimization problem has a formal solution [1] given by:

$$\begin{aligned}
q(t|x) &= \frac{q(t)}{Z(x,\beta)} \exp[-\beta D_{\mathrm{KL}}[p(y|x) \mid\ q(y|t)]] \\
q(y|t) &= \frac{1}{q(t)} \sum_x q(t|x) p(x,y) \\
Z(x,\beta) &\equiv \exp\left[-\frac{\lambda(x)}{p(x)} - \beta \sum_y p(y|x) \log \frac{p(y|x)}{p(x)}\right]
\end{aligned}$$

where $D_{\mathrm{KL}}(P(x) \mid Q(x)) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$ denotes the Kullback-Leibler divergence. The $\lambda(x)$ in the definition of $Z(x,\beta)$ is the Lagrange multiplier for the normalization of the conditional distributions $q(t \mid x)$ at each x.

This solution is only formal because the first two equations depend on each other which makes them impossible to compute. But with an iterative approach, we can compute a solution which converges to a local minimum of the cost function [1]. This iterative algorithm works as follows:

Choose some initial distribution $q^{(0)}(t \mid x)$. Compute

$$q^{(0)}(t) = \sum_x p(x) q^{(0)}(t \mid x) \quad \text{and} \quad q^{(0)}(y \mid t) = \frac{1}{q^{(0)}(t)} \sum_x p(x,y) q^{(0)}(t \mid x)$$

The n-th iteration of the algorithm is given by:

$$\begin{aligned}
d^{(n-1)}(x,t) &\equiv D_{\mathrm{KL}}\left[p(y \mid x) \mid q^{(n-1)}(y \mid t)\right] \\
q^{(n)}(t \mid x) &= \frac{q^{(n-1)}(t)}{Z(x,\beta)} \exp\left[-\beta d^{(n-1)}(x,t)\right] \\
q^{(n)}(t) &= \sum_x p(x) q^{(n)}(t \mid x) \\
q^{(n)}(y \mid t) &= \frac{1}{q^{(n)}(t)} \sum_x q^{(n)}(t \mid x) p(x,y)
\end{aligned}$$

1.4 Deterministic Information Bottleneck (DIB)

One of the property of the IB method is that it produces soft clustering which means that a given input can have multiple outputs with given probabilities. The idea behind the Deterministic Information Bottleneck method (DIB) [2] is to produce hard clustering, i.e. one input has a unique output. To this aim, we modify the cost function of the IB method as follows:

$$\min_{q(t|x)} L[q(t|x)] = H(T) - \beta I(T;Y)$$

still with the same Markov constraint $T \longleftrightarrow X \longleftrightarrow Y$.

We can see the difference between the two methods by subtracting one to another:

$$\begin{aligned}
L_{IB} - L_{DIB} &= I(X;T) - \beta I(T;Y) - H(T) + \beta I(T;Y) \\
&= I(X;T) - H(T) \\
&= H(T) - H(T \mid X) - H(T) \\
&= -H(T \mid X)
\end{aligned}$$

This simple calculation reveals that the IB method promotes stochasticity in the encoding distribution $q(t \mid x)$. Indeed, $H(T \mid X)$ represents the stochasticity in the mapping from X to T. However, to fully understand why the DIB algorithm produces hard clustering, we first have to derive its formal solution.

In order to find a solution, we cannot use the same calculus method as for the IB problem. Rather, we define the following family of cost functions [2]:

$$L_\alpha \equiv H(T) - \alpha H(T \mid X) - \beta I(T;Y)$$

It is trivial to see that $L_{IB} = L_1$. We could say the same for $L_{DIB} = L_0$ but we will use a different strategy in order to find a solution for the DIB method. Indeed, we define the DIB solution as

$$q_{DIB}(t \mid x) = \lim_{\alpha \to 0} q_\alpha(t \mid x)$$

We can now use the same variational approach as for the IB method to find a formal solution [2] which is given by:

$$\begin{aligned}
d_\alpha(x,t) &\equiv D_{\mathrm{KL}}[p(y \mid x) \mid q_\alpha(y \mid t)] \\
l_{\alpha,\beta}(x,t) &\equiv \log(q_\alpha(t)) - \beta d_\alpha(x,t) \\
q_\alpha(t \mid x) &= \frac{1}{Z(x,\alpha,\beta)} \exp\left[\frac{1}{\alpha} l_{\alpha,\beta}(x,t)\right] \\
q_\alpha(y \mid t) &= \frac{1}{q_\alpha(t)} \sum_x q_\alpha(t \mid x) p(x,y)
\end{aligned}$$

where $Z(x,\alpha,\beta)$ is a normalization factor given by

$$\begin{aligned}
z(x,\alpha,\beta) &= \frac{1}{\alpha} - 1 + \frac{\lambda(x)}{\alpha p(x)} + \frac{\beta}{\alpha} \sum_y p(y \mid x) \log\left(\frac{p(y \mid x)}{p(y)}\right) \\
Z(x,\alpha,\beta) &= \exp[-z(x,\alpha,\beta)]
\end{aligned}$$

Now that we have a general formal solution, we can take the limit $\alpha \to 0$ to finally have a formal solution for the DIB problem. By computing the limit, we find that

$$\lim_{\alpha \to 0} q_\alpha(t \mid x) = f : X \to T$$

where

$$\begin{aligned} f(x) &\equiv t^* = \arg\max_t [l(x,t)] \\ l(x,t) &\equiv \log(q(t)) - \beta D_{\mathrm{KL}}[p(y \mid x) \mid q(y \mid t)] \end{aligned}$$

More explicitly, consider the conditional probability distribution $q_\alpha(t \mid x)$, which represents the probability of the compressed representation t given the original input x. As the parameter α approaches 0, this conditional probability distribution converges to a delta function. To understand why this happens, let's delve into the role of α in shaping the distribution.

The parameter α controls the trade-off between the fidelity of the representation t to the original input x and the compression of the information. When α is large, the algorithm allows more variability and randomness in the assignment of t to x. This means that $q_\alpha(t \mid x)$ can have a spread-out distribution, indicating that multiple values of t could reasonably represent x with different probabilities.

However, as α decreases towards 0, the algorithm increasingly prioritizes the accuracy of the representation. This prioritization leads to a sharper focus on selecting the single most informative value of t for each x. Mathematically, this sharp focus is achieved by maximizing the function $l(x,t)$. As α approaches 0, the distribution $q_\alpha(t \mid x)$ assigns almost all the probability mass to the value of t that maximizes $l(x,t)$. In this limit, $q_\alpha(t \mid x)$ becomes a delta function centered at this optimal t. This means that for a given x, there is a single t that is selected with probability 1, and all other values of t have a probability of 0.

Because the conditional distribution $q_\alpha(t \mid x)$ turns into a delta function, the assignment of x to t becomes deterministic: each x is mapped to one specific t without any uncertainty. Hence, this approach is termed the Deterministic Information Bottleneck. The encoding distribution $q_\alpha(t \mid x)$ no longer has any randomness as α approaches 0. This deterministic nature simplifies the representation and ensures that the most informative aspects of x are captured in t.

Moreover, the algorithm includes a term $\log(q(t))$ in its objective function. This term serves an important role in shaping the solution. It promotes the minimization of the number of unique values that t can take. Practically, this means that the algorithm encourages the mapping of one t to multiple different x values. By doing so, it seeks to compress the data effectively, ensuring that each t represents a cluster of x values rather than a unique individual value. This clustering effect is crucial for achieving efficient compression while retaining the most informative aspects of the original data.

Meanwhile, the second term $D_{\mathrm{KL}}[p(y \mid x) \mid q(y \mid t)]$, similar to the original IB problem, ensures that the chosen t values retain as much information from x about y as possible. The parameter β has the same role as in the original problem and controls the balance between the importance placed on compression and prediction.

As before, the solution for the DIB problem is only formal and we need to use an iterative algorithm in order to find a local minimum of the cost function. The iterative algorithm works as follows:

Choose some initial distribution for $f^{(0)}(x)$. Set

$$q^{(0)}(t) = \sum_{x:f^{(0)}(x)=t} p(x) \quad \text{and} \quad q^{(0)}(y \mid t) = \frac{\sum_{x:f^{(0)}(x)=t} p(x,y)}{\sum_{x:f^{(0)}(x)=t} p(x)}$$

Then apply the following steps until convergence:

$$\begin{aligned}
d^{(n-1)}(x,t) &\equiv D_{\mathrm{KL}}\left[p(y \mid x) \mid q^{(n-1)}(y \mid t)\right] \\
l_\beta^{(n-1)}(x,t) &\equiv \log(q^{(n-1)}(t)) - \beta d^{(n-1)}(x,t) \\
f^{(n)}(x) &= \arg\max_t [l_\beta^{(n-1)}(x,t)] \\
q^{(n)}(t \mid x) &= \delta\left(t - f^{(n)}(x)\right) \\
q^{(n)}(t) &= \sum_x p(x) q^{(n)}(t \mid x) = \sum_{x:f^{(n)}(x)=t} p(x) \\
q^{(n)}(y \mid t) &= \frac{1}{q^{(n)}(t)} \sum_x q^{(n)}(t \mid x) p(x,y) = \frac{\sum_{x:f^{(n)}(x)=t} p(x,y)}{\sum_{x:f^{(n)}(x)=t} p(x)}
\end{aligned}$$

The intuition behind this algorithm is the following:

- Calculate the Kullback-Leibler (KL) divergence between the true conditional distribution $p(y \mid x)$ and the current estimated conditional distribution $q^{(n-1)}(y \mid t)$. The KL divergence tells us how much $p(y \mid x)$ is "close" to $q^{(n-1)}(y \mid t)$. If $D_{KL} = 0$, the two distributions are the same.
- Calculate $l_\beta^{(n-1)}(x,t) = \log(q^{(n-1)}(t)) - \beta \cdot d^{(n-1)}(x,t)$, which combines compression and relevance terms using the Lagrange multiplier β.
- Assign each data point x to the cluster t that maximizes $l_\beta^{(n-1)}(x,t)$.
- Update the conditional distribution of clusters given data points $q^{(n)}(t \mid x)$ by setting it to a Dirac delta function $\delta(t - f^{(n)}(x))$.
- Update the marginal distribution of clusters $q^{(n)}(t)$ by summing over all data points assigned to each cluster.
- Update the conditional distribution of data points given clusters $q^{(n)}(y \mid t)$ by computing the ratio of the joint distribution $p(x,y)$ to the marginal distribution $p(x)$ for each cluster.

2 Implementation of the IB and DIB algorithms

2.1 DIB algorithm

Firstly, we'll focus on implementing the Deterministic Information Bottleneck method based on Algorithm 2 of [2] and specifically use it for geometric clustering. This approach will help us to understand the underlying dynamics and behavior of the algorithm.

Let's recall what is clustering from a mathematical point of view. Clustering is a fundamental concept aimed at grouping similar data points into distinct subsets or clusters. It involves partitioning a dataset X containing n data points into k clusters, where each cluster represents a subset C_i of X such that $X = \bigcup_{i=1}^{k} C_i$ and $C_i \cap C_j = \emptyset$ for $i \neq j$. The objective of clustering algorithms is to find these clusters in such a way that the data points within each cluster are more similar to each other than to those in other clusters. Mathematically, this can be formulated as minimizing an objective function, often based on distances or similarities between data points. Through clustering, complex datasets can be organized into meaningful groups, facilitating analysis, visualization, and pattern recognition.

It is important to note that while DIB provides a distributional approach to clustering, this characteristic introduces certain challenges when applying it to classical clustering problems. Specifically, the probabilistic nature of the traditional Information Bottleneck algorithm contrasts with the deterministic assignments in standard clustering algorithms like hard K-means. The DIB method mitigates this by offering a deterministic encoding distribution, yet it still poses questions about its integration and effectiveness compared to conventional techniques. Therefore, we will apply the DIB method to a classical clustering problem to assess its efficacy as a clustering technique.

We consider two dimensions data points $\{\mathbf{x}_i\}_{i=1:n}$ and we want to cluster them. Here, we need to choose what is X and what is Y. We first make the same choice as in [3] where X is the data point index i and Y is the data point location $\mathbf{x}$. Thus, we want to cluster data indices i in a way that keeps as much information about data location as possible.

We need now to choose the joint distribution $p(i, \mathbf{x}) = p(i)p(\mathbf{x} \mid i)$. It is trivial to choose $p(i) = \frac{1}{n}$ but the choice for $p(\mathbf{x} \mid i)$ is more complicated. Following [3], we take

$$p(\mathbf{x} \mid i) \propto \exp\left[-\frac{1}{2s^2} d(\mathbf{x}, \mathbf{x}_i)\right]$$

where s corresponds to the units of distance and d is a distance metric, for example the Euclidean distance $d(\mathbf{x}, \mathbf{x}_i) = ||\mathbf{x} - \mathbf{x}_i||^2$.

One important observation is that if $q^{(n)}(t) = 0$, which means that the cluster t is empty, then this cluster will remain empty until convergence of the algorithm. Indeed, $\log\left(q^{(n)}(t)\right) = -\infty \implies l_\beta^m(x,t) = -\infty$ for all $m \geq n$. Thus, the number of non-empty clusters $|T|$ can only decrease during the computation of the algorithm. We then make the choice of initializing each point as its own cluster.

After multiple attempts, a persistent issue arises: the algorithm becomes stagnant from the first step, with distributions remaining constant throughout iterations. Indeed, the objective function

$$L_{DIB} = \min_{q(t|x)} H(T) - \beta I(T;Y)$$

is non-convex, leading to the possibility of converging to a local minimum rather than the global minimum. This phenomenon is particularly evident when initializing each point as its own cluster. At the initialization step, the distribution of x (or i in the case of geometric clustering) is uniform, resulting in $q(y|t)$ being identical to $p(y|x)$ when x is assigned to t, which leads to $d^{(0)}(x,t) = 0$ when x is assigned to t and $d^{(0)}(x,t) > 0$ otherwise. Consequently, the algorithm becomes trapped in a local minimum because $\arg\max_t \left[l_\beta^{(n)}(x,t) \right] = t_x$ for all $n \geq 0$ where t_x represents the cluster to which x is initially assigned. In other words, each point remains in its own cluster throughout the computation.

To overcome the issue of being trapped in a local minimum, we apply the same methodology as in [3]: at each iteration of the algorithm, we assess whether the objective function L_{DIB} could be minimized further by merging two clusters. By doing this, we ensure that the algorithm will not be stuck at a local minimum and should almost surely reach the global minimum.

These improvements in the original algorithm lead to great results. For the first example, we generate 90 data points from three different mutivariate gaussian distributions with covariance matrix $\frac{1}{10}I_2$ and mean $(3,0)$, $(0,4)$ and $(5,5)$ respectively. We run the algorithm for 100 iterations and we vary the parameter β. The result of the clustering is shown on Figure 1 and 2 below.

	x for mean = (3;0)	**y for mean = (3;0)**	**x for mean = (0;4)**	**y for mean = (0;4)**	**x for mean = (5;5)**	**y for mean = (5;5)**
1	3.42	0.31	0.67	4.17	5.10	5.01
2	3.00	-0.39	0.47	4.04	5.36	4.67
3	3.30	0.42	-0.32	4.09	5.36	5.09
4	3.48	0.56	0.21	4.22	4.88	4.81
5	3.18	-0.21	0.02	4.18	4.90	5.01
6	2.82	-0.02	-0.19	3.90	4.68	5.14
7	2.48	0.23	0.23	4.09	4.57	5.02
8	2.97	0.37	0.43	3.98	4.48	5.22
9	2.40	0.67	0.28	3.98	5.58	4.75
10	2.45	0.03	0.07	4.04	4.84	5.08
11	3.36	0.27	-0.9	4.03	4.86	4.86
12	3.36	0.62	-0.56	4.43	4.62	4.82
13	3.14	0.29	-0.34	4.23	5.23	4.88
14	3.02	-0.21	0.02	4.03	4.91	4.93
15	3.14	0.01	0.32	3.23	4.93	5.01
16	3.26	-0.45	0.52	4.19	4.73	4.76
17	3.22	-0.39	0.78	4.25	5.11	5.18
18	2.89	-0.27	-0.81	3.77	4.84	5.09
19	2.75	0.35	0.22	4.68	4.64	4.69
20	2.65	0.56	0.29	3.56	4.99	5.29
21	2.99	0.81	-0.21	4.01	5.12	5.37
22	3.07	0.44	0.31	3.94	5.01	5.23
23	3.34	0.05	0.54	4.45	5.09	4.78
24	3.41	-0.59	0.45	4.44	4.80	5.21
25	3.21	-0.02	-0.29	4.04	4.89	4.91
26	3.28	0.32	0.34	4.11	4.79	5.24
27	3.01	0.28	0.32	3.73	4.92	5.04
28	2.66	0.22	-0.21	3.40	5.21	5.19
29	3.02	0.18	-0.01	3.89	5.05	5.07
30	3.01	-0.02	-0.32	4.04	4.91	5.14

Table 1: Generated datapoints used to test the DIB algorithm for geometric clustering.

On Figure 1, we can see on the first plot that the DIB algorithm can find the true clusters with ease. The second plot shows us that the algorithm is quite robust to determine the three clusters. Indeed, for $\beta \in [2;60]$ the algorithm finds them with only 50 iterations.

Figure 2 shows what is called a DIB curve. A DIB curve illustrates the trade-off between compression and prediction. It plots the mutual information $I(T;Y)$ between the cluster representation T and the target variable Y against the entropy of the cluster representation T, $H(T)$. This curve provides insights into how much information about the target variable Y can be retained in the cluster representation T, given a certain level of compression. Solutions positioned below the DIB curve are suboptimal.

However, the DIB framework does not prescribe a method for selecting a single solution from the array of solutions along its boundary. Intuitively, when confronted with a Pareto-optimal boundary[1], targeting a solution at the curve's "knee" appears to be advantageous [3]. This "knee" point represents the maximum magnitude second derivative of the curve. In extreme scenarios, where the curve exhibits a kink, the second derivative could be infinite, highlighting significant points of interest.

In the case of DIB, where hard clustering is enforced, kinks inevitably appear. This arises because by constraining $q(t \mid x)$ to be a binary matrix, where elements are either 0 or 1, both $q(t)$ and $q(y \mid t)$ can only take a limited set of values. This limitation stems from the definitions $q(t) = \sum_x q(t \mid x)p(x)$ and $q(y \mid t) = \frac{1}{q(t)} \sum_x q(t \mid x)p(x,y)$, where $p(x)$ and $p(x,y)$ are given.

As a result, both the entropy $H(T)$ and the mutual information $I(T;Y)$ are directly influenced by the distributions $q(t)$ and $q(y \mid t)$ respectively. Due to the restricted nature of these distributions, being constrained to a finite set of possible values, both entropy and mutual information can only take on a limited number of distinct values. This limitation in the range of possible values for $H(T)$ and $I(T;Y)$ leads to the formation of noticeable kinks in the DIB curve as their relationship is non-linear. These kinks signify solutions applicable across a broad range of β values. Consequently, these kinks denote solutions which are robust to variations in parameters. Such solutions are likely to capture genuine underlying structures within the dataset.

In Figure 2, the presence of a big kink is evident, highlighting the algorithm's robustness to variations in β. Specifically, for β values in at least $[0.4, 60]$, the DIB algorithm identifies three clusters within the dataset. This observation underscores how the DIB curve facilitates the determination of the optimal number of clusters in a given dataset, which is crucial for guiding subsequent analysis. It is worth noting that the curve is consistently non-linear. Each point along the curve exhibits a kink, although these kinks are minor compared to the big kink that occurs when $\beta = 10$. As mentioned earlier, this big kink is particularly interesting as it indicates solutions that are robust to variations in parameters.

When we link the observations of Figure 2 on the DIB curve with the right plot of Figure 1, we see that for $\beta \in [0.4, 60]$, the algorithm indicates three clusters which is exactly what we expect. This highlights the utility of the DIB curve to find optimal values of β for a specific dataset.

[1]A Pareto-optimal boundary represents the set of solutions where it's impossible to improve one objective without seeing a degradation in another.

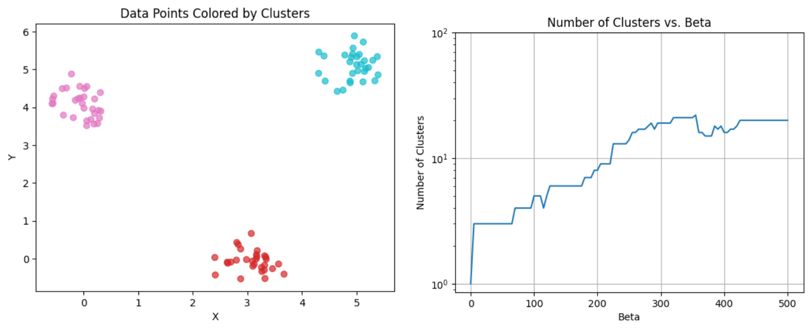


Figure 1: Left: Data points colored by clusters using the DIB algorithm with $|X| = 90$, $\beta = 10$ and 100 iterations. Right: Number of clusters determined by the DIB algorithm against β for 100 iterations. Right: DIB curve where we fix the number of iterations for each β to 100.

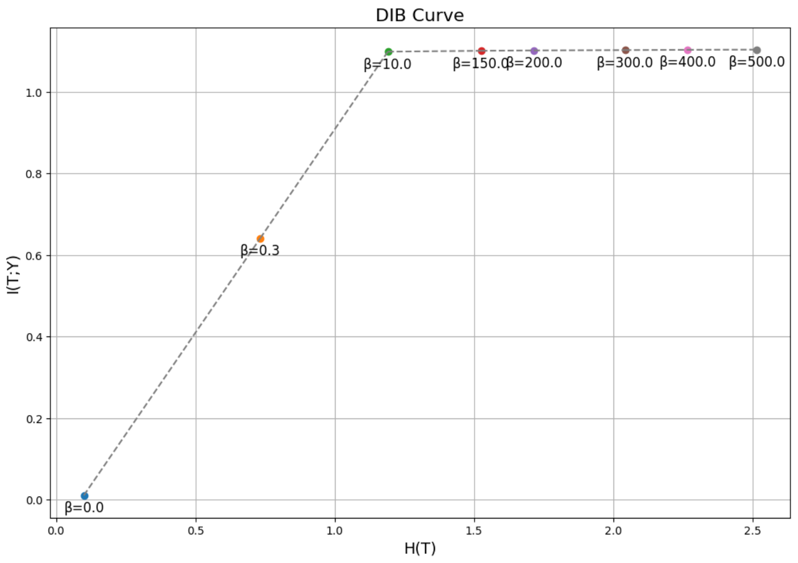


Figure 2: DIB curve where we fix the number of iterations for each β to 100. Each datapoint is on the Pareto-optimal boundary as the algorithm had enough iterations to converge for each β.

2.2 IB algorithm

Now that we have a better overview of what the deterministic information bottleneck works for geometric clustering, we implement the general information bottleneck algorithm based on Algorithm 1 of [2]. The objective is to determine whether we can replicate the IB curve in Figure 1 outlined in the paper mentioned above.

To this aim, we generate a random joint distribution $p(x, y)$ by generating a matrix of random numbers in $[0, 1]$ and normalizing the matrix to have $\sum_{x,y} p(x, y) = 1$. We then apply the IB algorithm to this joint distribution. Results are shown in Figure 3 below.

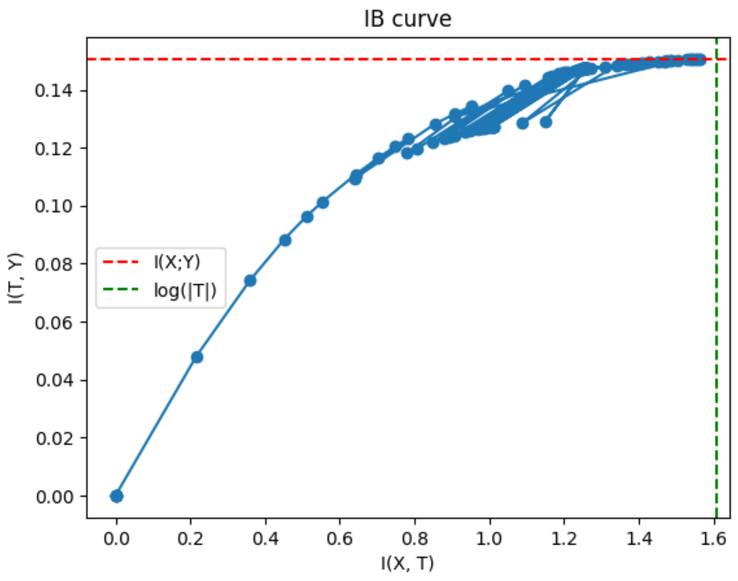


Figure 3: IB curve for a random joint distribution with $|X| = 5$, $|Y| = 3$ and $\beta \in [0; 200]$. We fix the number of iterations for each β to 1000.

As for the DIB curve, the purpose of an IB curve is to illustrate the balance between compression, represented by $I(X;T)$, and prediction, denoted by $I(T;Y)$. Indeed, at the bottom left of the curve, maximizing compression (minimizing $I(X;T)$) prioritizes reducing redundancy and extracting the most essential information from the input data X but we see that by doing this, we lose almost all information about Y and prediction is pretty bad. Conversely, at the other end, maximizing prediction (maximizing $I(T;Y)$) emphasizes capturing as much relevant information as possible to accurately predict the output Y given the latent variable T. By doing this, we see on the curve that we are forced to make little compression.

The bounds represented by a red horizontal line and a green vertical line are the theoretical bounds for $I(T;Y)$ and $I(X;T)$. $I(T;Y)$ is bounded by $I(X;Y)$ because $I(X;Y)$ is the mutual information if we do not compress input data. On the other hand, $I(X;T)$ is bounded by $\log(|T|)$ because

$$I(X;T) = H(T) - H(T \mid X) \leq H(T) \leq \log(|T|)$$

We can prove that $H(T) \leq \log(|T|)$ easily.

Proposition: Let X be a discrete random variable with distribution p. Then $H(X) \leq \log(|X|)$ with equality if and only if p is the uniform distribution [4].

Proof.

$$\begin{aligned} H(X) &= \sum_x p(x) \cdot \log\left(\frac{1}{p(x)}\right) = \mathbb{E}\left[\log\left(\frac{1}{p(X)}\right)\right] \\ &\leq \log\left(\mathbb{E}\left[\frac{1}{p(X)}\right]\right) \quad \text{by Jensen's inequality for concave functions, here } \log(x). \\ &= \log\left(\sum_x p(x) \cdot \frac{1}{p(x)}\right) = \log(|X|) \end{aligned}$$

Thus, $H(X) \leq \log(|X|)$. Furthermore, we know that Jensen's inequality[2]is an equality if and only if ϕ is affine or if X is constant. Here, $\phi(x) = \log(x)$ which is clearly not affine so $H(X) = \log(|X|) \iff p(X)$ is constant, i.e. p is the uniform distribution. □

Let's go back to the IB curve. Upon closer inspection, we notice certain points lying below the curve. These points correspond to β values where the algorithm may have struggled to converge, due to insufficient iterations. Indeed, setting the number of iterations to 1000 may not always ensure convergence of the algorithm.

Thus, two questions remain to fully understand how the algorithm works and how we can use it to the best advantage:

- What is the optimal number of iterations in order to ensure the convergence of the algorithm?
- Which value of β gives the best trade-off between compression and prediction?

2.2.1 Convergence of the algorithm

Instead of fixing the number of iterations, we could choose a metric that ensures convergence. For example, we can compare two consecutive values of the objective function $L_{\mathrm{IB}} = I(X;T) - \beta I(T;Y)$ and stop the algorithm when the difference between these two values is lower than a certain threshold. More precisely, if $L_{\mathrm{IB}}^{(n)}$ is the value of the objective function at the n-th step, we can iterate the algorithm while $|L_{\mathrm{IB}}^{(n)} - L_{\mathrm{IB}}^{(n-1)}| > \theta$ where θ is a chosen threshold. By doing this, we can achieve better results as shown in the plots below:

[2]Jensen's inequality for concave functions: $\mathbb{E}[\phi(X)] \leq \phi\left(\mathbb{E}[X]\right)$.

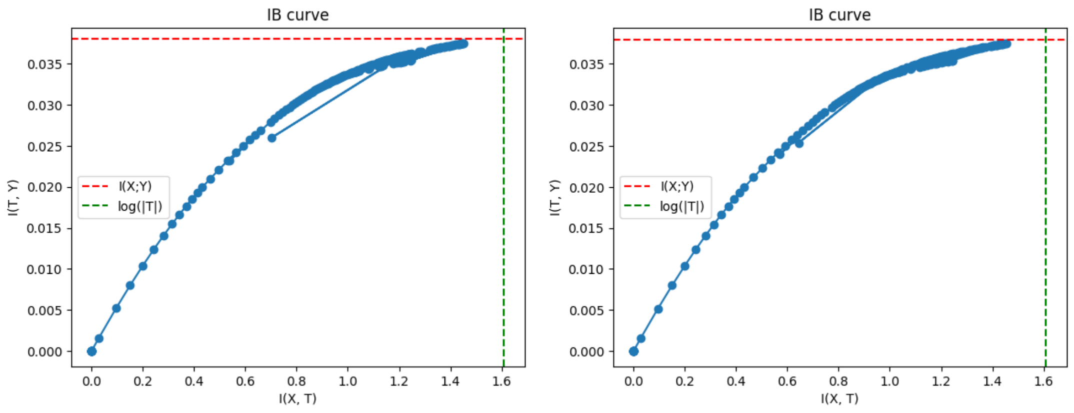


Figure 4: Comparison of the IB curve with a fixed number of iterations for each β (left) and the IB curve using the metric to ensure convergence (right). We fixed the number of iterations to 100 and the threshold θ to 1e-8.

We can clearly see that the algorithm using the metric to ensure convergence has indeed better performance. The curve is smoother and less points are below the curve. We can still see some points under the curve but this comes from the fact that we add a maximum number of iterations in the algorithm to decrease computation time. We fixed this maximum number of iterations to 10'000 which means that we could retrieve the right curve with the first algorithm if we set the number of iterations to 10'000.

2.2.2 Optimal β

The question of determining the ideal β value is not quite meaningful. Indeed, the research of an optimal β relies on the preferred quantity of clusters. The IB curve illustrates the upper bounds of prediction performance achievable when the number of clusters, represented by $I(X;T)$, is fixed. Consequently, this research depends on the specific goals of our analysis.

3 DIB and discrete choice

The DIB algorithm offers a compelling approach to distill essential insights from complex datasets while navigating the balance between relevance and compression of information. By optimizing this trade-off, DIB presents a promising avenue for enhancing model selection and evaluation within the discrete choice framework. Leveraging the principles of DIB, we propose a novel methodology that seeks to identify the most suitable discrete choice model by maximizing log-likelihood and minimizing the Akaike Information Criterion (AIC).

3.1 Formal conjecture

The main conjecture can be described informally as follows: the application of the Deterministic Information Bottleneck algorithm on a dataset can reveal hidden structural patterns, which are indicative of the appropriateness of more complex choice models like nested logit models over simpler models like logit models.

Formal Conjecture: Let D be a dataset composed of attributes and alternatives. We denote by X the attributes and by Y the alternatives. Let M_0 be a discrete choice model for D. We denote by $LL_{M_0} = \sum \log[p_{M_0}(y \mid x)]$ its log-likelihood. We can find the best model for D in terms of log-likelihood by applying iteratively the DIB algorithm on the joint distribution. More formally, let's denote by $DIB(M_i) = M_{i+1}$ the new model obtained from the results of DIB algorithm for the joint distribution $p_{M_i}(x, y) = p_{M_i}(y \mid x) \cdot p(x)$. With an optimal choice for the different parameters used, we have

$$LL_{M_{i+1}} \geq LL_{M_i} \quad \forall i \geq 0$$

Now, the goal is to prove that this conjecture is correct. First, we define the procedure that we are using to apply our conjecture. First, we create a model in order to obtain $p(y \mid x)$. We then obtain $p(x, y)$ by multiplying the conditional distribution $p(y \mid x)$ with $p(x)$ which is the empirical distirbution of X. After that, we apply the DIB algorithm on $p(x, y)$ to obtain $q(t \mid x)$, $q(t)$ and $q(y \mid t)$. From $q(t \mid x)$ and with some interpretation, we create a new model and we do the same procedure until convincing results. We summarize this procedure in the following diagram:

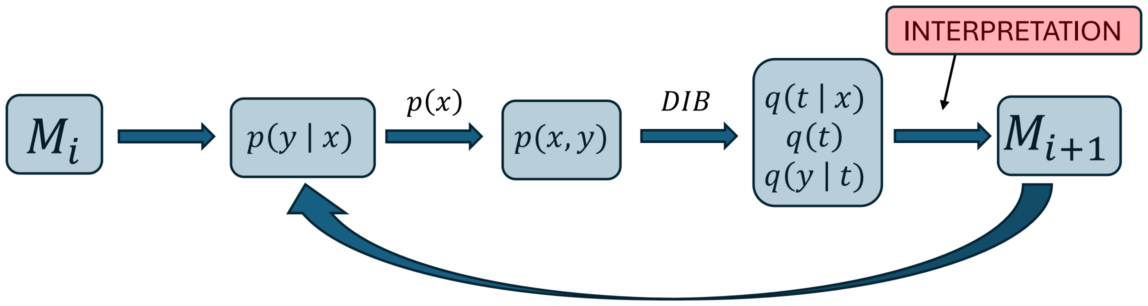


Figure 5: Diagram of the iterative procedure.

The first question that arises when seeing this diagram is: what is the interpretation step? Before answering this question, we need to understand some facts about the DIB algorithm and the log-likelihood of a model.

To interpret the results of the DIB algorithm, we are using a table of maximal probability against cluster:

max. proba	**1**	**...**	**n**
cluster			
1	c_{11}	...	c_{1n}
⋮	⋮	⋮	⋮
n	c_{n1}	...	c_{nn}

In this context, c_{ij} represents the count of individuals within cluster i whose maximal probability alternative corresponds to the j-th option. We can deduce that $\sum_{i,j} c_{ij}$ corresponds to the number of individuals in the dataset.

We create a *clustering matrix* from this table as

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix}$$

The choice of using maximum probability to fill the clustering matrix is motivated by the need for interpretability and clarity in the resulting clusters. By assigning each individual to the cluster corresponding to their highest probability alternative, we ensure that each cluster represents a distinct and most likely choice for the individuals within it. This approach simplifies the interpretation of clusters, making it easier to analyze and understand the predominant preferences within each cluster.

While filling the matrix with simulated choices could offer a more nuanced view of individual preferences, it also introduces complexity that may complexify the clear interpretation of the clusters. The maximum probability method, though potentially less precise, provides a straightforward and intuitive way to identify the dominant choice for each individual, thereby facilitating a more transparent analysis of the clustering results. This method is particularly useful in contexts where clear insights are required from the clustering process.

Claim: Let D be a dataset. If we can find a discrete choice model M_i for D such that C_{M_i} is diagonal, then the log-likelihood of D on simulated choices will be maximized almost surely.

Sketch of proof: If C_{M_i} is diagonal, it means that the DIB algorithm can exactly cluster the dataset depending on the maximal probability associated with each individual (not on the simulated choice of each individual as it is not deterministic). If this is the case, it means that our model does not miss any structure in the data, i.e. the DIB algorithm cannot find any hidden structure in the dataset. On the other hand, if C_{M_i} is not diagonal, it means that there are still some hidden structure in the dataset that the DIB algorithm can retrieve. Thus, the model is not the best that we can produce and the log-likelihood is not maximized.

Let's go back to the first question about the interpretation step. When we apply the DIB algorithm, we obtain C_{M_i}. At this point, our goal is to create a new model M_{i+1} such that the log-likelihood of this model will be greater than the log-likelihood of M_i. To do that, we observe C_{M_i} and we try to understand which structure is missing in M_i that we should add in M_{i+1}. For example, suppose that $C_{M_i} = \begin{bmatrix} 100 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$. In this case, the DIB algorithm indicates that we are missing some structure between alternative 1 and 2. Thus, we will consider a nested logit model for M_{i+1} where alternative 1 and 2 are in one nest and alternative 3 is in another nest. Indeed, our goal is to end up with a matrix as close as possible to a diagonal matrix.

Now the question is: why does the DIB algorithm can help us find models with better log-likelihood? To answer this question, we need to go back to the definitions of likelihood and information bottleneck.

We want to find the model with the biggest log-likelihood but we know that maximum likelihood estimation is the same as KL-divergence minimization [9]. Indeed, suppose that p^* is the true distribution of our dataset. Our goal is to find a distribution p which is as close as possible to p^*:

$$\begin{aligned} p_{\min} &= \arg\min_p \, D_{KL}[p^*(y \mid x) \mid\mid p(y \mid x)] = \arg\min_p \left[\sum p^*(y \mid x) \log\left(\frac{p^*(y \mid x)}{p(y \mid x)}\right)\right] \\ &= \arg\min_p \, \mathbb{E}_{y|x\sim p^*}\left[\log\left(\frac{p^*(y \mid x)}{p(y \mid x)}\right)\right] \\ &= \arg\min_p \, \mathbb{E}_{y|x\sim p^*}\left[\log(p^*(y \mid x)) - \log(p(y \mid x))\right] \end{aligned}$$

Here, $\log(p^*(y \mid x))$ does not affect the optimization as it is the "true" distribution and we can remove it from the equation.

$$\begin{aligned} p_{\min} &= \arg\min_p \, \mathbb{E}_{y|x\sim p^*}\left[-\log(p(y \mid x))\right] = \arg\max_p \, \mathbb{E}_{y|x\sim p^*}\left[\log(p(y \mid x))\right] \\ &= \arg\max_p \sum_i \log(p(y_i \mid x_i)) \end{aligned}$$

Which is exactly the maximization of the log-likelihood. Thus, we just showed that likelihood maximization is the same thing as KL-divergence minimization.

Now, let's remember how the DIB algorithm works. From a joint distribution $p(x, y)$, our goal is to find a compressed representation T of X which is a solution of the following problem:

$$\arg\min_{q(t|x)} [H(T) - \beta I(T; Y)]$$

The formal solution was given by $q(t \mid x) = \arg\max_t l(x, t)$ where

$l(x, t) = \log(q(t)) - \beta \cdot D_{KL} [p(y \mid x) \mid\mid q(y \mid t)]$.

Let's try to put everything together. Suppose that we want to find the best model in terms of log-likelihood. For that, we need to find the distribution $p(y \mid x)$ which is the closest to the true distribution $p^*(y \mid x)$, i.e. we need to solve $\arg\min_p D_{KL}[p^*(y \mid x) \mid\mid p(y \mid x)]$ but we have no information about p^*. First, we create a simple model M_1 to obtain p_1. We then apply the DIB algorithm to find q_1 and create a new model M_2 closer to the p^*. If we have no restriction on the class of models that we can choose and we iterate this procedure infinitely, we would end up by finding p^* but as our choice for a specific model is limited, we will never find the exact probability distribution of the dataset but rather, we will come closer and closer to this true distribution. This iterative procedure is summarized in the diagram below:

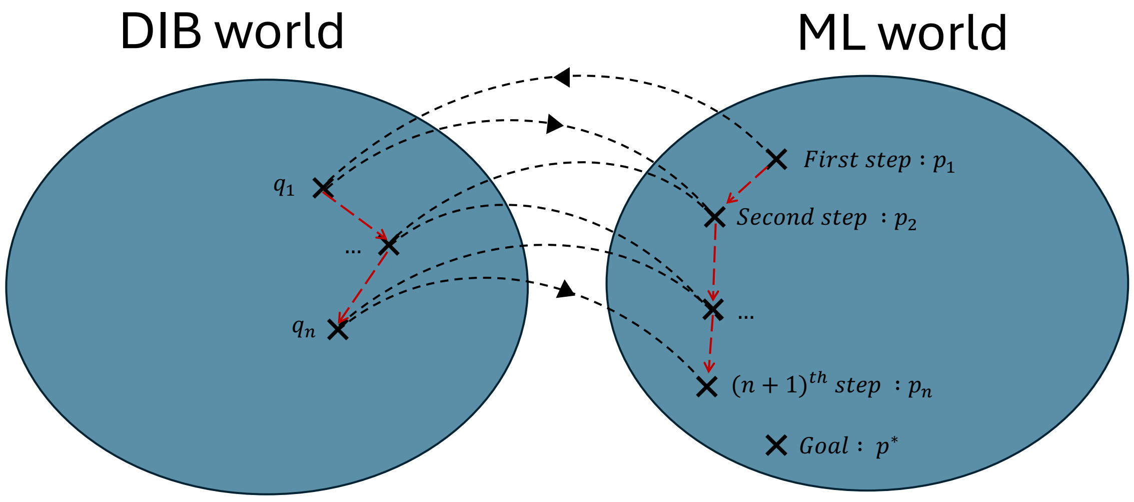


Figure 6: Diagram of the iterative procedure.

This iterative procedure where we use the DIB algorithm in order to find the best model in terms of log-likelihood can be described with the following equation:

$$\arg\max_{p(y|x),\ q(t|x)} \{\log(q(t)) - \beta \cdot D_{KL}[p(y \mid x) \mid\mid q(y \mid t)]\}$$

The first term $\log(q(t))$ promotes minimizing the number of unique values for t. However, since our goal is to find a diagonal clustering matrix, we might favor $|T| = |Y|$ as a special case, making this first term not relevant. It is important to note that the support of the distribution of the random variable T can change depending on the class of models that we are considering. Therefore, while this term may not always be directly relevant to our specific objective, it is important to keep in my mind that this term can have an impact when changing the class of choosable models.

This leaves us with the following equation:

$$\underset{p(y|x),\ q(t|x)}{\arg\min}\ D_{KL}[p(y \mid x) \mid\mid q(y \mid t)]$$

Recalling that T is a compressed representation of X, we now understand that our procedure is a reformulation of the maximum likelihood estimation for $p(y \mid x)$. Thus our conjecture seems to be valid.

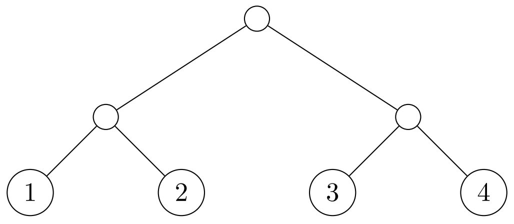
3.2 Interpretation step

The interpretation step is only made with the help of the clustering matrix defined above. The idea is that, if the DIB algorithm put two alternatives in one cluster based on the maximal probability of each individual, it indicates that we are missing some structures between these two alternatives in our base model.

For example, suppose that we find the following clustering matrix after applying DIB algorithm on a multinomial logit model:

$$C = \begin{bmatrix} 100 & 100 & 0 & 0 \\ 0 & 0 & 100 & 100 \end{bmatrix}$$

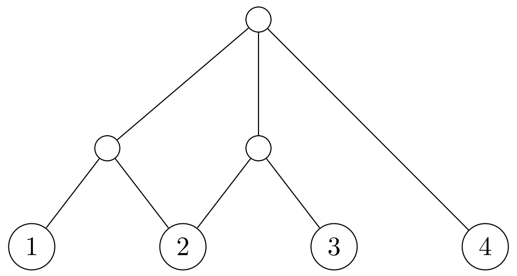
In this case, alternatives 1 and 2 are in one cluster and alternatives 3 and 4 in another. Thus, it indicates that we are missing some structure between alternatives 1 and 2, and also between alternatives 3 and 4. From this clustering matrix, we could infer that a nested logit model with alternatives 1 and 2 in the first nest and alternatives 3 and 4 in the second nest (see graph below) is a better representation than a multinomial logit model.



Suppose that we apply again the algorithm on our new nested logit model and the clustering matrix looks like this:

$$C = \begin{bmatrix} 100 & 50 & 0 & 0 \\ 0 & 50 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$$

Now, the algorithm seems to indicate a hidden structure between alternatives 1, 2 and 3. Thus, we could infer that a cross-nested logit model with three nests as below would be the best for this dataset.



Now, if we apply one last time the algorithm, we should end up with a diagonal clustering matrix but as we only consider three types of models (multinomial logit, nested logit and cross-nested logit), we generally find a final clustering matrix close to diagonal like this (this matrix is an arbitrary example):

$$C = \begin{bmatrix} 97 & 6 & 2 & 0 \\ 2 & 90 & 5 & 1 \\ 0 & 4 & 88 & 0 \\ 1 & 0 & 5 & 99 \end{bmatrix}$$

We did not find a formal rule to choose the M_{i+1} model from the clustering matrix but a general rule of thumb could be given as follows:

Rule of thumb: A model M_n is missing some structure between two alternatives j and k if

- $c_{ij} \geq \frac{1}{|L|+1} \sum_{l \in L_j} c_{lj}$ where L_j is the set of clusters where alternative j is not alone
- $c_{ik} \geq \frac{1}{|L|+1} \sum_{l \in L_k} c_{lk}$ where L_k is the set of clusters where alternative k is not alone

Let's take another example to well understand this rule. Let the clustering matrix C be as follows (this matrix is an arbitrary example):

$$C = \begin{bmatrix} 0 & 180 & 40 \\ 200 & 20 & 30 \\ 0 & 0 & 130 \end{bmatrix}$$

Intuitively, we could say that some structure is missing between alternatives 2 and 3 as well as between alternative 1 and 3. Indeed, the first cluster is composed of 180 individuals with maximal probability 2 and 40 individuals with maximal probability 3 which seems to reveal an hidden structure between these two alternatives. The second cluster is composed of 200 individuals with maximal probability 1, 20 individuals with maximal probability 2 and 30 individuals with maximal probability 3. This seems to reveal some structure between the three alternatives, but the structure between alternatives 1 and 2 seems weaker than the one between alternatives 1 and 3 as there are only 20 individuals with maximal probability 2 in this cluster. The third cluster is only composed of individuals with maximal probability 3, which does not indicate any hidden structure. Now, let's see if our rule of thumb also indicate what our intuition told us.

$L_1 = \{2\}$, $L_2 = \{1, 2\}$ and $L_3 = \{1, 2\}$. Indeed, $3 \notin L_3$ because alternative 3 is alone in this cluster.

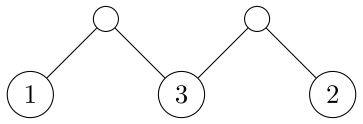
Now, if we apply the rule of thumb, we see from cluster 1 that we are missing some structure between alternatives 2 and 3 as

- $c_{12} = 180 \geq \frac{1}{3}(180 + 20) \approx 66.67$
- $c_{13} = 40 \geq \frac{1}{3}(40 + 30) \approx 23.33$

and we are also missing some structures between alternatives 1 and 3 as

- $c_{21} = 200 \geq \frac{200}{2} = 100$
- $c_{23} = 30 \geq \frac{1}{3}(40 + 30) = 23.33$

From that, we infer that M_{n+1} will be a cross-nested logit model where alternative 1 is in one nest, alternative 2 in another and alternative 3 is between these two nests as shown below:



3.3 Connection with AIC

We have demonstrated that our iterative procedure can identify the optimal model in terms of log-likelihood. However, one might argue that achieving this is relatively straightforward by always selecting the most general model, such as a cross-nested logit model, which would almost surely have the highest log-likelihood. We now demonstrate that the iterative procedure does not only give the best model in terms of log-likelihood, but also in terms of the Akaike Information Criterion (AIC) [9].

$$AIC = 2k - 2\log(L) = 2(k - \log(L))$$

where k is the number of parameters in the model and L is the likelihood of the model.

Conjecture: If the DIB algorithm indicates a new model M_{i+1}, then $AIC_{M_{i+1}} \leq AIC_{M_i}$.

The primary goal of Akaike Infomation Criterion is model selection. The smaller the AIC, the better is the model. Thus, in our case when we are searching for the best model, we could use the following function:

$$\underset{p(y|x)}{\arg\min}\ AIC = \underset{p(y|x)}{\arg\min}\{2k - 2\log\left(L_{p(y|x)}\right)\}$$

Recall that the search for the best model with DIB take the following form:

$$\underset{p(y|x),\ q(t|x)}{\arg\max}\ \{\log(q(t)) - \beta \cdot D_{KL}[p(y \mid x) \mid\mid q(y \mid t)]\}$$

These two functions seem to share some similarities. Indeed, we already mentioned the direct connection between log-likelihood maximization and KL divergence minimization. Now the question is, can we link the number of parameters k in the model with $\log(q(t))$?

In AIC, k encourages that we use as few parameters in the model as possible. In DIB, $\log(q(t))$ encourages that we use as few values of t as possible [2], i.e. it encourages that we have as few clusters as possible. But we learned from the interpretation step that if the number of clusters is smaller than the number of alternatives, the clustering matrix is not diagonal and we can find a better model with more parameters. Thus, $\log(q(t))$ helps to find the optimal k which will maximize the AIC. Thus, for AIC, we use k to penalize the use of too much parameters whereas for DIB, we push in the other direction to complexify the model until reaching optimal number of parameters. That is why we are using an $\arg\min$ for AIC and an $\arg\max$ for DIB.

One thing is still missing. Let's explore the relationship between the factor 2 in AIC and β in DIB.

Firstly, the factor 2 in AIC has historical roots [9], and it's worth noting that various sources may employ different scaling factors for this criterion.

Secondly, the introduction of the parameter β in the DIB algorithm serves a specific purpose. It ensures that both $\log(q(t))$ and $D_{KL}[p(y \mid x) \mid\mid q(y \mid t)]$ operate on a compatible scale. Since the goal of the Kullback-Leibler divergence is to approach 0, β is introduced to artificially enhance its influence within the function. This adjustment is necessary to ensure that both terms contribute meaningfully to the overall objective of the DIB algorithm.

Thus, it seems that there is a direct correspondence between AIC and DIB which justify our conjecture at the beginning of this chapter.

Note that our conjecture is highly dependent on the interpretation step. Therefore, it is crucial to exercise caution when applying the conjecture. We must always be mindful that our interpretation could be incorrect, particularly if the results do not align with our expectations. In such cases, it is important to revisit and critically assess our initial interpretation to ensure that any potential errors are identified and addressed. This vigilance will help to maintain the validity and reliability of our findings. For example, if there is no clear structure in the clustering matrix, one should first try to modify the value of β and conduct additional tests to ensure that nothing has been overlooked.

Furthermore, our iterative procedure is most suitable for relatively small datasets with a manageable number of alternatives. When we attempted to apply the procedure to a very large dataset consisting of 10,000 individuals choosing among 100 alternatives, the computation time was excessively long, making it impractical to complete within the project's remaining timeframe.

4 Test the conjecture on real data

Before testing the conjecture on real data, we need to have an idea of what are typical values for the trade-off parameter β. By making multiple tests and with the help of different references, we consider that $\beta \in [0; 100]$ is a good range of values for this parameter (see Appendix for more details).

4.1 Airline dataset

We first try the conjecture on the Airline dataset which is available on Biogeme (https://biogeme.epfl.ch/#data). This dataset contains responses from an Internet choice survey conducted by Boeing Commercial Airplanes in 2004-2005, where 3609 respondents ranked three flight options based on various attributes such as fare, travel time, and transfers, along with demographic and situational variables. We include a table of summary statistics for each variable that we will use:

	Average	**St. Dev.**	**Min**	**Max**
TripTimeHours_1	3.74	1.59	0.67	6.35
TripTimeHours_2	5.50	1.58	1.83	8.85
TripTimeHours_3	5.48	1.67	1.92	8.85
Fare_1	405.66	199.87	80.00	1330.00
Fare_2	407.07	200.96	80.00	1390.00
Fare_3	405.20	197.68	80.00	1275.00
BestAlternative_1	0.69	0.46	0	1
BestAlternative_2	0.16	0.37	0	1
BestAlternative_3	0.14	0.35	0	1

Table 2: Summary statistics for each variable that are used in the models for Airline dataset.

Let's begin with a simple multinomial logit model:

$$U_1 = \beta_{\text{time}} \cdot TripTimeHours_1 + \beta_{\text{cost}} \cdot Fare_1 + \epsilon_1$$
$$U_2 = ASC_2 + \beta_{\text{time}} \cdot TripTimeHours_2 + \beta_{\text{cost}} \cdot Fare_2 + \epsilon_2$$
$$U_3 = ASC_3 + \beta_{\text{time}} \cdot TripTimeHours_3 + \beta_{\text{cost}} \cdot Fare_2 + \epsilon_3$$

The table below shows the results of the estimated model.

Parameter	**Estimate**	**Robust Asymptotic SE**	**t-statistic**	**p-value**
ASC_2	-1.219947	0.029937	-40.751126	0.0
ASC_3	-1.443420	0.069661	-20.720730	0.0
BETA_Time	-0.317315	0.035511	-8.935792	0.0
BETA_Cost	-0.018839	0.000679	-27.745889	0.0
log-likelihood	-2425.22			
AIC	4858.44			

Table 3: Results of the estimation of the multinomial logit model for Airline dataset.

Now, we can apply our iterative procedure in order to find a new model with a higher log-likelihood and a lower AIC. We thus apply the DIB algorithm to the joint distribution $p(x, y) = p(y \mid x) \cdot p(x)$ where

$p(y \mid x)$ comes directly from the estimated logit model above and $p(x)$ is the empirical distribution of the attributes that we use in the model.

In the following, we present only the most relevant results. However, it is important to note that selecting an appropriate β value in the DIB algorithm is not straightforward. Multiple tests and careful consideration are required to determine a suitable β.

max. proba	**1**	**2**	**3**
cluster			
0	303	11	212
1	440	217	0
2	2426	0	0

simulated choice	**1**	**2**	**3**
cluster			
0	233	58	235
1	308	275	74
2	1956	250	220

Table 4: Results of the DIB algorithm with $\beta = 100$ when the base model is the multinomial logit model for Airline dataset.

By looking at the maximum probability table, the algorithm indicates that we should consider a cross-nested logit model where alternative 1 is in the two nests. Let's estimate this cross-nested logit model to see if the algorithm indicates a better model in terms of log-likelihood and AIC.

Parameter	**Estimate**	**Robust Asymptotic SE**	**t-statistic**	**p-value**
ASC__2	-1.119183	0.123657	-9.050731	0.0
ASC__3	-1.336384	0.120480	-11.092213	0.0
BETA__Time	-0.254901	0.051710	-4.929441	8.618421e-07
BETA__Cost	-0.015433	0.001766	-8.736950	0.0
lambda__21	0.853477	0.094546	9.027124	0.0
lambda__13	0.511141	0.083846	6.096208	1.200705e-09
alpha__1__with__2	1.722967	0.468653	3.676424	2.399755e-04
log-likelihood	-2422.16			
AIC	4858.31			

Table 5: Results of the estimation of the cross-nested logit model for Airline dataset where alternative 1 is in the two nests.

This indication is a good one as the AIC for the logit model is 4858.44 and the AIC for the cross-nested logit model is 4858.31. Now, let's see if the algorithm suggests another model if we apply it again but this time by taking $p(y \mid x)$ from the cross-nested logit model that we just estimated.

max. proba	**1**	**2**	**3**
cluster			
0	0	184	42
1	3164	32	7
2	6	0	174

simulated choice	**1**	**2**	**3**
cluster			
0	60	121	45
1	2379	459	365
2	53	14	113

Table 6: Results of the DIB algorithm with $\beta = 10$ when the base model is the cross-nested logit model with alternative 1 in the two nests for Airline dataset.

Here, the algorithm seems to indicate a nested logit model with alternatives 2 and 3 in the same nest. Again, we estimate this model to see if this indication is a good one.

Parameter	Estimate	Robust Asymptotic SE	t-statistic	p-value
ASC_2	-1.011323	0.042998	-23.520484	0.0
ASC_3	-1.162476	0.000314	-3700.142070	0.0
BETA_Time	-0.257938	0.034340	-7.511219	7.349676e-14
BETA_Cost	-0.015801	0.000744	-21.239385	0.0
lambda_23	0.530093	0.041568	12.752331	1.467150e-05
log-likelihood	-2397.53			
AIC	4805.06			

Table 7: Results of the estimation of the nested logit model for Airline dataset where alternatives 2 and 3 are in the same nest.

The log-likelihood and the AIC are indeed better for this model and we can see on the table below that this model is in fact the best one that we can produce for this dataset.

Model	log-likelihood	AIC
LM	-2425	4858
NLM, $n_1 = \{1, 2\}$	-2425	4860
NLM, $n_1 = \{1, 3\}$	-2424	4858
NLM, $n_1 = \{2, 3\}$	**-2397**	**4805**
CNLM, $n_1 \cap n_2 = \{1\}$	-2422	4858
CNLM, $n_1 \cap n_2 = \{2\}$	-2397	4809
CNLM, $n_1 \cap n_2 = \{3\}$	-2397	4809

Table 8: Log-likelihood and AIC for all possible models for Airline dataset.

Therefore, the DIB algorithm helps us to find the best model in terms of LL and AIC.

4.2 SwissMetro dataset

We now try the conjecture on the SwissMetro dataset which is also available on Biogeme. This dataset contains survey data collected in March 1998 from 441 rail travelers between St. Gallen and Geneva, Switzerland, who provided responses to assess the impact of the Swissmetro system compared to car and train options. We include a table of summary statistics for each variable that we will use:

	Average	St. Dev.	Min	Max
TRAIN_TT	166.63	77.35	31	1049
SM_TT	87.47	53.55	8	796
CAR_TT	123.80	88.71	0	1560
TRAIN_CO	514.34	1088.93	4	5040
SM_CO	670.34	1441.59	6	6720
CAR_CO	78.74	55.26	0	520
GA	0.14	0.35	0	1
Choice	2.15	0.63	1	3

We take the following multinomial logit model as first model:

$$\begin{aligned}
U_{\text{train}} &= \beta_{\text{time}} \cdot TRAIN_{TT} + \beta_{\text{cost}} \cdot TRAIN_{CO} \cdot I(GA = 0) + \epsilon_{\text{train}} \\
U_{\text{SM}} &= ASC_{\text{SM}} + \beta_{\text{time}} \cdot SM_{TT} + \beta_{\text{cost}} \cdot SM_{CO} \cdot I(GA = 0) + \epsilon_{\text{SM}} \\
U_{\text{car}} &= ASC_{\text{car}} + \beta_{\text{time}} \cdot CAR_{TT} + \beta_{\text{cost}} \cdot CAR_{CO} + \epsilon_{\text{car}}
\end{aligned}$$

The table below shows the results of the estimated model.

Parameter	Estimate	Robust Asymptotic SE	t-statistic	p-value
ASC__SM	0.655274	0.155548	4.212688	0.000025
ASC__CAR	0.667142	0.005898	113.116901	0.000000
BETA__time	-0.012750	0.000815	-15.640170	0.000000
BETA__cost	-0.007934	0.005884	-1.348313	0.177586
log-likelihood	-8663.06			
AIC	17363.24			

Table 9: Results of the estimation of the multinomial logit model for SwissMetro dataset.

We can now apply our iterative procedure.

max. proba	**1**	**2**	**3**
cluster			
0	0	6842	1619
1	0	0	2249

simulated choice	**1**	**2**	**3**
cluster			
0	1049	4916	2496
1	367	1313	569

Table 10: Results of the DIB algorithm with $\beta = 10$ when the base model is the multinomial logit model for SwissMetro dataset.

By looking at the maximum probability table, the algorithm indicates that we should consider a nested logit model where alternative 1 (train) is in one nest and alternatives 2 and 3 (SM and car) in another one. We thus estimate this new model to see if this indication is a good one.

Parameter	Estimate	Robust Asymptotic SE	t-statistic	p-value
ASC__SM	-0.152823	0.196117	-0.779244	0.0435853
ASC__CAR	-0.423693	0.123328	-3.435497	0.000594
BETA__time	-0.020277	0.002313	-8.765210	0.000000
BETA__cost	-0.013852	0.000846	-16.380949	0.000000
lambda__CAR__SM	0.457684	0.196259	11.132768	0.000000
log-likelihood	-8576.62			
AIC	17163.23			

Table 11: Results of the estimation of the nested logit model with alternatives 2 and 3 in the same nest for SwissMetro dataset.

This indication is a good one as the AIC for the logit model is 17334 whereas the AIC for the nested logit model is 17163. Now, let's see if the algorithm suggests another model if we apply it again but this time by taking $p(y \mid x)$ from the nested logit model that we just estimated.

DIB results for $\beta = 8$: 2 clusters

max. proba	**1**	**2**	**3**
cluster			
0	128	1570	0
1	0	7303	1709

simulated choice	**1**	**2**	**3**
cluster			
0	527	1171	9
1	902	4996	3114

Table 12: Results of the DIB algorithm with $\beta = 8$ when the base model is the nested logit model with alternatives 2 and 3 in the same nest for SwissMetro dataset.

Here, the algorithm indicates a cross-nested logit model where alternative 2 (SM) is in the two nests. Once again, this indication corresponds to the best model in terms of log-likelihood and AIC.

Parameter	**Estimate**	**Robust Asymptotic SE**	**t-statistic**	**p-value**
ASC_SM	0.206893	0.116850	1.770587	0.076658
ASC_CAR	-2.148966	0.219706	-9.781088	0.0
BETA_time	-0.029486	0.001467	-20.092575	0.0
BETA_cost	-0.021405	0.000984	-21.752910	0.0
lambda_CAR_SM	8.781123	0.025130	349.425149	0.0
lambda_SM_TRAIN	1.814824	0.126595	14.335714	0.0
alpha_SM_WITH_TRAIN	-7.769435	0.506294	-15.345705	0.0
log-likelihood	-8454.64			
AIC	16923.28			

Table 13: Results of the estimation of the cross-nested logit model with alternative 2 in the two nests for SwissMetro dataset.

Model	**log-likelihood**	**AIC**
LM	-8663	17334
NLM, $n_1 = \{1, 2\}$	-8662	17334
NLM, $n_1 = \{1, 3\}$	-8519	17049
NLM, $n_1 = \{2, 3\}$	-8576	17163
CNLM, $n_1 \cap n_2 = \{1\}$	-8486	16986
CNLM, $n_1 \cap n_2 = \{2\}$	**-8454**	**16923**
CNLM, $n_1 \cap n_2 = \{3\}$	-8468	16951

Table 14: Log-likelihood and AIC for all possible models for SwissMetro dataset. Alternative 1, 2 and 3 correspond to train, SwissMetro and car respectively.

Unfortunately, this cross-nested logit model is not a viable model as $\lambda_{\text{car,SM}}, \lambda_{\text{SM,train}}, \alpha_{\text{SM\_with\_train}} \notin [0, 1]$.

4.3 Optima dataset

We now try the conjecture on the Optima dataset which is also available on Biogeme. This dataset contains revealed preference data from a 2009-2010 survey conducted by Car Postal and EPFL researchers, involving 1124 respondents from rural areas in French and German-speaking regions of Switzerland, who recorded trip details and socio-economic information to analyze travel behavior and mode choice. We include a table of summary statistics for each variable that we will use:

	Average	St. Dev.	Min	Max
TimePT	107.88	86.52	0	745
TimeCar	40.68	47.61	0	494
ReportedDuration	60	72.92	0	855
MarginalCostPT	11.11	16.13	0	230
CostCarCHF	5.76	8.34	0	67.65
WalkingTimePT	39.63	28	0	213
WaitingTimePT	13.13	22.07	0	392
Choice	0.78	0.54	0	2

We begin with the following multinomial logit model:

$$\begin{aligned} U_{\text{PT}} &= \beta_{\text{time}} \cdot (TimePT + WalkingTimePT + WaitingTimePT) + \beta_{\text{cost}} \cdot MarginalCostPT + \epsilon_{\text{PT}} \\ U_{\text{private}} &= ASC_{\text{private}} + \beta_{\text{time}} \cdot TimeCar + \beta_{\text{cost}} \cdot CostCarCHF + \epsilon_{\text{private}} \\ U_{\text{soft}} &= ASC_{\text{car}} + \beta_{\text{time}} \cdot ReportedDuration + \epsilon_{\text{soft}} \end{aligned}$$

The table below shows the results of the estimated model.

Parameter	Estimate	Robust Asymptotic SE	t-statistic	p-value
ASC__private	0.389411	0.086557	4.498920	7.242624e-06
ASC__soft	-2.241692	0.127427	-17.591950	0.000000e+00
BETA__Time	-0.002885	0.000662	-4.358362	1.380328e-05
BETA__Cost	-0.038338	0.004736	-8.095621	8.881784e-16
log-likelihood	-1465.09			
AIC	2938.18			

Table 15: Results of the estimation of the multinomial logit model for Optima dataset.

We then apply the iterative procedure.

DIB results for $\beta = 30$: 3 clusters

max. proba	**1**	**2**	**3**
cluster			
0	41	1743	0
1	1	6	2
2	0	113	0

simulated choice	**1**	**2**	**3**
cluster			
0	527	1156	101
1	1	4	4
2	10	95	8

Table 16: Results of the DIB algorithm with $\beta = 30$ when the base model is the multinomial logit model for Optima dataset.

Here, it seems that the DIB algorithm indicates a nested model where alternatives 1 and 2 are together, i.e. public transport and private transport. We thus estimate this new model to see if this indication is a good one.

Parameter	Estimate	Robust Asymptotic SE	t-statistic	p-value
ASC__private	-4.663075	1.396242	-3.339732	8.548050e-04
ASC__soft	-9.292362	1.394756	-6.662358	3.517853e-11
BETA__Time	0.005494	0.003071	1.789281	7.372870e-02
BETA__Cost	0.736106	0.109439	6.726174	2.296896e-11
lambda__PT__private	-8.702495	1.386370	-6.277179	4.259970e-10
log-likelihood	-1324.05			
AIC	2658.10			

Table 17: Results of the estimation of the nested logit model where alternatives 1 and 2 are in the same nest for Optima dataset.

This model is indeed better in terms of log-likelihood and AIC than the multinomial logit model. We thus continue the procedure to see if we can find another model with a higher log-likelihood and a lower AIC. Let's apply the DIB algorithm on $p(x, y)$ with $p(y \mid x)$ coming from the nested logit model above.

DIB results for $\beta = 30$: 3 clusters

max. proba	**1**	**2**	**3**
cluster			
0	14	791	6
1	98	0	0
2	0	997	0

simulated choice	**1**	**2**	**3**
cluster			
0	241	463	107
1	67	30	1
2	210	779	8

Table 18: Results of the DIB algorithm with $\beta = 30$ when the base model is the nested logit with alternative 1 and 2 in the same nest for Optima dataset.

There is no clear trend for another model. Let's see if indeed, any cross-nested model has a lower LL or AIC than the nested model above:

Model	log-likelihood	AIC
LM	-1465	2938
NLM, $n_1 = \{1, 2\}$	**-1324**	**2658**
NLM, $n_1 = \{1, 3\}$	-1513	3037
NLM, $n_1 = \{2, 3\}$	-1426	2862
CNLM, $n_1 \cap n_2 = \{1\}$	-1324	2662
CNLM, $n_1 \cap n_2 = \{2\}$	-1324	2662
CNLM, $n_1 \cap n_2 = \{3\}$	-1382	2780

Table 19: Log-likelihood and AIC for all possible models for Optima dataset. Alternative 1, 2 and 3 correspond to public transport, private transport and soft transport respectively.

No model has a higher log-likelihood and a lower AIC than the nested logit model with alternatives 1 and 2 in the same nest. Thus, once again, the DIB algorithm made a good guess !

Unfortunately, this model is not a viable model. Indeed, $\lambda_{PT,private} \notin [0; 1]$ and β_{cost} and β_{time} are greater than 0 which makes no sense.

5 Conclusions

This thesis aimed to explore a new approach in decision-making by connecting information theory and discrete choice analysis. Through careful investigation, we sought to improve our understanding and prediction of complex human behavior.

We began with an overview of discrete choice analysis by defining the main models that were useful for us. With a solid understanding of these models, we then explored information theory to build a foundation for our research.

At the heart of this thesis is the integration of the Deterministic Information Bottleneck (DIB) algorithm with discrete choice theory. This integration offers a new approach for selecting models. Our practical tests showed that models selected with our approach performed better on key metrics such as log-likelihood and AIC.

Our findings suggest that the DIB algorithm can improve decision-making processes by extracting valuable insights from complex data. By combining information theory with decision-making models, we gained both theoretical and practical insights for analyzing and predicting human behavior.

In summary, this thesis shows how integrating different fields can address complex challenges in decision-making analysis. Our exploration of the DIB algorithm and its application to discrete choice analysis contributes to the broader understanding of decision-making processes.

6 Appendix

6.1 Typical values for β

To find what are typical values for β, we analyze the results of the DIB algorithm for different values of β on the same dataset. Typical values of β correspond to values where we can observe a clear change in results.

6.1.1 Logit model on Optima dataset

$\beta = 0 \rightarrow q(t) = [0.99, 3.99e{-}04, 3.98e{-}04] \quad q(y \mid t) = [[0.28, 0.65, 0.059], [0.28, 0.65, 0.059], [0.28, 0.65, 0.059]]$

$\beta = 1 \rightarrow q(t) = [0.99, 3.99e{-}04, 3.98e{-}04] \quad q(y \mid t) = [[0.28, 0.65, 0.059], [0.28, 0.65, 0.059], [0.28, 0.65, 0.059]]$

$\beta = 5 \rightarrow q(t) = [0.99, 5.32e{-}05, 8.51e{-}04] \quad q(y \mid t) = [[0.28, 0.65, 0.059], [0.31, 0.31, 0.38], [0.04, 0.82, 0.14]]$

$\beta = 10 \rightarrow q(t) = [0.99, 5.08e{-}04, 1.01e{-}03] \quad q(y \mid t) = [[0.28, 0.65, 0.059], [0.78, 0.08, 0.13], [0.03, 0.46, 0.51]]$

$\beta = 50 \rightarrow q(t) = [0.90, 0.02, 0.07] \quad q(y \mid t) = [[0.29, 0.65, 0.05], [0.55, 0.33, 0.11], [0.08, 0.80, 0.12]]$

$\beta = 100 \rightarrow q(t) = [0.85, 0.02, 0.12] \quad q(y \mid t) = [[0.29, 0.65, 0.05], [0.54, 0.35, 0.10], [0.11, 0.79, 0.10]]$

$\beta = 500 \rightarrow q(t) = [0.79, 0.03, 0.18] \quad q(y \mid t) = [[0.31, 0.64, 0.05], [0.52, 0.38, 0.10], [0.13, 0.77, 0.10]]$

6.1.2 Nested logit model on Optima dataset

$\beta = 0 \rightarrow q(t) = [0.99, 3.99e{-}04, 3.98e{-}04] \quad q(y \mid t) = [[0.28, 0.66, 0.055], [0.28, 0.65, 0.054], [0.27, 0.66, 0.055]]$

$\beta = 1 \rightarrow q(t) = [0.99, 3.99e{-}04, 3.98e{-}04] \quad q(y \mid t) = [[0.28, 0.66, 0.055], [0.28, 0.65, 0.054], [0.27, 0.66, 0.055]]$

$\beta = 5 \rightarrow q(t) = [0.99, 1.15e{-}05, 8.74e{-}04] \quad q(y \mid t) = [[0.28, 0.66, 0.055], [0.97, 0.02, 0.002], [0.07, 0.85, 0.07]]$

$\beta = 10 \rightarrow q(t) = [0.98, 0.015, 0.001] \quad q(y \mid t) = [[0.27, 0.67, 0.05], [0.83, 0.15, 0.01], [0.009, 0.91, 0.07]]$

$\beta = 50 \rightarrow q(t) = [0.79, 0.02, 0.17] \quad q(y \mid t) = [[0.30, 0.63, 0.05], [0.74, 0.23, 0.02], [0.08, 0.85, 0.07]]$

$\beta = 100 \rightarrow q(t) = [0.76, 0.03, 0.21] \quad q(y \mid t) = [[0.31, 0.63, 0.05], [0.73, 0.25, 0.02], [0.09, 0.83, 0.06]]$

$\beta = 500 \rightarrow q(t) = [0.73, 0.03, 0.23] \quad q(y \mid t) = [[0.31, 0.63, 0.05], [0.71, 0.27, 0.02], [0.11, 0.83, 0.06]]$

From these different values, we can justify our choice of using β values between 0 and 100. Indeed, we can see here that we make significant changes in the results when β goes from 10 to 50. As this result is only for a specific dataset, we do not restrict too much the range of β values that we will be using and we consider that $\beta \in [0; 100]$ is a good range.

7 References

[1] Tishby, N., Pereira, F. C., & Bialek, W. (2000). The information bottleneck method. arXiv preprint physics/0004057.

[2] Strouse, D. J., & Schwab, D. J. (2017). The deterministic information bottleneck. Neural computation, 29(6), 1611-1630.

[3] Strouse, D. J., & Schwab, D. J. (2019). The information bottleneck and geometric clustering. Neural computation, 31(3), 596-612.

[4] MacKay, D. J. (2003). Information theory, inference and learning algorithms. Cambridge university press.

[5] Ben-Akiva, M. E., & Lerman, S. R. (1985). Discrete choice analysis: theory and application to travel demand (Vol. 9). MIT press.

[6] Simeone, O. (2018). A brief introduction to machine learning for engineers. Foundations and Trends® in Signal Processing, 12(3-4), 200-431

[7] Slonim, N., & Weiss, Y. (2002). Maximum likelihood and the information bottleneck. Advances in neural information processing systems, 15.

[8] Friedman, N., Mosenzon, O., Slonim, N., & Tishby, N. (2013). Multivariate information bottleneck. arXiv preprint arXiv:1301.2270.

[9] Anderson, D., & Burnham, K. (2004). Model selection and multi-model inference. Second. NY: Springer-Verlag, 63(2020), 10.