Modelaje econométrico con monitoreo de desempeño vía estrategias de ML

Alejandro Hidalgo
Maestría en ciencias en computación
ITAM
Ciudad de México, México
ahidalg4@itam.mx

***Abstract*— Este proyecto implementa, adapta y compara distintas especificaciones para modelos econométricos de estimación de sistemas de demanda del tipo AIDS (Almost Ideal Demand Systems, por sus siglas en inglés), en mercados cuasi-diferenciados (e.g. RTEC "Ready to eat cereal" por sus siglas en inglés) mediante el enfoque metodológico de aprendizaje de máquina. El objetivo del proyecto consiste en estimar las relaciones directas entre las variables de entrada, es decir, los precios de cada marca, y las variables de salida, entendidas como las diferentes demandas de cada marca. El proyecto adapta algunos de estos modelos para poder incluir información del desempeño de cada una de las marcas, utilizando una base de datos que contiene observaciones semanales y distintas métricas de ejecución de los productos a nivel marca (e.g. inventarios promedio por tienda). Adicionalmente, flexibiliza los modelos al controlar por estacionalidad, lo que se realiza empleando bases de datos recolectadas en la industria de consumo masivo nacional (NIELSEN). La relevancia de estos modelos consiste en la posibilidad de calcular estimaciones de impacto sobre distintos indicadores clave de producto (Key Performance Indicator, por sus siglas en inglés) del negocio para poder contestar la siguiente pregunta: Dado un nivel agregado de incrementos de precios requeridos por una empresa multimarca en un mercado cuasi-diferenciado, ¿Cuál es el valor de ciertos indicadores clave de producto ante distintas estrategias de incremento de precios sobre sus marcas que le lleve al nivel -ponderado- de incremento de precio requerido? Finalmente, con la mejor especificación, se construye un simulador de escenarios de incrementos de precios que le permite a la empresa multimarca decidir sobre sus distintas estrategias de incrementos de precio en términos de distintos indicadores clave de producto (KPIs).**

***Index Terms*— Modelos AIDS (almost ideal demand systems), diferenciación de productos, industria de cereales listos para comer (RTEC), poder de mercado, dinámicas de la competencia a precios, aprendizaje de maquina, descenso en gradiente.**

I. INTRODUCCIÓN

La industria de cereales procesados listos para comer (Ready to Eat Cereals -RTEC-, por sus siglas en inglés) se caracteriza por una importante concentración de mercado, altos márgenes entre el precio y el costo, una gran inversión en mercadotecnia, así como una introducción agresiva de nuevas variedades [1]. Adicionalmente, el alto nivel de dificultad en la creación de los productos, así como los elevados costos de entrada asociados, generan la necesidad de producir múltiples marcas (variedades de cereales) en la misma línea de producción, con la finalidad de capitalizar en las economías de escala en aspectos tales como producción, embalaje y distribución. Esto genera un tipo de mercado caracterizado por contar con pocas empresas, que poseen múltiples marcas o variedades de cereales, las cuales compiten entre sí, no sólo en sus precios, sino también en su variedad y difusión. Desde el punto de vista de alguna de estas empresas lo que se busca año con año, en un proceso de estabilidad financiera, es generar incrementos de precios sobre sus productos que permitan mantener estos márgenes ante la evolución de los distintos costos asociados a la venta (e.g. insumos) al mismo tiempo que se minimice la pérdida total en ventas asociada a estas estrategias. Cuales son estas estrategias y como compararlas es,entonces,un problema de vital importancia para la empresa. Este problema es el que denominaremos problema de *pricing*

Para categorías de productos (e.g. cereales) que son propias del canal de venta moderno (e.g. Autoservicios, tiendas de conveniencia, farmacias, etc.) se cuenta con información detallada, de relativa alta frecuencia y excelente confiabilidad del desempeño de cada una de las marcas en cada una de las distintas categorías. la idea de este trabajo es generar un proceso que integre múltiples especificaciones de modelos econométricos con las metodologías de validación por estimación del error de predicción propias de aprendizaje de maquina en contextos no tradicionales (para el econometrista) de grandes bases de micro-datos consolidados para dar soluciones de negocio al problema de *pricing*. Por lo tanto, el alcance de esta investigación es predecir la respuesta esperada de las variables dependientes, las demandas por marca, ante impulsos de ciertas variables exógenas -desde el punto de vista del consumidor- los precios.

A pesar de que existe una gran literatura sobre cómo modelar sistemas de demanda [2-11], éstos se desenvuelven en contextos con pocos datos y gran nivel de agregación (e.g. información anual a nivel categorías durante 10 años), mientras que el presente proyecto utiliza información desagregada a nivel marca para una sola categoría, con una periodicidad semanal para dos años de observaciones. Estas especificaciones requieren implementar adaptaciones a los modelos propuestos hasta la fecha. Finalmente el trabajo emplea la técnica de descenso en gradiente para ajustar los distintos modelos propuestos y obtiene estadísticas útiles sobre cada uno de los distintos modelos, empleando la estrategia de particionar la muestra en entrenamiento, validación y prueba, para elegir la especificación que mejor modele el mercado analizado. El mejor modelo permite construir un simulador de escenarios de precios que le facilita a la empresa visualizar el efecto sobre su portafolio de las distintas estrategias de *pricing* consideradas.

Buscamos modelar la demanda de un mercado cuasi-diferenciado en un contexto de alta concentración utilizando teoría microeconómica, a saber, el contexto de un agente representativo que optimiza su ingreso eligiendo las cantidades de las diferentes marcas disponibles en el mercado y considerando sus preferencias sobre los productos. Como parte de este proceso asumiremos que semana a semana el agente representativo realiza un proceso de optimización en el que elige cantidades de cada uno de las diferentes marcas en el mercado observando los precios de cada uno de ellos y considerando su ingreso disponible. Las cantidades observadas -los datos-, serán entonces consideradas como las elecciones óptimas del agente representativo en función de variables como los precios y el índice de precios.

La presente investigación difiere al los estudios de los autores antes mencionados [1-11] en el hecho de que aquí se busca resolver una pregunta de negocios para una temporalidad futura especifica (e.g. los siguientes 3 meses), lo cual nos lleva a replantear el concepto de modelo "óptimo" a términos de la predicción misma, en lugar de su comportamiento asintótico. Adicionalmente, debido a la dinámica de la recolección de la información -mensual-, buscamos que este proceso se realice de manera periódica (e.g. una vez al año) con el fin de incluir esta herramienta en el proceso anual de toma de decisiones de las empresas.

La metodología compara la propuesta realizada en esta investigación de implementación con aquella propuesta por [12] a términos del error de predicción en las bases, tanto de blancifort (publicada en librería R "micEconAids"), como nuestra base de cereales (privada). Después haremos modificaciones a algunos modelos analizados en [12] para incluir variables de ejecución como la distribución y los inventarios promedios y compararemos estos modelos entre sí y contra los escenarios base- todo a términos del error de predicción.

El resto del documento está organizado como sigue: En la sección II se describe brevemente la base de datos utilizada, la sección III contiene el marco teórico. En la sección IV, describimos el desarrollo del proyecto, mientras que en la sección V se muestran los resultados obtenidos. Cerramos con conclusiones y trabajo futuro en la sección VI.

II. Sobre la Base de Datos

La información que utilizaremos es una base de datos recolectada de las principales tiendas de autoservicios del país a nivel semana con información de ventas en pesos kilos unidades a nivel marca (i.e. ventas en pesos para la primer semana de 2018 de la marca "x") junto con variables de ejecución, es decir distribución numérica semanal (el numero de tiendas en las cuales se vendió el producto como porcentaje del total de las tiendas) inventario promedio por tienda, distribución ponderada (el porcentaje de cobertura que tienes en las tiendas con respecto a la proporción de ventas que estas representan para la categoría). La poseemos información de 105 semanas y 18 marcas.

III. Marco teórico

Esta sección describe brevemente el modelo teórico propuesto en [13,14], así como en [12]. Supongamos que existe un agente representativo que minimiza su gasto sobre una canasta de bienes (en nuestro caso las marcas de la categoría de cereales) observando el precio de cada uno de los bienes y sujeto a un nivel de utilidad mínimo requerido. Considere el siguiente problema: (1)

$$\begin{aligned} \underset{x_1,\dots,x_M}{\text{minimizar}} \quad & \sum_{i\in M} p_m * X_m \\ \text{sujeto a} \quad & U(X_i) \geq U_0 \end{aligned}$$

Donde $\{x_1,..,x_M\}$son las cantidades elegidas por el agente representativo de la variedad (marca) 1 a M, U_0 es un numero positivo que representa el nivel de utilidad mínimo que el agente requiere de su canasta óptima $U(x)\ x \in \mathscr{R}^M \to \mathscr{R}^{++}$ entonces $U()$ es una función que mapea de la canastas disponibles para el agente a los números positivos (para cada posible combinación de productos el agente recibe una utilidad). Dado un nivel de utilidad U_0, un vector de precios y asumiendo una forma funcional adecuada para la función de utilidad, es posible encontrar una forma analítica para la función objetivo en función de los precios, el nivel de utilidad y ciertos coeficientes. Esta función es llamada la función de egresos y está definida de la siguiente manera:
(2)

$$\ln(m_t) = \ln(e(p_t, U_t)) =$$

$$\alpha_0 + \sum_i \alpha_i * \ln(p_{i,t}) + \frac{1}{2}\sum_i \sum_j \gamma^*_{i,j} \ln(p_{i,t}) \ln(p_{j,t}) + U_t \beta_0 \prod_i p_{i,t}{}^{\beta_i}$$

Donde m_t es el gasto total en el mercado al tiempo t,$e(p_t,U_t)$ es la funcion de gasto optimo en funcion del precio y del nivel de utilidad base en el periodo t ,$p_{i,t}$ es el precio del bien i al tiempo t, U_t es el nivel de utilidad al tiempo t, y α, β, γ^* son vectores de coeficientes.

Al aplicar el lema de Shepard a la función de egresos óptima y sustituir el nivel inobservable de utilidad por la función de utilidad indirecta que le corresponde a la función de gastos propuesta en [13] podemos obtener las demandas Marshallianas:

(3)

$$x_{i,t}(p_t,m_t) = \frac{m_t}{p_{i,t}}(\alpha_i + \sum_j \gamma_{i,j}\ln(p_{j,t}) + \beta_i \ln(\frac{m_t}{P_t}))$$
$$\forall i \in M, t \in T$$

Donde $x_{i,t}$ reperesenta la cantidad consumida del bien i al tiempo t y P_t es el índice de precios translog (translog price index), que se define de la siguiente manera:
(3)

$$\ln(P_t) = \alpha_0 + \sum_i \alpha_i \ln(p_{i,t}) + \frac{1}{2}\sum_i\sum_j \gamma_{i,j}\ln(p_{i,t})\ln(p_{j,t})$$

Este sistema de ecuaciones puede simplificarse más al ser expresado en función de proporciones de mercado (Share of Market) de la siguiente manera:
(4)

$$s_{i,t}(p_t,m_t) = \alpha_i + \sum_j \gamma_{i,j}\ln(p_{j,t}) + \beta_i \ln(\frac{m_t}{P_t})$$

donde $s_{i,t} = \frac{p_{i,t} * x_{i,t}}{m_t}$ corresponde al porcentaje de gasto del agente representativo en la marca i en el periodo t.

Dos características son requeridas en este modelo[3]. La primera asegura que la suma de las partes de mercado estimadas en cada periodo sume 1, conocida como "Adding up condition". Esta condición se satisface automáticamente por que los $w_{i,t}$ observados suman a uno en cada periodo [12]. La segunda condición se impone a través de la función de perdida y el gradiente; esta condición requiere que los coeficientes gamma sigan la siguiente dinámica:
(simetría)

$$\forall i, j \in M$$
$$\gamma_{i,j} = \gamma_{j,i}$$

A. Estimación del Modelo

Para la estimación procedemos a sustituir de la ecuación (4) la variable inobservable -y determinística- $s_{i,t}$ por su análogo observable -y estocástico- en la muestra $w_{i,t}$. Adicionalmente incluimos términos de error para cada ecuación o marca en cada periodo llamémoslos, u_{it}. Así definimos el siguiente sistema de ecuaciones:
(6)

$$w_{i,t}(p_t,m_t) = \alpha_i + \sum_j \gamma_{i,j}\ln(p_{j,t}) + \beta_i \ln(\frac{m_t}{P_t}) + u_{i,t}; i = 1,\ldots,M.$$

Donde M es el número de marcas en la categoría y T es la cantidad total de periodos observados. En nuestro caso, se tienen 16 marcas y 152 semanas.
Nótese que este sistema de ecuaciones es no lineal únicamente en el índice de precios translog por lo que en [13] se propone la siguiente especificación:
(7)

$$w_{i,t}(p_t,m_t) = \alpha_i^S + \sum_j \gamma_{i,j}^S \ln(p_{j,t}) + \beta_i^S \ln(\frac{m_t}{P_t^S}) + u_{i,t}^S; i = 1,\ldots,M.$$

$$\ln(P_t^S) = \sum_{j\in M} w_{j,t}\ln(p_{j,t})$$

Donde $\ln(P_t^S)$ es conocido como el índice Stone.

Un problema para la especificación del modelo [7] es la simultaneidad; esto es, la estimación del índice Stone contiene a las partes de mercado $w_{i,t}$ lo que implica que este vector está en el lado derecho e izquierdo de las ecuaciones. La correlación entre el covariante $\ln(x_i/P_t)$ y el error $u_{i,t}$ genera una estimación sesgada por simultaneidad [15]. En [15] y [16] proponen corregir este problema empleando los porcentajes de mercado rezagados en vez de los contemporáneos al momento de estimar el índice translog:
(8)

$$\ln(P_t^{SL}) = \sum_{j\in M} w_{j,t-1} ln(p_{j,t})$$

Finalmente consideraremos otra especificación de modelo para aproximar el índice translog propuestas en la literatura: el ínidice Paasche:
(9)

$$\ln(P_t^P) = \sum_{m\in M} w_{k_0,t} * \ln(\frac{p_{m,t}}{p_{k_0,t}})$$

Donde w_{ik_0} son proporciones de mercado base (como por ejemplo el promedio) y p_{ik_0} son precios base (como por el ejemplo el promedio)
Recordemos que todas estas implementaciones de modelos son, finalmente, aproximaciones lineales a un modelo teórico no lineal, por lo que en [17] proponen una aproximación lineal iterativa al problema no lineal AIDS. A este estimador le llaman ILLE. En [17][18] y [19] prueban que el modelo no lineal AIDS puede ser aproximado por iteraciones de sistemas lineales. El proceso es el que sigue: En un primer inicio se calculan las ecuaciones de SoM (Share of Market), dejando el estimador del índice translog fijo; una vez obtenida la estimación se emplean los coeficientes para recalcular el índice translog. Con esta nueva estimación del índice se repite el proceso hasta que los valores de los coeficientes se estabilizan. Para los valores iniciales de los coeficientes del índice translog en [12] proponen usar los valores de los coeficientes obtenidos vía alguna de las especificaciones para la aproximación lineal del modelo AIDS.

Este proyecto adapta los modelos mencionados al contexto y metodologías de aprendizaje de máquina. A continuación se muestra la función de perdida propuesta para solucionar el problema, así como el gradiente y se discute la modificación que se realiza al sistema para garantizar la ecuación (simetría)

B. El método de aprendizaje de máquina.

Aprendizaje de máquina es una disciplina de inteligencia artificial. El proceso de aprendizaje de máquina consiste en generar un algoritmo iterativo con una regla de aprendizaje que conduzca al sistema a encontrar una solución óptima al problema requerido, en nuestro caso, ajustar el sistema de ecuaciones al vector observado de partes de mercado. Primero, necesitamos una función de pérdida que cuantifique la calidad de cada una de las predicciones en función del dato observado y de los parámetros utilizados en la estimación. Segundo, necesitamos un algoritmo de optimización para minimizar esta función eligiendo sus parámetros. El enfoque de aprendizaje de máquina tiende a ser muy orientado a los datos, al calificar el desempeño en función del ajuste del modelo a nuevos datos (errores de predicción). A continuación se muestra la función de pérdida que utilizaremos: (10)

$$f(\alpha,\gamma,\beta|(p_t)_{t=1}^{T}(m_t)_{t=1}^{T}) = \frac{1}{2} * \frac{\sum_{t\in T}\sum_{j\in M}(w_{m,t}-\hat{w}_{m,t})^2}{T*M}$$

Dado el hecho de que las partes de mercado observadas siempre suman cero, la condición de suma (adding up condition) se satisface automáticamente [14] lo que implica que: $\sum_i \alpha_i = 1, \sum_i \beta_i = 0, \sum_i \gamma_{i,j} = 0, \forall i,j$ y los términos del error, siempre suman 0 (en cada periodo) $\sum_{m\in M} u_{mt} = 0; \forall t \in T$ y por lo tanto la matriz de covarianzas de los errores es singular (Blanciforti and Green 1983, p. 512). Para evitar problemas de estimación debido a esta singularidad una de las ecuaciones debe ser desechada del sistema de ecuaciones en ecuación (7). Si se desea, los coeficientes de la ecuación desechada se pueden recuperar gracias a las condiciones de suma. En [20] se muestra que al desechar una ecuación arbitraria no se pierde ninguna información, por lo que los resultados de la estimación no dependen de que ecuación se deseche.

La vectorización del sistema de ecuaciones a estimar es la siguiente:

$$\hat{w}_t = \alpha + \gamma * \ln(p_t) + \beta .* \ln(m_t) - \beta .* \ln(P_t)$$

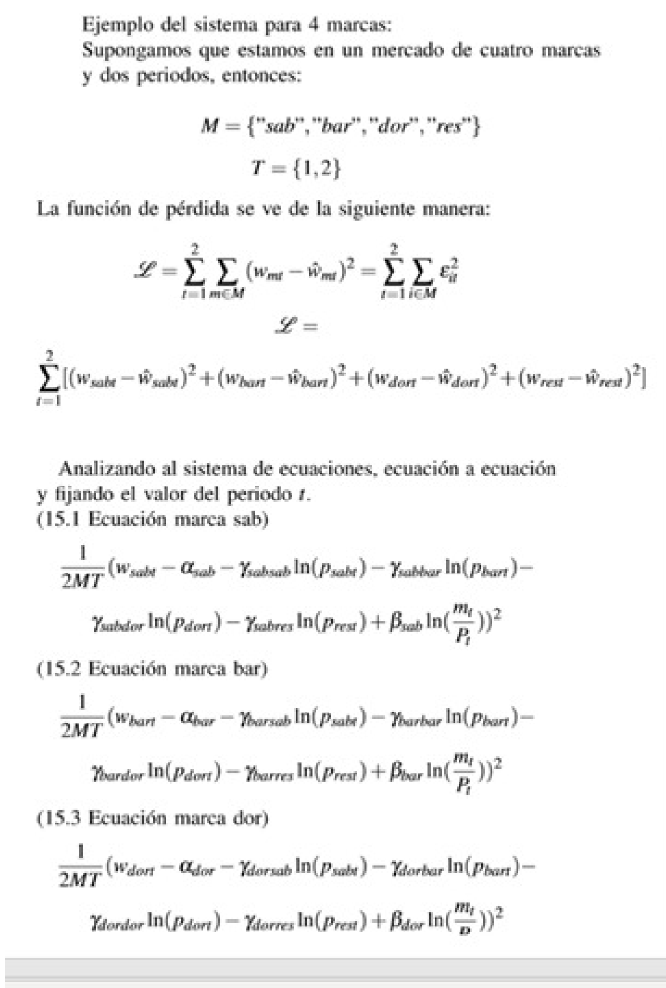
Donde α es un vector de dimensión $M-1x1$, γ es una matriz simétrica (puesto que se impuso la condición $\forall i,j; \gamma_{i,j} = \gamma_{j,i}$ de dimensión $(M-1)xM$ (a la matriz simmetrica γ le quitamos la ultima fila) y β es un vector de $(M-1)x1$ además $.*$ define a la operación producto elemento a elemento.

Al momento hemos defindo a $\hat{w}_t$ como un vector con M-1 elementos. La parte de mercado estimada para la marca M se define de la siguiente manera:

$$\hat{w}_{M,t} = 1 - \sum_{j=1}^{M-1} \hat{w}_{j,y}; \forall t \in T$$

Notemos un par de cosas: En primer lugar se ha separado el problema de cómo estimar a la parte no lineal del sistema (i.e., el índice translog) del problema de estimar el sistema -lineal- de ecuaciones simultáneas. En todos los métodos propuestos se estima linealmente el sistema una vez aproximado el índice translog (incluso en el caso del método ILLE), por lo que esta función de error genérica nos servirá para todas las implementaciones. Por otro lado, notemos también que hemos incluido la condición de simetría dentro de la función de pérdida al requerir que la matriz γ sea simétrica.

Ejemplo del sistema para cuatro marcas:


Ejemplo del sistema para 4 marcas:
Supongamos que estamos en un mercado de cuatro marcas y dos periodos, entonces:

$$M = \{"sab","bar","dor","res"\}$$

$$T = \{1,2\}$$

La función de pérdida se ve de la siguiente manera:

$$\mathscr{L} = \sum_{t=1}^{2}\sum_{m\in M}(w_{mt}-\hat{w}_{mt})^2 = \sum_{t=1}^{2}\sum_{i\in M}\varepsilon_{it}^2$$

$$\mathscr{L} = \sum_{t=1}^{2}[(w_{sabt}-\hat{w}_{sabt})^2 + (w_{bart}-\hat{w}_{bart})^2 + (w_{dort}-\hat{w}_{dort})^2 + (w_{rest}-\hat{w}_{rest})^2]$$

Analizando al sistema de ecuaciones, ecuación a ecuación y fijando el valor del periodo t.

(15.1 Ecuación marca sab)

$$\frac{1}{2MT}(w_{sabt} - \alpha_{sab} - \gamma_{sabsab}\ln(p_{sabt}) - \gamma_{sabbar}\ln(p_{bart}) - \gamma_{sabdor}\ln(p_{dort}) - \gamma_{sabres}\ln(p_{rest}) + \beta_{sab}\ln(\frac{m_t}{P_t}))^2$$

(15.2 Ecuación marca bar)

$$\frac{1}{2MT}(w_{bart} - \alpha_{bar} - \gamma_{barsab}\ln(p_{sabt}) - \gamma_{barbar}\ln(p_{bart}) - \gamma_{bardor}\ln(p_{dort}) - \gamma_{barres}\ln(p_{rest}) + \beta_{bar}\ln(\frac{m_t}{P_t}))^2$$

(15.3 Ecuación marca dor)

$$\frac{1}{2MT}(w_{dort} - \alpha_{dor} - \gamma_{dorsab}\ln(p_{sabt}) - \gamma_{dorbar}\ln(p_{bart}) - \gamma_{dordor}\ln(p_{dort}) - \gamma_{dorres}\ln(p_{rest}) + \beta_{dor}\ln(\frac{m_t}{p}))^2$$

Fig. 1: Ejemplo del sistema.

Ahora analicemos el gradiente $\nabla f(\alpha,\gamma,\beta|(p_t)_{t=1}^{T}(m_t)_{t=1}^{T})$ este objeto puede ser visto como una colección de tres listas:

$$\nabla f(\alpha,\gamma,\beta|(p_t)_{t=1}^{T}(m_t)_{t=1}^{T}) = \{A,B,C\}$$

Tales que A B y C son listas que se definen de la siguiente manera:

$$A = \{\frac{\partial f}{\partial \alpha_1}, \ldots, \frac{\partial f}{\partial \alpha_{M-1}}\}$$

$$B = \{\frac{\partial f}{\partial \gamma_{1,1}}, \ldots, \frac{\partial f}{\partial \gamma_{M-1,M-1}}, \frac{\partial f}{\partial \gamma_{1,2}}, \ldots,$$

$$\ldots \frac{\partial f}{\partial \gamma_{1,M}}, \frac{\partial f}{\partial \gamma_{2,3}}, \ldots, \frac{\partial f}{\partial \gamma_{2,M}}, \ldots, \frac{\partial f}{\gamma_{M-1,M}}\}$$

$$C = \{\frac{\partial f}{\partial \beta_1}, \ldots, \frac{\partial f}{\partial \beta_{M-1}}\}$$

Note que los primeros $M-1$ elementos de la lista B son los elementos asociados a las derivadas de los coeficientes de la diagonal de la matriz γ. La definición de las derivadas está contenida dentro de las funciones anexas en la sección III. Dado la función de pérdida genérica así como el gradiente es fácil ver como este modelo es lo suficientemente flexible para estimar las especificaciones de los modelos con índices Stone, Stone rezagado, Paasche e ILLE. En cada caso se comienza calculando el índice de precios -en el caso de ILLE se usan los coeficientes α, γ, β tenidos de alguno de los otros métodos- para calcular la primera iteración del índice translog (ecuación (3)). Después se usa descenso en gradiente para ajustar por los nuevos coeficientes.

Para el caso de ILLE un ciclo consiste en la actualización de la aproximación del índice translog y el ajuste del sistema de ecuaciones con descenso en gradiente. Una vez obtenidos los coeficientes del ciclo estos actualizan a la estimación del índice translog y se reinicia el proceso. Este proceso se repite hasta que los coeficientes se estabilicen. En cada ciclo (e.g. al finalizar cada ciclo de descenso en gradiente), se estará guardando el valor de la función de perdida en los coeficientes del ciclo para el caso de la muestra de entrenamiento y de validación (el algoritmo de descenso en gradiente únicamente se corre sobre la muestra de entrenamiento).

Recordemos que nuestro objetivo es proponer una nueva metodología que se adapte mejor a la pregunta de negocios por lo tanto nuestra metodología no debe ser considerada una estimación econométrica tradicional sino como un proceso de aprendizaje de maquina inspirado en modelos econométricos. Como parte del ejercicio se contrastan los modelos propuestos (y ajustados) en este proyecto y los implementados por [12] entre sí -a términos del error de entrenamiento, validación y predicción- pero esta comparativa nos es interesante únicamente en la medida en la que nos permite elegir el mejor de ellos para montar sobre él las simulaciones.

IV. DESARROLLO

La implementación del presente proyecto se desarrolló con apoyo de funciones en lenguaje R, cuyos códigos implementados son los siguientes:

- La siguiente función calcula el número de coeficientes gammas que tendrá el modelo como funciones de las marcas totales en el mercado:

```
num_gammas <- function(names_prices){
  j<- length(names_prices)
  aux<-0
  coefs<-0
  for(m in 0:j-2){
    aux <-  j-m
    if(m==0){
      coefs <- aux
    }else{
      coefs <- coefs + aux
    }
  }
  return(coefs)#esto es un numero
}
```

- La siguiente función calcula los nombres de los coeficientes y si estos pertenecen o no a la diagonal de la matriz gamma:

```
nombres_gammas_plus_one <- function(names_prices){
  txt <- "gamma_"
  f <- rep("vacio",5)
  effecto_propio <- rep(FALSE,num_gammas(names_prices))
  products<-names_prices
  x <- 1
  for(i in 1:length(products)){
    for(j in i :length(products)){
      if(i==j){
        effecto_propio[x] <-  TRUE
      }
      txt2 <- paste(products[i],products[j], sep= "_")
      f[x]<- paste0(txt,txt2)
      #print(f[x])
      x<-x+1
    }
  }
  ajuste <- x-1
  resultado <- f[1:ajuste]
  ajuste2 <- length(resultado)
  effecto_propio <- effecto_propio[1:ajuste2]
  names(effecto_propio)<- resultado
  return(effecto_propio)  #esto es una lista con nombres
}
```

- Esta función regresa una matriz de errores, donde el número de renglones es igual al número de observaciones y las columnas de las marcas:

```
errores_aids <- function(datos_proc,short_names,SoM_last_ec){
  nombres_coef_gam <- nombres_gammas_plus_one(short_names)
  temps <- datos_proc$fecha
  obser <- length(unique(datos_proc$fecha))
  covariantes <- select(datos_proc,short_names)
  constante <- select(datos_proc, const)
  indice <- select(datos_proc, covBeta)
  cov_fin <- cbind(constante,covariantes)
  cov_fin$fecha <- NULL
  cov_fin$fecha1 <- NULL
  cov_fin <- cbind(cov_fin,indice)
  cov_fin$fecha <- NULL
  cov_fin_m <- cov_fin %>% as.matrix()
  nombres_ec <- short_names[1:length(short_names)-1]
  estimaciones_er <- function(coeficientes_iter){
    #a partir de la lista de coeficientes generamos la matriz de
    #alga gamma y beta
    alfa_iter <- coeficientes_iter[[1]]
    beta_iter <- coeficientes_iter[[3]]
    gamma_list_iter <- coeficientes_iter[[2]]
    #genero la matriz gamma
    gamma_diag <- gamma_list_iter[which(names(gamma_list_iter)==TRUE)]
    gamma_rest <- gamma_list_iter[which(names(gamma_list_iter)==FALSE)]
    m <- matrix(NA, ncol = length(gamma_diag), nrow = length(gamma_diag))
    m[lower.tri(m)] <-gamma_rest
    m[upper.tri(m)] <- t(m)[upper.tri(t(m))]
    #m <- t(m)
    diag(m) <- gamma_diag
    #quito la ultima ecuacion
    gammas_mat_iter <- m[1:length(short_names)-1,] %>% as.matrix()
    td_aux <- cbind(alfa_iter,gammas_mat_iter)
    td_aux <- cbind(td_aux,beta_iter)
    td_aux <- do.call("rbind",replicate(obser,td_aux,simplify = FALSE))
    Mat_est <- cov_fin_m*td_aux
    Mat_est <- t(t(rowSums(Mat_est)))
    auxiliar_p <-Mat_est[,1]
    estimados <- tibble(
      fechas = datos_proc$fecha,
      prediccion= auxiliar_p,
      observados =datos_proc$SoM,
      short_names=datos_proc$ec_nom
    )
    estimados$error<-estimados$observados- estimados$prediccion
    #estimados <- estimados %>% group_by(fechas)
        %>% mutate(short_names=nombres_ec)
    error_by_brand <- estimados %>% ungroup()
        %>% select(c(fechas,error,short_names))
        %>% spread(short_names,error)

    last_ec <- estimados %>% group_by(fechas)
        %>% summarise(otros=1-sum(prediccion))
    names(SoM_last_ec)[1]<-"fechas"
    last_ec <- left_join(last_ec,SoM_last_ec)
        %>% mutate(otros_e=SoM-otros)
        %>% select(c(fechas,otros_e))
    names(last_ec)[2]<- "otros"
    errores_marca <- left_join(error_by_brand,last_ec)
    return(errores_marca)
  }
  return(estimaciones_er)
}
```

- Esta función se encarga de realizar el cálculo del gradiente:

```
gradiente_aids <- function(datos_proc,short_names,SoM_last_ec){
  #instancio el la funcion de error
  erreur <- errores_aids(datos_proc,short_names,SoM_last_ec)
  precios_log <- select(datos_proc,short_names)
  num_marcas <- length(short_names)
  alfa_grad<- rep(NA,num_marcas-1)
  beta_grad <-rep(NA,num_marcas-1)
  gamma_grad <- rep(NA,length(nombres_gammas_plus_one(short_names)))
  obs <- length(unique(datos_proc$fecha))
  marcas_fin <- setdiff(short_names,"otros")
  num_finales <- length(marcas_fin)
  indice_td <- tibble(
    fecha=unique(datos_proc$fecha),
```

```r
      covBeta = unique(datos_proc$covBeta)
  )
  precios_per <- precios_log %>% group_by(fecha) %>% summarise_all(max)

  gradiente <- function(coeficientes_iter){
    mat_errors <-erreur(coeficientes_iter)
    alfa_grad <- rep(NA,num_finales)
    beta_grad <- rep(NA,num_finales)
    for(s in 1:num_finales){
      aux_alf <- short_names[s]
      alfa_grad[s]<- (1/obs)*sum(-mat_errors[,aux_alf]+mat_errors[,"otros"])
      beta_grad[s]<-(1/obs)*sum(indice_td[,"covBeta"]
                     *(-mat_errors[,aux_alf]+mat_errors[,"otros"]))
    }
    ac_g<-1
    ac_s<-1
    names(gamma_grad)<-rep(FALSE,length(gamma_grad))
    for(i in 1:num_marcas){
      for(j in i:num_marcas){
        if(i==j){
          names(gamma_grad)[ac_g]<-TRUE
          if(j==num_marcas){
            gamma_grad[ac_g]<-0

          }else{
            nomb <- short_names[ac_s]
            gamma_grad[ac_g]<-(1/obs)*sum(precios_per[,nomb]
                          *(-mat_errors[,nomb]+mat_errors[,"otros"]))
          }
          ac_s<-ac_s+1
        }else{
          nomb1<- short_names[i]
          nomb2<- short_names[j]
          gamma_grad[ac_g]<-(1/obs)*sum( -mat_errors[,nomb1]
*precios_per[,nomb2]-mat_errors[,nomb2]
                          *precios_per[,nomb1]
                          +mat_errors[,"otros"]*(precios_per[,nomb1]
                          +precios_per[,nomb2]))

        }
        ac_g <- ac_g+1
      }
    }
    grad_list <- list(alfa_grad,gamma_grad,beta_grad)
    return(grad_list)
  }
  return(gradiente)
}
```

- Esta función realiza el descenso en gradiente:

```r
#descenso en gradiente
library(tidyverse)
setwd("C:/Users/Alejandro/Documents/Tesis_Maestria")
source("funciones_modelo.R")
datos_proc_CDMX <- readRDS("datos_proc_CDMX.rds")
names(datos_proc_CDMX)[21] <- "covBeta"
SoM_last_ec_CDMX <- readRDS("SoM_last_ec_CDMX.rds")
short_names_CDMX <-readRDS("short_names_CDMX.rds")
tiempos <- datos_proc_CDMX %>% group_by(fecha, muestra) %>% summarise(partes=sum(SoM))
SoM_last_ec_CDMX<- left_join(SoM_last_ec_CDMX, select(tiempos,c(fecha,muestra)))

#inicializo en sero la lista de coeficientes alfa gamma y beta
las <- nombres_gammas_plus_one(short_names_CDMX)
tabla_gamma <- tibble(
  gammas_name = names(las),
  own_efect = las,
  val_coef = rep(0,length(las))
)

alfa1<- rep(0,17)
beta1 <- rep(0,17)
gamma1 <- rep(0,length(tabla_gamma$val_coef))
names(gamma1)<- tabla_gamma$own_efect
coeficientes_iter<-list(alfa1,gamma1,beta1)

descenso_gradiente <- function(num_iter,coef_init,lrng_rate,grad_model,
                        fun_error,fun_error_val){
  error <- tibble(intentos = c(1:num_iter),val_error=NA)
  error_validacion <- tibble(intentos = c(1:num_iter),val_error=NA)
  coef_viejo <-  coef_init
  #calculo el error de entrenamiento para la primer iteracion de los coeficientes
  aux_e<- fun_error(coef_viejo) %>% ungroup
  aux_e<- select(aux_e,-fechas)
  aux_e <- sum(aux_e*aux_e)/(74*18)
  #calculo el error de validacion para la primer iteracion de los coeficientes
  aux_e_val <-fun_error_val(coef_viejo) %>% ungroup
  aux_e_val <- select(aux_e_val,-fechas)
  aux_e_val <- sum(aux_e_val*aux_e_val)/(16*18)
  #guardo el error de entrenamiento en la tabla error
  error[1,"val_error"]<- aux_e
  #guardo el error de validacion en la tabla error_validacion
  error_validacion[1,"val_error"]<- aux_e_val
  #guardo la lista de los coeficientes finales
  best_coef <- coef_init
  aux_txt_1 <- paste("error entrena:", aux_e)
  aux_txt_2 <- paste("error valida:", aux_e_val)
  aux_txt_3 <- paste(aux_txt_1,aux_txt_2)
  aux_txt_1 <- paste("iteracion num",1)
  aux_txt_3 <- paste(aux_txt_1,aux_txt_3)
  print(aux_txt_3)
  for(i in 2:num_iter){
    #calculo el gradiente
    gradiente <- grad_model(coef_init)
    coef_nuevo <- coef_viejo
    coef_nuevo[[1]] <- coef_viejo[[1]] - lrng_rate*gradiente[[1]]#para alfa
    coef_nuevo[[2]] <- coef_viejo[[2]] - lrng_rate*gradiente[[2]]#para gamma
    coef_nuevo[[3]] <- coef_viejo[[3]] - lrng_rate*gradiente[[3]]#para beta
    #calculo del nuevo error de entrenamiento
    aux_e<- fun_error(coef_nuevo) %>% ungroup
    aux_e<- select(aux_e,-fechas)
    aux_e <- sum(aux_e*aux_e)/(74*18)
    #calculo del nuevo error de validacion
    aux_e_val <-fun_error_val(coef_nuevo) %>% ungroup
    aux_e_val <- select(aux_e_val,-fechas)
    aux_e_val <- sum(aux_e_val*aux_e_val)/(16*18)
    #guardo los valores en las tablas errores y errores_validacion
    error[i,"val_error"]<- aux_e
    error_validacion[i,"val_error"]<- aux_e_val
    #actulizo los nuevos coeficientes
    coef_viejo <- coef_nuevo

    if(as.numeric(error_validacion[i,"val_error"]) <
    as.numeric(error_validacion[i-1,"val_error"])){
      best_coef <- coef_nuevo
    }
    aux_txt_1 <- paste("error entrena:", aux_e)
    aux_txt_2 <- paste("error valida:", aux_e_val)
    aux_txt_3 <- paste(aux_txt_1,aux_txt_2)
    aux_txt_1 <- paste("iteracion num",i)
    aux_txt_3 <- paste(aux_txt_1,aux_txt_3)
    print(aux_txt_3)
  }
  error$muestra<- "entrenamiento"
  error_validacion$muestra <- "validacion"
 DB_errors <- rbind(error,error_validacion)
  return(list(DB_errors,best_coef))
}
datos_ent_cdmx <- datos_proc_CDMX %>% filter(muestra=="entrenamiento")
datos_val_cdmx <-datos_proc_CDMX %>% filter(muestra=="validacion")
SoM_last_ec_ent_cdmx <-filter(SoM_last_ec_CDMX, muestra=="entrenamiento") %>%
ungroup %>% select(c(fecha,SoM))
SoM_last_ec_val_cdmx <-filter(SoM_last_ec_CDMX, muestra=="validacion")
%>% ungroup %>% select(c(fecha,SoM))
fun_error_ent<- errores_aids(datos_ent_cdmx,short_names_CDMX,SoM_last_ec_ent_cdmx)
grad_model <- gradiente_aids(datos_ent_cdmx,short_names_CDMX,SoM_last_ec_ent_cdmx)
fun_error_val <-errores_aids(datos_val_cdmx,short_names_CDMX,SoM_last_ec_val_cdmx)
shala <- descenso_gradiente(1000,coeficientes_iter,0.0000001,
        grad_model,fun_error_ent, fun_error_val)
db <-shala[[1]]
opt_coef <- shala[[2]]
ggplot(db, aes(x=intentos, y= val_error, color= as.factor(muestra))) +
  geom_line() +
  theme_minimal() +
  labs(x= "Iteraciones", y="Error de Entrenamiento")
ggplot(db, aes(x=intentos, y= val_error)) +
  geom_line() +
  theme_minimal() +
  labs(x= "Numero iteraciones", y="Error por iteracion") + facet_wrap(~muestra)
# presento en una tabla los coeficientes
alf_nom <- rep(NA,17)
bet_nom <- rep(NA,17)
for(j in 1:17){
  alf_nom[j]<-paste0("alfa_",short_names_CDMX[j])
  bet_nom[j]<-paste0("beta_",short_names_CDMX[j])
}
gam_nom <-nombres_gammas_plus_one(short_names_CDMX)
gam_nom <- names(gam_nom)
alf <- tibble(coef_name= alf_nom,
              coef_val=opt_coef[[1]] )
bet <- tibble(coef_name= bet_nom,
              coef_val=opt_coef[[3]] )
gam <- tibble(coef_name= gam_nom,
              coef_val=opt_coef[[2]] )
DB_Coeficientes <- rbind(alf,gam)
DB_Coeficientes <- rbind(DB_Coeficientes,bet)
#comparo con la libreria micEcon_AIDS
Base <-readRDS("base.rds")
cat_brands <- c("ab","cher","chkr","clab","cook","crnf",
    "crnp","sqr","ex","fit","frotl","azuc","nesqk","nescf",
    "munif","otros","spk","zuc")
catalogo_marcas <- tibble(
  marca=unique(Base$marca),
  short_names=cat_brands
)
Tabla <- left_join(Base, catalogo_marcas)
Tabla$prxkg <- Tabla$s/Tabla$kg
letemps <- unique(Tabla$fecha)
fecha_ent <- letemps[1:74]
fecha_val <- letemps[74:90]
fecha_prueba <- letemps[90:105]
Tabla <- ungroup(Tabla)
Price_EconAids <- Tabla %>% filter(fecha %in% fecha_ent)
    %>% select(c(fecha,prxkg,short_names))
    %>% spread(key=short_names,value=prxkg)
SoM_EconAids <- Tabla %>% filter(fecha %in% fecha_ent)
    %>% group_by(fecha) %>% mutate(SoM=s/sum(s)) %>% ungroup()
    %>% select(c(fecha,SoM,short_names))
    %>% spread(key=short_names,value=SoM)
aux_pr<- names(Price_EconAids)
aux_som<-names(SoM_EconAids)
for(i in 2:dim(Price_EconAids)[2]){
  aux_pr[i]<-paste0("pr_",aux_pr[i])
  aux_som[i]<-paste0("SoM_",aux_som[i])
}
Gasto <- Tabla %>% filter(fecha %in% fecha_ent) %>% group_by(fecha) %>% summarise(gasto = sum(s))
names(Price_EconAids) <- aux_pr
names(SoM_EconAids) <- aux_som
Base_MicEconAids <- left_join(Price_EconAids,SoM_EconAids)
Base_MicEconAids <- left_join(Base_MicEconAids,Gasto)
DatosEcon <- as.data.frame(Base_MicEconAids)
price_names <- aux_pr[2:length(aux_pr)]
share_names <- aux_som[2:length(aux_som)]
laaidsResultP <- aidsEst( price_names, share_names, "gasto", data = DatosEcon, priceIndex = "P" )
prediccionesEconAids <- predict(laaidsResultP, newdata = NULL, observedShares = T)
SoM_pred_MicEcon <- prediccionesEconAids[[1]]
SoM_pred_MicEcon <- cbind(fecha_ent,SoM_pred_MicEcon)
donnes <- SoM_pred_MicEcon %>% gather(nombres,SoM_Aids,-fecha_ent)
cat_Econ <- tibble( nombres=unique(donnes$nombres),
                    short_names = cat_brands
)
SoM_pred_MicEcon <- left_join(donnes,cat_Econ)
names(SoM_pred_MicEcon)[1] <-"fecha"
ObservacionesEcon <- Tabla %>% filter(fecha %in% fecha_ent) %>% group_by(fecha)
```

```r
    %>% mutate(SoM=s/sum(s)) %>% ungroup()
    %>% select(c(fecha,SoM,short_names))
SoM_pred_MicEcon <-left_join(SoM_pred_MicEcon,ObservacionesEcon)
SoM_pred_MicEcon$error <- (SoM_pred_MicEcon$SoM-SoM_pred_MicEcon$SoM_Aids) ^2
sum(SoM_pred_MicEcon$error)/(74*18)
## validacion
Price_EconAids_val <- Tabla %>% filter(fecha %in% fecha_val)
    %>% select(c(fecha,prxkg,short_names))
    %>% spread(key=short_names,value=prxkg)
SoM_EconAids_val <- Tabla %>% filter(fecha %in% fecha_val)
    %>% group_by(fecha) %>% mutate(SoM=s/sum(s)) %>% ungroup()
    %>% select(c(fecha,SoM,short_names))
    %>% spread(key=short_names,value=SoM)
aux_pr_val<- names(Price_EconAids)
aux_som_val<-names(SoM_EconAids)
Gasto_val <- Tabla %>% filter(fecha %in% fecha_val) %>% group_by(fecha)
    %>% summarise(gasto = sum(s))
names(Price_EconAids_val) <- aux_pr_val
names(SoM_EconAids_val) <- aux_som_val
Base_MicEconAids_val <- left_join(Price_EconAids_val,SoM_EconAids_val)
Base_MicEconAids_val <- left_join(Base_MicEconAids_val,Gasto_val)
DatosEcon_val <- as.data.frame(Base_MicEconAids_val)
prediccionesEconAids_val <- predict(laaidsResultP, newdata = DatosEcon_val, observedShares = T)
SoM_pred_MicEcon_val <- prediccionesEconAids_val[[1]]
SoM_pred_MicEcon_val <- cbind(fecha_val,SoM_pred_MicEcon_val)
donnes_val <- SoM_pred_MicEcon_val %>% gather(nombres,SoM_Aids,-fecha_val)
cat_Econ <- tibble( nombres=unique(donnes$nombres),
                    short_names = cat_brands
)
SoM_pred_MicEcon_val <- left_join(donnes_val,cat_Econ)
names(SoM_pred_MicEcon_val)[1] <-"fecha"
ObservacionesEcon_val <- Tabla %>% filter(fecha %in% fecha_val)
    %>% group_by(fecha) %>% mutate(SoM=s/sum(s))
    %>% ungroup() %>% select(c(fecha,SoM,short_names))
SoM_pred_MicEcon_val <-left_join(SoM_pred_MicEcon_val,ObservacionesEcon_val)
SoM_pred_MicEcon_val$error <- (SoM_pred_MicEcon_val$SoM-SoM_pred_MicEcon_val$SoM_Aids) ^2
sum(SoM_pred_MicEcon_val$error)/(17*18)
```

V. Resultados

Después de aplicar descenso en gradiente, llegamos a los siguientes resultados, tanto en la muestra de entrenamiento como en la de validación:

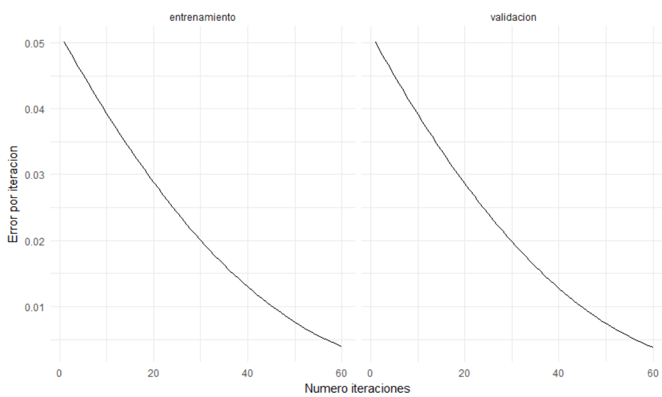


Fig. 2: Resultados obtenidos.

La gráfica anterior corresponde a el número de iteraciones (eje X) contra el error por iteración (eje Y). Podemos apreciar que conforme el número de iteraciones va aumentando, tanto el error de entrenamiento como de validación va acercándose a cero, que es nuestro valor deseado después de realizar variaciones al tamaño de *learning rate*, ajustado a un valor de: 0.000001. Con este valor comprobamos que nuestro modelo está aprendiendo coeficientes que le ayudan a generalizarse con nuevos datos.

VI. Siguientes pasos

El modelo de descenso en gradiente converge al mismo error de entrenamiento que el de nuestro modelo base (el implementado en [12]), aunque los coeficientes no son los mismos. Falta hacer una comparativa cuantitativa sobre el efecto de éstos sobre la relación entre los precios y las demandas. Sin embargo, el método de descenso en gradiente parece ser una buena solución, rápida y que sustituye al modelo econométrico de inversión de matrices.

Como trabajo futuro queda incluir las distintas especificaciones de modelo con Stone index, Stone lagged e ILLE; todos se pueden implementar con las funciones que ya tenemos, simplemente resta hacer ajustes para estimar estos índices y en el caso de ILLE, correr las iteraciones pertinentes.

Otra es incluir penalizaciones a los coeficientes para reducir la dimensionalidad del problema. Esto se puede lograr modificando la función de pérdida y el descenso en gradiente para reconocer las penalizaciones de la regularización Ridge.

References

[1] Nevo, A. (2001). ”Measuring market power in the ready-to-eat cereal industry”. Econometrica, 69(2), 307-342.

[2] Brannlud R, Ghalwash T, Nordstrom J(2007). “Increased Energy Efficiency and the Re- bound Effect: Effects on Consumption and Emissions.” Energy Economics, 29(1), 1–17.

[3] Chambwera M, Folmer H (2007). “Fuel Switching in Harare: An Almost Ideal Demand System Approach.” Energy Policy, 35(4), 2538–2548.

[4] Farrell L, Shields MA (2007). “Children as Consumers: Investigating Child Diary Expenditure Data.” Canadian Journal of Economics, 40(2), 445–467.

[5] Hausman JA, Leonard GK (2007). “Estimation of Patent Licensing Value Using a Flexible Demand Specification.” Journal of Econometrics, 139(2), 242–258.

[6] Henning CHCA, Henningsen A (2007). “Modeling Farm Households’ Price Responses in the Presence of Transaction Costs and Heterogeneity in Labor Markets.” American Journal of Agricultural Economics, 89(3), 665–681.

[7] Huang MH, Jones E, Hahn DE (2007). “Determinants of Price Elasticities for Private Labels and National Brands of Cheese.” Applied Economics, 39(5), 553–563.

[8] Moore T, Green CJ (2007). “A Portfolio Approach to Firms’ Financing Decisions: Evidence from India Using the Almost Ideal Demand System.” In V Murinde (ed.), Accounting, Bank- ing and Corporate Financial Management in Emerging Economies, number 7 in Research in Accounting in Emerging Economies, pp. 347–368. Elsevier, Amsterdam.

[9] Raknerud A, Skjerpen T, Swensen AR (2007). “A Linear Demand System within a Seemigly Unrelated Time Series Equations Framework.” Empirical Economics, 32(1), 105–124.

[10] West SE, Williams Roberton C I (2007). “Optimal Taxation and Cross-Price Effects on Labor Supply: Estimates of the Optimal Gas Tax.” Journal of Public Economics, 91(3/4), 593–617.

[11] Xiao N, Zarnikau J, Damien P (2007). “Testing Functional Forms in Energy Modeling: An Application of the Bayesian Approach to U.S. Electricity Demand.” Energy Economics, 29(2), 158–166.

[12] Henningsen, A (2017). Demand Analysis with the “Almost Ideal Demand System” in R: Package micEconAids.

[13] Deaton A, Muellbauer J (1980a). “An Almost Ideal Demand System.” The American Economic Review, 70(3), 312–326.

[14] Deaton A, Muellbauer J (1980b). Economics and Consumer Behaviour. Cambridge University Press, Cambridge.

[15] Eales JS, Unnevehr LJ (1988). “Demand for Beef and Chicken Products: Separability and Structural Change.” American Journal of Agricultural Economics, 70, 521–532.

[16] Blanciforti LA, Green RD, King GA (1986). “U.S. Consumer Behavior Over the Postwar Period: An Almost Ideal Demand System Analysis.” Monograph Number 40 (August 1986), Giannini Foundation of Agricultural Economics, University of California.

[17] Blundell RW, Robin JM (1999). “Estimation in Large and Disaggregated Demand Systems: An Estimator for Conditionally Linear Systems.” Journal of Applied Econometrics, 14(3), 209–232.

[18] Browning M, Meghir C (1991). “The Effects of Male and Female Labor Supply on Commodity Demands.” Econometrica, 59, 925–951.

[19] Michalek J, Keyzer MA (1992). “Estimation of a Two-stage LES-AIDS Consumer Demand System for Eight EC Countries.” European Review of Agricultural Economics, 19(2), 137– 163.

[20] Barten AP (1969). "Maximum Likelihood Estimation of a Complete System of Demand Equations." European Economic Review, 1(1), 7–73.