Learning Multinomial Logit Model

January 20, 2025

Contents

1. Learning-MNL

El presente archivo busca ser una explicación general de la arquitectura de Learning-MNL, para ello comienza dando ideas generales en la modelación del problema y como los autores del paper lo abordan, el valor de este archivo es que busca ser clarificador respecto a como se construye la arquitectura y agrega observaciones y comentarios respecto a su funcionamiento, los cuales fueron recogidos a lo largo del estudio del modelo. Cabe mencionar que este archivo no busca ser una revisión exhaustiva del paper, si no que una revisión enfocada en el modelo propuesto, por lo que diversos puntos comentados en el paper no han sido tratados. Entre ellos, se puede mencionar discusiones respecto a las redes neuronales, su papel como aproximadores universales y sus limitaciones en modelos de elección discreta, entre otros.

1.1. Ideas Generales

En el contexto de modelización de elecciones discretas existen múltiples formas de afrontar como los consumidores realizan sus elecciones, una forma muy común de modelar como toman sus elecciones es asumiendo que lo hacen siendo racionales y con ello, eligen la alternativa que más utilidad les otorga, pese a que se ha observado que esto no siempre es así, resulta ser una buena aproximación al problema. A partir de esto surge una gran familia de modelos de elección discreta, entre ellos el modelo Logit Multinomial (MNL), Nested Logit (NL), Mixed Logit y otros.

El modelo más simple es el MNL, que determina su función de utilidad como una función lineal en los parámetros con respecto a los atributos de cada alternativa, lo que puede resultar en una sobre simplificación y no representativa de los datos reales, sin embargo, no es sencillo cambiar este supuesto por alguno más complejo porque de hacerse, se debe escoger una función de utilidad que sea adecuada a los datos, es decir, debemos conocer la distribución de los datos *a priori*, de modo que esta tarea es en general difícil. Además, la elección de una función de utilidad lineal es también provechosa porque permite un alto nivel de interpretabilidad, característica que es de crítica importancia para los modeladores.

1.2. Modelación de Learning-MNL

La idea de fondo en Learning-MNL (L-MNL) es dividir en sub-parte la utilidad determinística de la siguiente manera:

$$U_n = f(\chi_n; \beta) + r(\mathcal{Q}_n; \omega) + \varepsilon_n \tag{1}$$

Donde:

- U_n representa la utilidad del individuo n
- $f(\chi_n, \beta)$ representa el término impulsado por el conocimiento, que se asume interpretable. Acá χ_n es un conjunto de características interpretables, por ejemplo, costo, tiempo, distancia, etc. β corresponde a un vector con los parámetros a ser estimados para el buen ajuste del modelo.
- $r(\mathcal{Q}_n, \omega)$ es el término impulsado por los datos, es una representación aprendida desde el conjunto de características $\mathcal{Q}_n$, que pueden ser variables que se piensa que afectan a las decisiones de los individuos pero que no se sabe interpretar como, por ejemplo, el clima, la

hora, etc. Este conjunto no necesariamente debe ser disjunto de χ_n pero se muestra en el paper que conviene que lo sea. Por su parte, ω corresponde a los parámetros aprendidos por la red neuronal.

- ε_n corresponde al error, en la modelación se asume que se distribuye como una variable aleatoria Gumbel° con locación 0 y escala 1, esto se asume así para que tenga sentido aplicar un modelo Logit Multinomial.

Con ello, la probabilidad de que el n-ésimo individuo escoga la ì-ésima alternativa de un conjunto C_n de alternativas posibles, está dada por:

$$P_n(i) = \frac{e^{f_i(\chi_n;\beta)+r_i(\mathcal{Q}_n;\omega)}}{\sum_{j\in C_n} e^{f_j(\chi_n;\beta)+r_j(\mathcal{Q}_n;\omega)}} \tag{2}$$

1.3. Arquitectura de Learning-MNL

La red neuronal que aproxima a la función $r(\mathcal{Q}_n;\omega)$ es flexible en cuánto a su estructura, los autores consideran la red neuronal más básica que corresponde al perceptrón multicapa (MLP, por sus siglas en inglés) denso, es decir, una DNN. Con ello, la arquitectura del modelo se puede escribir como una red neuronal cuyo output sea el de la ecuación (2), como se ve en la siguiente figura:

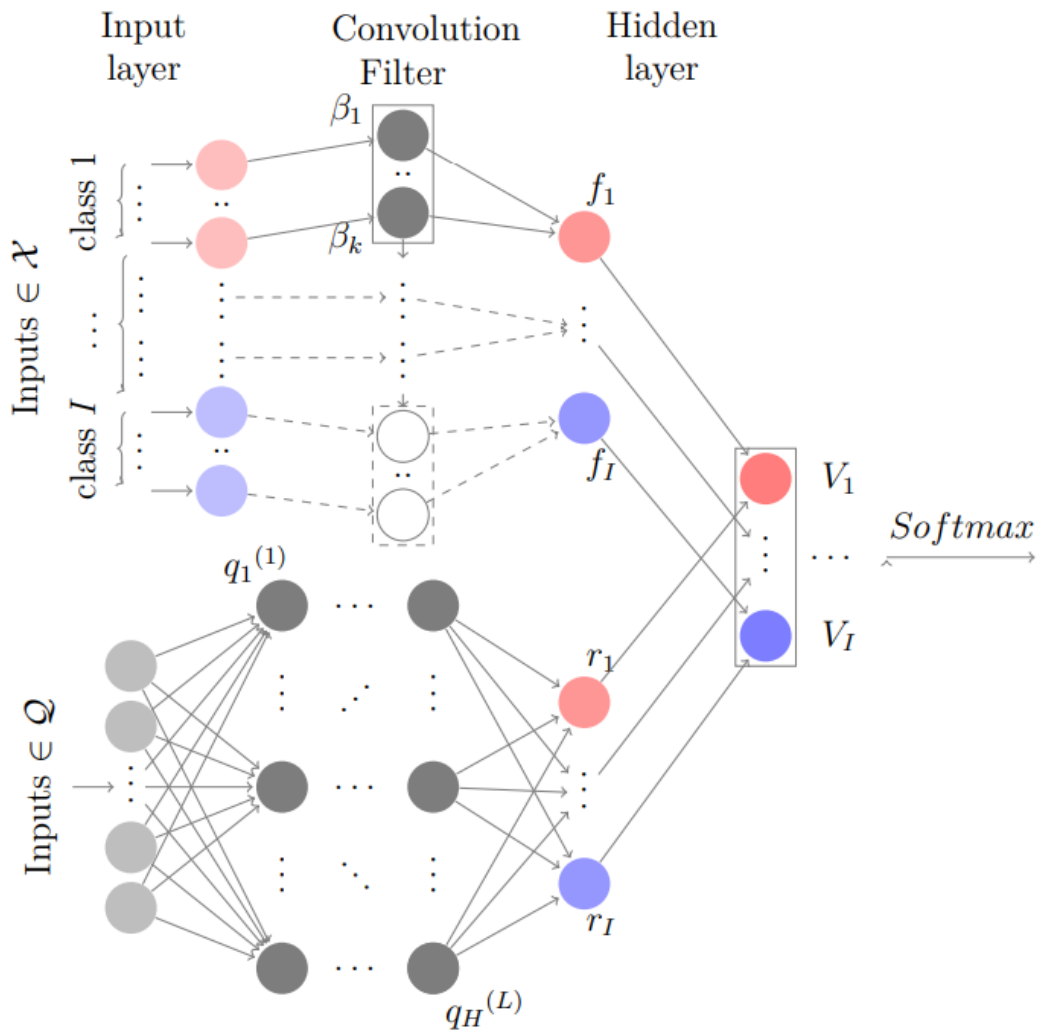

Figure 1: Arquitectura del modelo L-MNL. En la parte superior, se tiene la implementación de un modelo MNL con el uso de una CNN. En la parte inferior, tenemos una DNN que permite obtener el término de aprendizaje de representación r_i. Los resultados de cada parte se suman definiendo la nueva función sistemática de la ecuación (2).

Es de suma importancia comprender la imagen anterior, dado que esta lleva en sí la idea de trasfondo del paper. Primero, notemos que la presencia de una red neuronal convolucional (CNN) cuyo input está ordenado verticalmente para cada clase. Esta CNN no es más que una manera astuta de implementar el MNL, ya que las especificaciones de su arquitectura (filtro y stride del tamaño del número de betas, una sola capa oculta y la disposición de su input) permiten que utilizando como función de activación Softmax, su output sea justamente el MNL, por lo que no se debe pensar como algo más sofisticado que eso. Por otra parte, se tiene en gris una DNN, que recibe como input el conjunto de características Q, las cuales no presentan ningún orden en particular ya que es una red densa, obteniendo de acá el término de representación r_i que luego será adicionado al término f_i saliente de la CNN, para que al aplicar Softmax se recupere el modelo planteado por los autores. Con ello, se verifica que la arquitectura presentada corresponde a una manera de escribir el modelo de DCM como una RNN.

Vale la pena observar que hace sentido pensar que esta astuta manera de escribir la parte lineal de la utilidad tiene relación con la implementación en código, ya que los parámetros de la red neuronal se deben ajustar en simultáneo para llegar (intentar llegar) al óptimo, y también notar que la interpretabilidad de esta arquitectura viene dada únicamente de la parte relacionada al MNL. En la modelación, los autores establecen que el input relativo a X tiene las características Interpretables, mientras que el input relativo a Q no, con esto y la condición de que la elasticidad se impone (o asume) nula en la salida de la DNN, se concluye que toda la interpretabilidad del modelo reside en el MNL, lo que puede ser un punto de mejora ya que hoy en día existen herramientas que permiten interpretar las decisiones tomadas por RNN. En este punto, cabe mencionar que la imposición de que $\frac{\partial r_i}{\partial t_i} = 0$, con t_i la alternativa i, no tiene consecuencias prácticas en el sentido de la preparación del dataset o limitantes de casos en que el modelo se pueda aplicar, sus implicaciones no son aclaradas en el paper, y según lo reflexionado en torno a esto, puede tener consecuencias en la capacidad predictiva del modelo.

Como último comentario, notar que la decisión de considerar una DNN para el término de representación, implica que para los parámetros no ajustados, todas las características influyen en las demás. Por ejemplo, en un estudio de elección entre Auto y Bus para transportarse, las características presentes en Bus afectarían la elección de Auto, lo que debe ser aprendido por la red neuronal, pero que a priori dirá que sí; esto se estudia en el paper al enfrentar el modelo a data sintética.

Bibliography

[1] Brian Sifringer, Virginie Lurkin, Alexandre Alahi. (2020). Enhancing Discrete Choice Models with Representation Learning

Index of Figures