APPLICATION OF THE INFORMATION-THEORETIC CLUSTERING TO THE DISCRETE CHOICE

Alexandre Monti

Supervised by Pavel Ilinov, Evangelos Paschalidis and Professor 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 modeling 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

Before diving into the main part of this work where we will apply IB framework to discrete choice, we first recall some basics about utility theory.

Utility theory serves as a fundamental framework in various fields, offering insights into how individuals make decisions amid competing alternatives. Rooted in the principles of rational decision-making, utility theory asserts that individuals aim to optimize their decision outcomes when presented with choices. To quantify individuals' preferences and rationalize decision-making, utility theory introduces the utility functions, a mathematical construct that assigns numerical values to potential outcomes. These utility functions should reflect individuals' subjective preferences and attitudes toward various options, encapsulating their desires and needs.

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 inherent 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 the stochastic element in the utility function. It embodies the unobservable factors that influence decision-making but cannot be captured by the deterministic component. This component introduces variability into the utility function, accounting for the uncertainty inherent in human decision-making processes.

1.1.1 The logit model

The logit model 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. Moreover, its versatility allows for application across diverse fields, including transportation and economics. In essence, the logit model serves as a practical and accessible tool for dissecting decision processes.

Definition: Let C be a choice set. Assume that the random components of the alternatives ϵ_i are i.i.d. $\mathrm{EV}(\eta, \mu)$. The logit model is derived as

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

In this context, $V_i = \sum_k \beta_k z_{ik}$, denoting a linear combination of the vector of attributes z_i for alternative i. 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. This is what we will do in the following of this work.

1.1.2 Nested logit model

While the logit model serves as a cornerstone in discrete choice analysis, offering simplicity and interpretability, it does come with its limitations. One notable constraint is its assumption of independence of irrelevant alternatives (IIA), which can sometimes lead to unrealistic choice behaviors. Additionally, the logit model may struggle to capture complex decision-making processes where choices exhibit hierarchical or nested structures. In response to these limitations, the nested logit model emerges as a powerful extension, offering a solution to address these shortcomings. By allowing for correlated choices within specified nests or groups, the nested logit model provides a more nuanced understanding of decision-making dynamics, making it particularly well-suited for scenarios where choices are influenced by both individual attributes and group characteristics.

Definition: Let C be the choice set and $C_1, ..., C_M$ a partition of C. 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

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

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 on the nested logit model:

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 is 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, buse, 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 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. 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 lack 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 modeling and provide valuable guidance for researchers and practitioners grappling with the complexities of choice behavior analysis.

1.2 Basics of information theory

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

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: Suppose that we have a movie recommendation system that suggests movies to users based on their past viewing history and ratings. In this context, let's consider the uncertainty in predicting a user's movie rating. We denote by R the ratings and $p(r)$ the probability distribution of the ratings.

$$H(R) = \sum_r p(r) \cdot \log(1/p(r))$$

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: Let's go back to the first example about movie recommendation system. We denote by G the genre of a movie. Then, we could interpret $H(R,G)$ as the joint uncertainty of a user's movie rating and the genre of 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: 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

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

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 eliminating any irrelevant or duplicated elements. More precisely, the goal of this method is to find a concise representation of the input data X while preserving the relevant information that X gives about the output data Y.

To this aim, we need to solve the following optimization problem: let T be the compressed input data. 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.

The idea behind this optimization problem is that the first term pushes for compression, while the second term emphasizes the importance of keeping the 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 $T = G$.

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 issue 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 $T \longleftrightarrow X \longleftrightarrow Y$.

How can we interpret the difference between IB and DIB methods? The disparity arises from a modification in the first term where $I(X;T)$ shifts to $H(T)$. In the domain of channel coding, $I(X;T)$ represents compression and suggests a limitation on communication cost between X and T. However, by adopting a source coding perspective, we can substitute $I(X;T)$ with $H(T)$, which denotes the representational cost of T. We can also see the difference between the two methods by substracting one to another:

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

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.

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:

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

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 multiple different x values to the same t. 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 KL divergence term, 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 methods

2.1 Geometric DIB algorithm

Firstly, we'll focus on implementing the Deterministic Information Bottleneck method specifically tailored for geometric clustering. This approach will help us to understand the underlying dynamics and behavior of the algorithm. In this chapter, we demonstrate how to apply the DIB method to a standard clustering problem, for which classical algorithms such as K-means are typically used.

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 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.

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 tradeoff 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 inherently 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. 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.

For example, suppose that we have $|X| = 3$, $|T| = 2$, $|Y| = 2$ and $q(t \mid x) = [[1,0],[1,0],[0,1]]$ which means that x_1 and x_2 are in the first cluster and x_3 is in the second cluster. We set $p(x_i) = p_i$ and $p(x_i, y_j) = z_{ij}$. In this case, $q(t) = [p_1 + p_2, p_3]$ and

$$q(y \mid t) = \begin{bmatrix} \frac{z_{11}+z_{21}}{p_1+p_2} & \frac{z_{31}}{p_3} \\ \frac{z_{12}+z_{22}}{p_1+p_2} & \frac{z_{32}}{p_3} \end{bmatrix}$$

But p_i and z_{ij} are given. Thus we understand why $q(t)$ and $q(y \mid t)$ can only take a limited set of values when we use DIB.

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

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. 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 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.

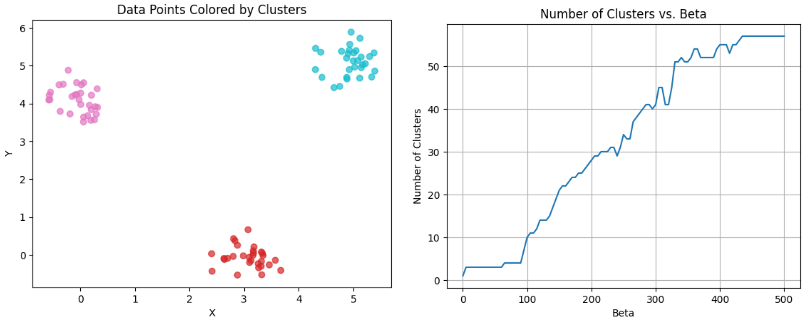


Figure 1: Left: Data points colored by clusters using the DIB algorithm with $|X| = 90$, $\beta = 10$ and 100 iterations. Right: Number of clusters determined by the DIB algorithm against β for 100 iterations. Right: DIB curve where we fix the number of iterations for each β to 100.

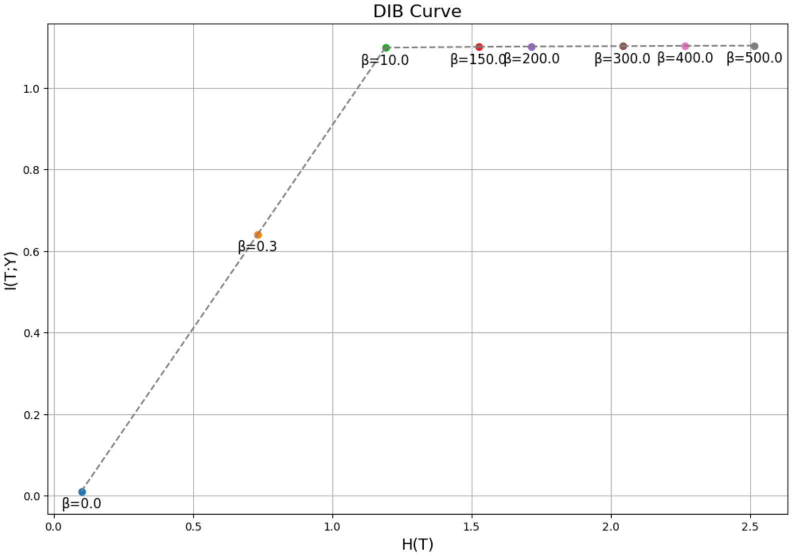


Figure 2: DIB curve where we fix the number of iterations for each β to 100.

2.2 General 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. After a few trials, the first results appear:

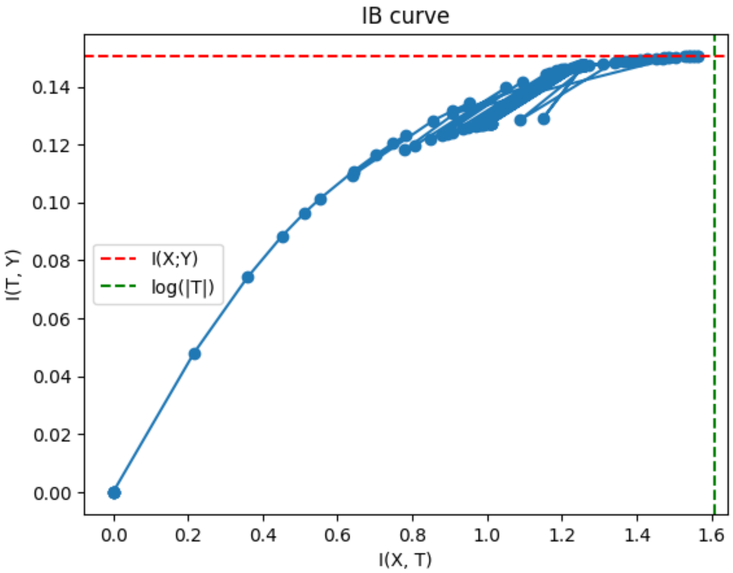


Figure 3: IB curve for a random joint distribution with $|X| = 5$, $|Y| = 3$ and $\beta \in [0; 200]$. We fix the number of iterations for each β to 1000.

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.

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:

- What is the optimal number of iterations in order to ensure the convergence of the algorithm?
- Which value of β gives the best tradeoff between compression and prediction? In other words, which β is at the crossroads of the red and the green curves?

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 $L_{\text{IB}} = I(X;T) - \beta I(T;Y)$ and stop the algorithm when the difference between these two values is lower than a certain threshold. More precisely, if $L_{\text{IB}}^{(n)}$ is the value of the objective function at the n-th step, we can iterate the algorithm while $|L_{\text{IB}}^{(n)} - L_{\text{IB}}^{(n-1)}| > \theta$ 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: $\mathbb{E}[\phi(X)] \leq \phi\left(\mathbb{E}[X]\right)$.

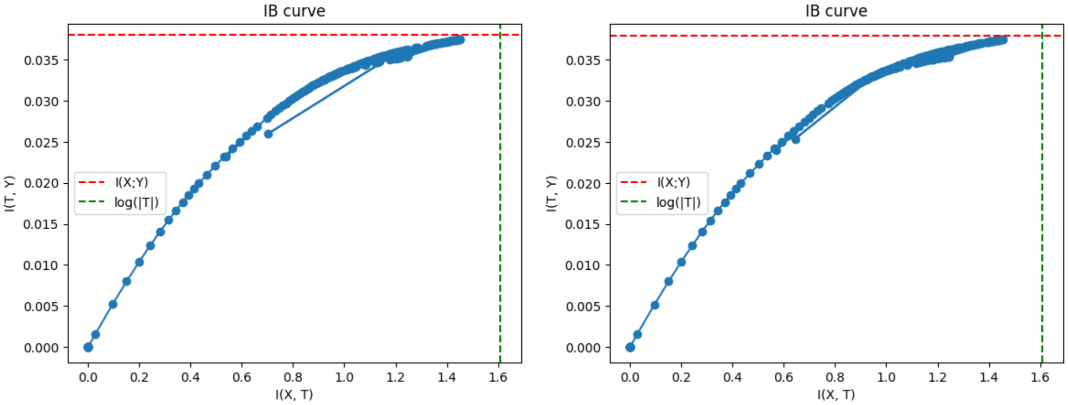


Figure 4: Comparison of the IB curve with a fixed number of iterations for each β (left) and the IB curve using the metric to ensure convergence (right). We fixed the number of iterations to 100 and the threshold θ to 1e-8.

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, where we intersect the red and green curves on the IB curve, is not quite meaningful. It typically leads to a large β value that diminishes compression, essentially negating the utility of the variable T.

Instead, 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.

2.3 General DIB algorithm

Our next objective is to develop the general DIB algorithm. This step is essential as we seek to evaluate and contrast the performance of both IB and DIB algorithms in discrete choice scenarios. To achieve this, we need to have both general algorithms readily available for comparison and analysis.

It appears that the DIB algorithm we developed for geometric clustering serves also as a solution to the general DIB problem. Indeed, our implementation is based on the general DIB algorithm proposed in [2], which accounts for its inherent generality. With this insight, we can seamlessly transition to analyzing the distinctions between IB and DIB algorithms.

3 DIB and discrete choice

The DIB algorithm offers a compelling approach to distill essential insights from complex datasets while navigating the delicate 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). This integration of DIB with discrete choice theory not only enhances predictive performance but also enhances the interpretability of the chosen models.

3.1 Formal conjecture

Main Conjecture: 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 makes sense. First, we recall the procedure that we are using to apply our conjecture. First, we create a model in order to obtain $p(y \mid x)$ and $p(x, y)$ as we are only using the empirical distribution for $p(x)$. 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:

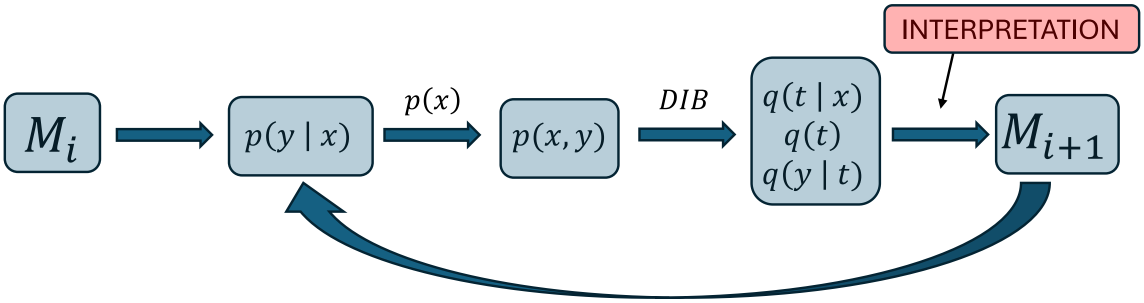


Figure 5: Diagram of the procedure.

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}	$\dots$	c_{1n}
⋮	$\vdots$	$\vdots$	$\vdots$
n	c_{n1}	$\dots$	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

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

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

Sketch of proof: If C_M 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 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. 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. To do that, we observe C_M and we try to understand which structure is missing in M that we should add in $M+1$. For example, suppose that $C_M = \begin{bmatrix} 100 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$. 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 $M+1$ 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. 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.

$$p_{\min} = \arg\min_p \ \mathbb{E}_{y|x \sim p^*} \left[-\log(p(y \mid x)) \right] = \arg\max_p \ \mathbb{E}_{y|x \sim p^*} \left[\log(p(y \mid x)) \right]$$

At this point, we need to use the law of large numbers. Indeed, as $y_i \mid x \sim p^*$ for all i, we know by the LLN that

$$\begin{aligned} p_{\min} &= \arg\max_p \ \mathbb{E}_{y|x \sim p^*} \left[\log(p(y \mid x)) \right] = \arg\max_p \lim_{n \to \infty} \frac{1}{n} \sum_i \log(p(y_i \mid x_i)) \\ &= \arg\max_p \sum_i \log(p(y_i \mid x_i)) \end{aligned}$$

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:

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

The formal solution was given by $q(t \mid x) = \underset{t}{\arg\max}\; l(x,t)$ where

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

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 $\underset{p}{\arg\min}\; D_{KL}[p^*(y \mid x) \mid\mid p(y \mid x)]$ 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 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:

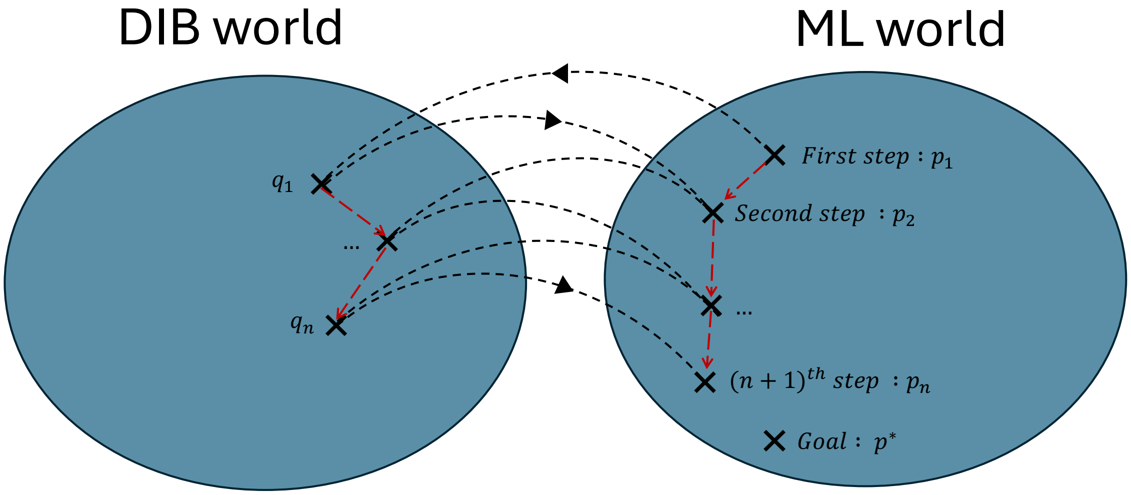


Figure 6: Diagram of the iterative procedure.

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:

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

The first term $\log(q(t))$ promotes minimizing the number of unique values for t, but as our goal is to find a diagonal clustering matrix, we will always favor $|T| = |Y|$. Thus, this first term is not relevant for our purpose.

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.

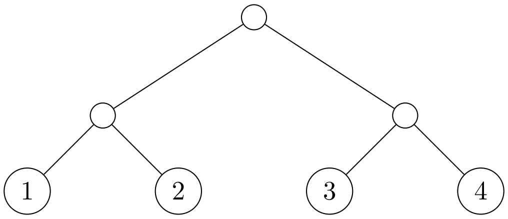
3.2 Interpretation step: how the model changes given the output of the clustering

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:

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

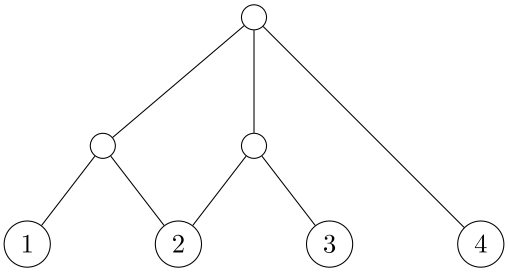
In this case, alternatives 1 and 2 are in one cluster and alternatives 3 and 4 in another. Thus, it indicates that we are missing some structure between alternatives 1 and 2, and also between alternatives 3 and 4. From this clustering matrix, we could infer that a nested logit model with alternatives 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:

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

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

Rule of thumb: A model M 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 in one example above:

$$C = \begin{bmatrix} 0 & 167 & 42 \\ 3170 & 49 & 37 \\ 0 & 0 & 144 \end{bmatrix}$$

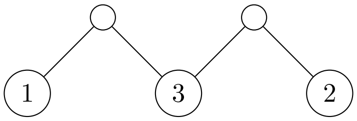
$L_1 = \{2\}$, $L_2 = \{1,2\}$ and $L_3 = \{1,2\}$. Indeed, $3 \notin L_3$ because alternative 3 is alone in this cluster. Now, if we apply the rule of thumb, we see from cluster 1 that we are missing some structure between alternatives 2 and 3 as

- $c_{12} = 167 \geq \frac{1}{3}(167+49) = 72$
- $c_{13} = 42 \geq \frac{1}{3}(42+37) = 26.33$

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

- $c_{21} = 3170 \geq \frac{3170}{2} = 1585$
- $c_{23} = 37 < \frac{1}{3}(42+37) = 26.33$

From that, we infer that $M+1$ 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

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

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 $M+1$, then $AIC_{M+1} \leq AIC_M$.

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 in the model k 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.

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 tradeoff 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.

Let's begin with a simple logit model:

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

DIB results for $\beta = 100$: 3 clusters

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

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. This indication is a good one as the LL for the logit model is -2425 and the LL for the cross-nested logit model is -2422. Now, let's see if the algorithm suggests another model if we apply it again.

DIB results for $\beta = 9$: 3 clusters

max. proba	**1**	**2**	**3**
cluster			
0	0	167	42
1	3170	49	37
2	0	0	144

simulated choice	**1**	**2**	**3**
cluster			
0	55	111	43
1	2399	473	384
2	38	10	96

Here, the algorithm seems to indicate a cross-nested logit model with alternative 3 in the two nests and once again, this indication is a good one.

	LM	NLM 12	NLM 13	NLM 23	CNLM with 1	CNLM with 2	CNLM with 3
AIC	4858	4860	4858	4805	4858	NaN	4781
LL	-2425	-2425	-2424	-2397	-2422	NaN	-2383

Once again, 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 take the following logit model as first model:

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

DIB results for $\beta = 10:$ 2 clusters

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

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. This indication is a good one as the LL for the logit model is -8663 and the LL for the nested logit model is -8519. Furthermore, the AIC for the logit model is 17334 whereas the AIC for the nested logit model is 17049. Now, let's see if the algorithm suggests another model if we apply it again.

DIB results for $\beta = 10:$ 2 clusters

max. proba	**1**	**2**	**3**
cluster			
0	31	1887	0
1	0	6922	1870

simulated choice	**1**	**2**	**3**
cluster			
0	542	1367	9
1	897	4829	3066

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 LL and AIC:

	LM	**NLM 12**	**NLM 13**	**NLM 23**	**CNLM with 1**	**CNLM with 2**	**CNLM with 3**
AIC	17334	17334	17049	17163	16986	16923	NaN
LL	-8663	-8662	-8519	-8576	-8486	-8454	NaN

4.3 Optima dataset : model 1

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 take the following logit model as first model:

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

DIB results for $\beta = 50$: 3 clusters

max. proba	**1**	**2**	**3**
cluster			
0	6	1708	0
1	38	1	3
2	0	149	1

simulated choice	**1**	**2**	**3**
cluster			
0	503	1114	97
1	20	17	5
2	18	122	10

The interpretation of the results is a bit complicated as many maximal probabilities are for the private alternative. But if we focus on the maximal probability table, we could suppose that the algorithm suggest a nested logit model where one nest is public transport with soft transport and the other nest is private transport.

Indeed, this model is the best in terms of LL and AIC compared to the other nested logit models:

	PT/soft vs. private	**PT/private vs. soft**	**soft/private vs. PT**
AIC	2799	2838	2867
LL	-1394	-1414	-1428

Let's see now if the algorithm suggest a cross-nested model from the nested logit model.

DIB results for $\beta = 10$: 2 clusters

max. proba	**1**	**2**	**3**
cluster			
0	80	1769	0
1	57	0	0

simulated choice	**1**	**2**	**3**
cluster			
0	495	1232	122
1	37	6	14

DIB results for $\beta = 50$: 3 clusters

max. proba	**1**	**2**	**3**
cluster			
0	40	1398	0
1	97	0	0
2	0	371	0

simulated choice	**1**	**2**	**3**
cluster			
0	442	891	105
1	57	15	25
2	33	332	6

There is no clear trend for a cross-nested logit model. And indeed, the nested logit model has the best LL and AIC overall.

	CNLM with PT	**CNLM with private**	**CNLM with soft**
AIC	NaN	2852	2871
LL	NaN	-1419	-1429

In this case, the theory seems to work but the base model was not really interesting. Let's try with another base model to see what is happening.

4.4 Optima dataset : model 2

We begin with the following new logit model:

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

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

Here, is seems that the DIB algorithm indicates a nested model where alternatives 1 and 2 are together, i.e. public transport and private transport. Let's see if this model is the best nested model that we can produce:

	PT/soft vs. private	**PT/private vs. soft**	**soft/private vs. PT**
AIC	3037	2658	2862
LL	-1513	-1324	-1426

Indeed, the nested model with PT and private transport together is the best that we can produce for the nested models. Now, let's see if the DIB algorithm suggests a cross-nested model.

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

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

	CNLM with PT	**CNLM with private**	**CNLM with soft**
AIC	NaN	2732	NaN
LL	NaN	-1359	NaN

Thus, once again, the DIB algorithm made a good guess !

4.5 Restaurant dataset

Continue on file week-11-LM-restaurant.ipynb

5 Conclusions

This thesis has aimed to explore new paths in decision-making by connecting the worlds of information theory and discrete choice analysis. Through careful investigation and analysis, we have sought to uncover insights and methods that improve our understanding and prediction of complex human behavior.

Our work began with a dive into the basics of dscrete choice models, recognizing their vital role in understanding decision-making across various areas. With a solid understanding of these concepts, we delved into information theory, laying a strong foundation for our subsequent exploration.

At the core of our thesis lies the integration of the Deterministic Information Bottleneck (DIB) algorithm with discrete choice theory, presenting a fresh approach for selecting and evaluating models. By harnessing the power of DIB, we developed a method aimed at maximizing predictive accuracy while making selected models easier to understand. Through practical testing, we demonstrated the superior performance of DIB-selected models, showing their effectiveness in outperforming alternative approaches based on key metrics.

Our findings highlight the potential of DIB to transform decision-making processes, providing a valuable tool for distilling essential insights from complex data. By combining information theory with decision-making models, we not only expanded our theoretical understanding but also enriched our practical toolkit for analyzing and predicting human behavior.

In summary, this thesis demonstrates the transformative power of interdisciplinary research, showing how the integration of different fields can address complex challenges. Through our exploration of the Deterministic Information Bottleneck algorithm and its application to discrete choice analysis we developed the collective knowledge of decision-making analysis.

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

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

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

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

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

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

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

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

6.1.2 Nested logit model on Optima dataset

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

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

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

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

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

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

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

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 FOR YOU AND ME

7.1 TO DO LIST & MEETING

- DONE: Write about the fact that everything depends on the interpretation step -> end of chapter 4.3
- DONE: Rewrite and polish the report
- Try the restaurant dataset

8 References in APA format

[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.