Computational Economics: Topics on Decision Making

Victor H. Aguiar

This version: January 2024

Part I

Parametric Methods

Chapter 1

Structural Modeling

The Cowles Commision defined econometrics as a "branch of economics in which economic theory and statistical methods are fused in the analysis of numerical and institutional data" Hood and Koopmans (1953). Econometrics nowadays has a broader definition, but the branch of econometrics that combines economic theories with statistical models is called structural econometric models.

1.1 The Gravity Model of Trade

We will illustrate an instance of structural modeling using an example from international trade. The gravity model of trade as formalized by Anderson and Van Wincoop (2003) is one of the most successful applications of structural modeling. The gravity model of trade tries to provide a theoretical explanation of the following empirical fact the nominal bilateral trade is directly proportional to the mass of the countries and inversely proportional to distance. This fact can be generalized to many countries.

Example 1. (Theory) Gravity model of trade.

We have I countries, with typical elements $i, j \in I$.

We assume that each country specializes in the production of only one good. The supply of each good is fixed (equivalently, each region is endowed with only positive quantity of one good, and there is no production).

If c_{ij} is the consumption by country j consumers of goods from region i, consumers in region j maximize

$$U_j(c) = \left(\sum_{i \in I} \beta_i^{(1-\sigma)/\sigma} c_{ij}^{(\sigma-1)/\sigma} \right)^{\sigma/(\sigma-1)},$$

subject to the budget constraint

$$\sum_i p_{ij} c_{ij} = y_j.$$

We have σ as the elasticity of substitution between all goods, β_i is a positive distribution parameter, y_j is a regional income of country j consumers, and p_{ij} is the price of region i good for j consumers.

Notice that p_{ij} differs per location j due to trade-costs.

Let p_i denote the exporter's supply price, net of trade costs, and $t_{ij} \geq 1$ is the trade cost factor between i, j (when $t_{ij} = 1$ then there is free-trade).

$$p_{ij} = p_i t_{ij}.$$

We assume that trade costs are absorbed by the exporter.

Formally, we assume that for each good shipped from i to j the exporter bears a costs equal to $t_{ij} - 1$ of country i goods. The exporter passes on these trade costs to the importer.

The nominal value of exports is:

$$x_{ij} = p_{ij} c_{ij}.$$

Market clearing implies here that in nominal terms endowments are equal to aggregate demand:

$$y_i = \sum_{j \in I} x_{ij}.$$

1) Show that the nominal demand for country i goods by country j consumer is:

$$x_{ij} = \left(\frac{\beta_i p_i t_{ij}}{P_j} \right)^{(1-\sigma)} y_j,$$

where P_j is the consumer price index of j, given by

$$P_j = \left[\sum_{i \in I} (\beta_i p_i t_{ij})^{1-\sigma} \right]^{1/(1-\sigma)}.$$

Important: I need step-by-step derivation.

2) Show, using the nominal market-clearing conditions and the results from 1) that:
$\beta_i p_i = \frac{y_i^{1/(1-\sigma)}}{(\sum_{j\in I}(t_{ij}/P_j)^{1-\sigma} y_j)^{1/(1-\sigma)}}$.

Important: I need step-by-step derivation.

3) Let $y^W = \sum_{j\in I} y_j$ be the world income, and income share $\theta_j = y_j/y^W$. Show that, replacing $\beta_i p_i$ on the nominal demand, we can obtain:

$$x_{ij} = \frac{y_i y_j}{y^W}\left(\frac{t_{ij}}{\Pi_i P_j}\right)^{1-\sigma},$$

where

$$\Pi_i = \left(\sum_{j\in I}(t_{ij}/P_j)^{1-\sigma}\theta_j\right)^{1/(1-\sigma)}.$$

Important: I need step-by-step derivation.

4) Assume that $t_{ij} = t_{ji}$, where trade barriers are symmetric, show that:

$$\Pi_i = P_i.$$

Under this condition, we have an implicit solution to the price indexes P_i given by

$$P_j^{1-\sigma} = \sum_{i\in I} P_i^{\sigma-1}\theta_i t_{ij}^{1-\sigma}\forall j,$$

show that there is at least one solution to this system of equations.

For this part use the file \GitHub\Microeconomics1\finalexam\theory\fixedpoint_contraction_mapping_jacobian.p
Theorem 1 and Theorem 2. In particular, show that the implicit solution to the price index P_i for all $i \in I$ for a contraction mapping. You can assume that P_i take values in a closed set for simplicity. Also note that, that you have to rewrite the price index equation in the form.

$$P_j = g_j(P_1, \cdots P_I).$$

For the gravity model g is continuously differentiable, so you can compute it's Jacobian. For the computing the norm of the Jacobian use whatever norm makes easier your computation. Euclidean or Max matrix norms are good candidates. I used the Euclidean matrix norm.

Important: Only for this item, assume that $I = \{1, 2\}$, $P_i \geq 1\forall i$, $\sigma = \frac{1}{2}, t_{ij} = 1, \theta_i = \frac{1}{2}$. Remember you have to show that the the norm of the Jacobian of the mapping g is less than 1,

for all values of P_i.

5) Show that, the gravity model implies that there are constants $\alpha_i \forall i \in I$, and $\rho = (1-\sigma)$ such that:

$$z_{ij} = logx_{ij} - log(y_i) - log(y_j)$$

$$z_{ij} = -\alpha_i - \alpha_j + \rho log(t_{ij}).$$

Problem 1. (Programming Part) The results from Problem 1, can be used here even if you did not answered correctly to the previous question. No need to proof anything here. With the previous results, we obtain the gravity system of equations characterizing the general equilibrium of trade:

$$x_{ij} = \frac{y_i y_j}{y^w}\left(\frac{t_{ij}}{P_i P_j}\right)^{1-\sigma},$$

$$P_j^{1-\sigma} = \sum_{i \in I} P_i^{\sigma-1}\theta_i t_{ij}^{1-\sigma} \forall j.$$

1) Solve this system of equations in Julia, for the parameters provided in the file /GitHub/Microeconomics1/finalexam/programming/gravity_model_parameters.jl or in the /GitHub/Microeconomics1/finalexam/programming/data t.csv, y.csv and lxhat.csv files inside this folder. Remember for this you have $I = \{1, \cdots, 30\}$.

Hint: This is a triangular system of equations. The parameters are $y_i, t_{i,j} \forall i, j \in I$ and $\sigma = 1/2$. You do not need anything else.

Hint: Say you want to minimize the following function $f(z_1, z_1 \cdots, z_I) = \sum_{j=1}^{I}(\sum_{i=1}^{I} z_i \theta_{ij})^2$ in JuMP.

```
import JuMP
import Ipopt
##Read CSV files and DataFrames management.
I=30
theta=ones(I,I)
example=JuMP.Model(Ipopt.Optimizer)
JuMP.@variable(example,z[1:I]>=0)
JuMP.@NLobjective(,Min,sum((sum( x[i]*theta[i,j] for i in 1:I))^2 for j in 1:I))
```

```
JuMP.optimize!(example)
#get the solution as a vector
xsol=JuMP.value.(x)
```

2) I will provide a dataset of: (i) perturbed nominal trade flows $log\hat{x}_{ij} = logx_{ij} + \epsilon_{ij}$, where ϵ_{ij} is a draw of a random distribution that is mean zero and pure measurement error (you can assume it is independent from the observable variables), (ii) income by country, (iii) bilateral trade barriers.

Using this dataset, estimate σ.

Hint: Write down in Julia a contrained OLS problem.

In Problem 1 we showed that the gravity model implies that there are constants $\alpha_i \forall i \in I$, and $\rho = (1 - \sigma)$ such tat:

$$z_{ij} = log\hat{x}_{ij} - log(y_i) - log(y_j)$$

$$z_{ij} = \alpha_i - \alpha_j + \rho log(t_{ij}) + \epsilon_{ij}.$$

Estimate the linear equation above using JuMP and Ipopt.

3) As the file .../parameters.jl indicates $\sigma = 1/2$, use this parameter, to simulate a counterfactual of total trade liberalization where $t_{ij}^{free} = 1$ for all $j, i \in I$.

Hint: Use part 1). Note that you have to solve the model given σ, the new t_{ij}^{free}, and incomes that have not changes $y_i \forall i$.

4)

Proof. 1) See julia file.

2) $log\hat{x}_{ij} = log(y_i) + log(y_j) - log(y^W) + (1 - \sigma)logt_{ij} - (1 - \sigma)logP_i - (1 - \sigma)logP_j + \epsilon_{ij}$

$$log\hat{x}_{ij} - log(y_i) - log(y_j) = \alpha + \beta log(\frac{t_{ij}}{P_i P_j}) + \epsilon_{ij}$$

$$z_{ij} = \alpha + \beta[log(t_{ij}) - log(P_i) - log(P_j)] + \epsilon_{ij}$$

st.

$$P_j^{\beta} = \sum_{i \in I} P_i^{-\beta} \frac{y_i}{\alpha} t_{ij}^{\beta}$$

□

1.2 Understanding Identification and Reduced Form: Supply and Demand.

Consider a simple market with aggregate demand given by $\mathbf{D} = a - bp + \mathbf{u}$, with the usual convention that bold letters represent random variables. Aggregate supply $\mathbf{S} = \alpha + \beta p + \mathbf{v}$ for a given price p. The analyst only observes price and quantities in equilibrium.

$$\mathbf{y} = \frac{a\beta + b\alpha}{b + \beta} + \frac{\beta\mathbf{u} + b\mathbf{v}}{b + \beta}$$

$$\mathbf{p} = \frac{a - \alpha}{b + \beta} + \frac{\mathbf{u} - \mathbf{v}}{b + \beta}.$$

In other words, the analyst observes $(\mathbf{y}, \mathbf{p})$.

This is the joint distribution of quantities and prices in equilibrium. Say that a naive analyst wants to obtain the structural elasticity of demand, b, from this observation.

He writes down the model:

$$\mathbf{y} = \gamma_0 + \gamma_1\mathbf{p} + \mathbf{e}.$$

He thens assumes that $E[\mathbf{e}|\mathbf{p}] = 0$. Under this we can obtain:

$$\gamma_1 = \frac{Cov(\mathbf{y}, \mathbf{p})}{Var(\mathbf{p})} = \frac{\beta\sigma_u^2 - b\sigma_v^2}{\sigma_u^2 + \sigma_v^2}.$$

It is obvious that γ_1 is not equal to β, in fact it is a weighted average of β and b and one cannot recover b from it. This is called **simultaneity bias,** which is a form of endogeneity. What we have witnessed here is a failure of identification due to simultaneity bias.

1.2.1 Moments and Instrumental Variables

Demand curves and supply curves are identifiable if there is additional variation in our observed data. For instance consider the model where we break the demand and supply shocks into an observed and an unobserved part.

$$\mathbf{u} = c_u\mathbf{x}_u + \epsilon_u$$

$$\mathbf{v} = c_v \mathbf{x}_v + \epsilon_{\mathbf{v}}.$$

Moreover, we know that $Cov(x_u, \epsilon_v) = 0$ and $Cov(x_v, \epsilon_u) = 0$.

Then we can identify b by noting that we have the following moment:

$$E[\mathbf{x}_v \epsilon_{\mathbf{u}}] = 0.$$

The IV estimator of b is:

$$b^{IV} = \frac{Cov[\mathbf{y}\mathbf{x}_{\mathbf{v}}]}{Cov[\mathbf{p}\mathbf{x}_v]}$$

Note that:

$$E[\mathbf{y}\mathbf{x}_v] = E[\frac{a\beta + b\alpha}{b+\beta}\mathbf{x}_v + \frac{\beta \mathbf{u}\mathbf{x}_{\mathbf{v}} + b\mathbf{v}\mathbf{x}_{\mathbf{v}}}{b+\beta}] = E[\frac{a\beta + b\alpha}{b+\beta}\mathbf{x}_v + \frac{b\mathbf{v}\mathbf{x}_v}{b+\beta}]$$

$$E[\mathbf{p}\mathbf{x}_{\mathbf{v}}] = E[\frac{a-\alpha}{b+\beta}\mathbf{x}_v + \frac{\mathbf{u}x_v - \mathbf{v}\mathbf{x}_v}{b+\beta}] = E[\frac{a-\alpha}{b+\beta}\mathbf{x}_v + \frac{\mathbf{v}\mathbf{x}_v}{b+\beta}]$$

$$E[\mathbf{y}]E[\mathbf{x}_{\mathbf{v}}] = [\frac{a\beta + b\alpha}{b+\beta}\mathbf{x}_{\mathbf{v}}],$$

with $E[\mathbf{u}] = E[\mathbf{v}] = 0$.

$$E[\mathbf{p}]E[\mathbf{x}_{\mathbf{v}}] = E[\frac{a-\alpha}{b+\beta}\mathbf{x}_v]$$

$$b^{IV} = \frac{Cov[\mathbf{y}\mathbf{x}_{\mathbf{v}}]}{Cov[\mathbf{p}\mathbf{x}_v]} = b.$$

Note that this is all done at the population level. In real life you will get a finite sample and will have to provide sample analogues of the the quantities above and will obtain an estimator $\hat{b}^{IV}$, that will converge to b as the sample size goes to infinity (i.e., consistent estimator). The short story about this is that identifying structural parameters often needs (i) additional exogenous variation, (ii) assuming a functional form, (iii) assuming exclusion restrictions. Of course, the credibility of your model and your identification will rely on how plausible it is that the exclusion restriction holds.

1.2.2 Two Stages Least Squares

$$\mathbf{p} = \frac{a-\alpha}{b+\beta} + \frac{\mathbf{u}-\mathbf{v}}{b+\beta}$$

$$\mathbf{u} = c_u \mathbf{x}_u + \epsilon_u$$

$$\mathbf{p} = \frac{a-\alpha}{b+\beta} + \frac{\mathbf{c_u x_u} + \epsilon_\mathbf{u} - \mathbf{v}}{b+\beta}$$

$$\mathbf{p} = \frac{a-\alpha}{b+\beta} + \frac{\mathbf{c_u}}{b+\beta}\mathbf{x}_u + \frac{\epsilon_u - \mathbf{v}}{b+\beta}$$

$$p = \alpha_1 + \beta_1 x_u + e_1$$

$$e_1 = \frac{\epsilon_u - \mathbf{v}}{b+\beta}$$

$$E[p|x_u] = \frac{a-\alpha}{b+\beta} + \frac{\mathbf{c_u}}{b+\beta}\mathbf{x}_u$$

$$p = E[p|x_u] + e_1$$

$$\mathbf{D} = a - bp + \mathbf{u},$$

$$\mathbf{D} = a - b[E[p|x_u] + \mathbf{e_1}] + \mathbf{u},$$

$$\mathbf{D} = a - bE[p|x_u] + \mathbf{u} + b\mathbf{e}_1$$

We can estimate b using $E[p]$ versus quantities, $D = Y$

$$Y = a - bE[p|x_u] + e_2$$

$$e_2 = \mathbf{u} + b\mathbf{e}_1$$

$$E[e_2 E[p|x_u]] = 0,$$

by construction.

Chapter 2

BLP

Berry, Levinshohn and Pakes (Econometrica, 1995) develop a technique for empirically analyzing demand and supply in differentiated product markets.

The econometrician observes for each good j: (s_j, p_j, x_j, z_j) market shares, market prices, product attributes and instruments.

2.1 Theory: Utility and Demand

Consumer i chooses good j if and only if $U(\zeta_i, p_j, x_j, \xi_j; \theta) \geq U(\zeta_i, p_r, x_r, \xi_r; \theta)$ for $r = 0, 1, \cdots, J$, where alternatives $r = 1, \cdots, J$ represent purchases of differentiated products. The vector (x, ξ, p) represents the observed product attributes, the unobserved (to the econometrician) product attributes, and the product prices.

Let

$$A_j = \{\zeta : U(\zeta, p_j, x_j, \xi_j; \theta) \geq U(\zeta, p_r, x_r, \xi_r; \theta) \forall r = 0, 1, \cdots, J\}$$

then market shares are

$$s_j(p, x, \xi; \theta) = \int_{\zeta \in A_j} P_0 d(\zeta).$$

A special case of this model that we will be interested here is:

$$U(\zeta_i, p_j, x_j, \xi_j; \theta) = x_j\beta - \alpha p_j + \xi_j + \epsilon_{i,j} = \delta_j + \epsilon_{i,j},$$

$$\delta_j = x_j\beta - \alpha p_j + \xi_j,$$

The optimal choice is the one where utility is the highest. Since there are unobserved variables, all consumers do not make the same choice. We solve the model for *choice probabilities*, which are driven by the distribution of ε_{ij} and the distribution of ξ_j. If we let these be extreme value distributions, then the choice probabilities follow the familiar logit form.

In particular, suppose that $\xi_j = 0$ and ε_{ij} are iid Type-1- Extreme-Value (T1EV) random variates (or suppose that $\xi_j + \varepsilon_{ij}$ are iid T1EV) satisfying $f(\varepsilon_{ij}) = \exp(-\exp(-\varepsilon_{ij}))\exp(-\varepsilon_{ij})$. Then,

$$P[x_j = 1] = \frac{\exp(x_j\beta - \alpha p_j)}{\sum_k \exp(x_k\beta - \alpha p_k)}.$$

If ε_{ij} is iid (both across products j for a consumer i, and across consumers i for a product j), then the distribution of a consumer's preferences over products other than the product it bought does not depend on the product they bought. The iid assumption comes at some cost.

McFadden noted a few problems with this model, all due to the iid nature of the T1EV errors. First, note that we can write the ratio of probabilities for choosing any two products j, k as

$$\frac{P[x_j = 1]}{P[x_k = 1]} = \frac{\exp(x_j\beta - \alpha p_j)}{\exp(x_k\beta - \alpha p_k)}.$$

This ratio of probababilities does not depend on the characteristics or price of any other good l.

Suppose the price of the product you buy rises. Then, you may want to switch. But, the good you switch to doesn't depend on the good you switched from. That is, if for good j, you drew ε_{ij} such that $x_j\beta - \alpha p_j + \varepsilon_{ij}$ is not the maximum across all goods, then you could switch to any other good with a sufficiently large ε. In reality, we might think that ε_{ij} are correlated across goods j for a given person i. For example, if some cereals are sweet and some not so sweet, you might think that some consumers like the sweet ones and others like the not so sweet ones. In that case, the $\varepsilon's$ would be correlated such that you either draw a bunch of high values for sweet and low values for not sweet, or vice versa. This would imply that when a person switches from a sweet cereal, they switch to a different sweet cereal.

Suppose 2 products have the same choice probabilities. Then, they have the same own-price derivatives, equal to $-\gamma s_j$ where s_j is the choice probability. This means they'd have the same markups (as these are determined in oligopoly by demand responses).

These problems with the iid errors do not disappear if we include the unobserved characteristic ξ_j. Rather, these just act like fixed effects in the logit probability expression above.

BLP address these problems by allowing for valuations β that vary across consumers. Let x_j include both observed product characteristics and p_j, the price of good j. (That is, let the price be one of the K characteristics.) Let z_i and ν_i be a T^o-vector *observed* consumer characteristics and a T^u-vector of *unobserved* consumer characteristics. Then, let utility be given by

$$U_{ij} = x_j\theta_i + \xi_j + \varepsilon_{ij},$$

where

$$\theta_i = \overline{\theta} + \Theta^o z_i + \Theta^u \nu_i.$$

This expression for utility is identical to the simple form given above in that it depends on observed characteristics, price (one of the characteristics now), an unobserved characteristic and an error term. Here, θ_i is a $K+1-$vector of parameters (K characteristics and 1 price) that varies across individuals. For a consumer with $z_i = \nu_i = 0$, $\theta_i = \overline{\theta}$. But, for other consumers, the vector θ_i adds the T^o-vector of observed characteristics z_i multiplied by the $T^o \times (K+1)$-matrix of parameters Θ^o, and the T^u-vector of unobserved characteristics ν_i multiplied by the $T^u \times (K+1)$-matrix of parameters Θ^u. The unobserved consumer characteristics break the restriction that everyone sees every alternative to their good as identical.

All utilities are relative to that of an outside good (U_{i0}), so we think of these as "utility above the outside good". Now the assumption of iid ε_{ij} does not bite so terribly, because we have Θ^u to allow for correlation across equations. So, let ε_{ij} be iid T1EV (so that we get a multinomial logit at the end).

Now, we want to express this utility function in terms of the bit that is easy to deal with, and the hard bits. Substituting the valuations above into the utility function, we get

$$U_{ij} = \delta_j + x_j\Theta^o z_i + x_j\Theta^u \nu_i + \varepsilon_{ij},$$

where

$$\delta_j = x_j\overline{\theta} + \xi_j.$$

At the market level, we are going to have δ_j which are not troublesome because the unobserved characteristic does not vary across i and because $\overline{\theta}$ does not vary across i. However, we do have 2 potentially troublesome terms: the interaction between observed product characteristics x and observed consumer characteristics z; and the interaction between observed product characteristics x and unobserved consumer characteristics ν.

2.2 Estimation of BLP

Consider the case where we have product x market level data that gives the market share of every product in every market (it doesn't need to be a balanced panel though). Also, assume there exist some instruments w satisfying $E[\xi|w] = 0$. Since we only observed these data, we obviously do not observe any consumer characteristics z. All are loaded onto the unobserved consumer characteristics ν. We assume that we do know the distribution(s) of these consumer characteristics.

The parameters of the model are δ, θ_2. Note that δ includes ξ. Let $\theta = (\theta_1, \theta_2)$. Rename

$$\delta_j = x_j'\theta_1 + \xi_j.$$

1. If you know δ, θ_2 and the distribution of v, you can compute predicted market shares σ_j. Conditional on the unobserved consumer characteristics ν, and given a particular parameter vector δ, θ_2, market shares have the logit form $\exp(\tau_j)/1 + \sum_s \exp(\tau_s)$. So, we integrate over the distribution of ν (there are no observed z's) to get market shares σ_j:

$$\sigma_j(\delta, \theta) = \int \frac{\exp\left(\delta_j + \sum_k x_{jk}\theta_{2k}\nu_k\right)}{1 + \sum_s \exp\left(\delta_s + \sum_k x_{jk}\theta_{2k}\nu_k\right)} f(\nu)\partial\nu.$$

   Nobody likes integrating. Pakes (1986) suggests sampling some ν from a distribution, and summing:

$$\sigma_j(\delta, \theta; P^{ns}) = \sum_{r=1}^{ns} \frac{\exp\left(\delta_j + \sum_k x_{jk}\theta_{2k}\nu_{rk}\right)}{1 + \sum_s \exp\left(\delta_s + \sum_k x_{jk}\theta_{2k}\nu_{rk}\right)}.$$

   Here, P^{ns} is a distribution from which you sample s times for each of the n observations in a particular market. This gives a vector of σ_j given $\delta, \theta; P^{ns}$. This step requires a distribution P^{ns} to sample from. You might choose a normal, e.g.. Since δ_j has an intercept, your normal could be mean-zero, thus having only a variance parameter.

2. Berry (1994) shows that there's only one δ that goes with any set of market shares and Θ, and that it is a contraction mapping. Alternatively put, market shares s_j identify δ_j conditional on Θ. So, we can solve for δ. BLP (1995) show that can be done by iterating the following linear equation

$$\delta_j^{new}(\theta) = \delta_j^{old}(\theta) + \ln s_j - \ln \sigma_j(\theta, \delta^{old}; P^{ns})$$

   until $\sigma_j(\delta, \theta; P^{ns}) = s_j$ for all j. This means we can identify δ given θ, including identifi-

cation of ξ given θ. In particular, with θ in our pocket and δ_j identified, we can compute $\xi_j = \delta_j - \sum_k x_{jk}\theta_{1,k}$. This iteration step would typically use a *while* loop in Stata.

3. Now we use our instruments for ξ to identify θ_2. Find θ_2 that bring ξ as close as possible to orthogonal to instruments w. In particular, one may use GMM for the moment conditions $E[\xi|w] = 0$. This is not the friendliest GMM, though, because for each value of θ, we have to do step 2 above. Luckily, it is canned in the Stata module *blp*.

4. The GMM requires instruments w that are uncorrelated with unobserved characteristics ξ. One might well suspect that observed product characteristics x (and consumer characteristics z, if used) are thus elements of w. But are prices p uncorrelated with unobserved characteristics? This seems a far cry. One might use supply-side instruments for prices. Or, one might model the supply side of the market, e.g., by modeling the pricing equations for oligopolistic supply (this would still require instruments).

2.2.1 Supply Side

We consider that there are F firms, each producing some subset of goods $\mathcal{F}_f$, of the J products. The marginal cost of the firm per good j is assumed to be

$$ln(mc_j) = w_j\theta_3 + \omega_j,$$

where γ is a vector of parameters, w_j, ω_j are observed and unobserved cost characteristics. We can allow x_j to be part of w_j and ω_j to be correlated with ξ_j.

The firms profit maximize with full knowledge of the behavior of consumers:

$$\Pi_f = \sum_{j\in\mathcal{F}_j} (p_j - mc_j)Ms_j(p, x, \xi; \theta),$$

where mc_j is given above. Firms can choose prices to maximize its profit given the attributes of products and the prices and attributes of competing products. M denotes the size of the market.

The first order conditions of the profit problem (assumed that it is well behaved) are:

$$s_j(p, x, \xi, \theta) + \sum_{r\in\mathcal{F}_f} (p_r - mc_r)\frac{\partial s_r(p, x, \xi; \theta)}{\partial p_j} = 0.$$

$$\Delta_{jr} = \begin{cases} -\frac{\partial s_r}{\partial p_j} & \exists f : r, j \in \mathcal{F}_f \\ 0 & \text{otherwise} \end{cases}.$$

Price-cost markups:

$$s(p, x, \xi; \theta) - \Delta(p, x, \xi; \theta)[p - mc] = 0$$

$$p = mc + \Delta(p, x, \xi, \theta)^{-1} s(p, x, \xi; \theta).$$

$$ln(p - b(p, x, \xi, \theta)) = w\theta_3 + \omega.$$

We impose $E[\omega|z] = 0$ as well.

2.3 References

1. Berry, Steve, 1994, "Estimating Discrete Choice Models of Product Differentiation," RAND, vol. 25, no. 2, pp. 242-262.

2. Berry, S., J. Levinsohn and A. Pakes, 1995, "Automobile Prices in Market Equilibrium," Econometrica, vol. 63, no. 4, pp. 841-890.

3. Blow, Laura, Browning, Martin and Crawford, Ian. 2008. "Revealed Preference Methods for the Consumer Characteristics Model" Review of Economic Studies, vol 75, pp. 371-389.

4. Pakes, Ariel, 1986, "Patents as Options: Some Estimates of the Value of Holding European Patent Stocks," Econometrica, vol. 54, pp. 755-784.

5. Daniel Ackerberg, C. Lanier Benkard, Steven Berry, and Ariel Pakes, 2011, "Econometric Tools for Analyzing Market Outcomes", Handbook Chapter.

6. McFadden, Daniel, 1974, "The measurement of urban travel demand." Journal of public economics 3.4: 303-328.

7. McFadden, Daniel, and Paul Zarembka, 1974, "Conditional logit analysis of qualitative choice behavior" in Frontiers in Econometrics, 105-142.

Part II

Nonparametric Methods

Chapter 3

Entropic Latent Variable Integration via Simulation (ELVIS)

We consider again a consumer setup. $X = \mathbb{R}_+^K$, $K = 2$, and $T = 2$, with time window $\mathcal{T} = \{1, 2\}$. Let $\mathbf{x} = (\mathbf{p_t}, \mathbf{c_t})_{t\in\mathcal{T}}$ be an observed random vector of consumptions and prices and $\mathbf{e}$ be an unobserved random vector $\mathbf{e} = (\alpha, (\lambda_\mathbf{t})_{\mathbf{t}\in\mathcal{T}}, (\mathbf{w}_t)_{t\in\mathcal{T}})$. Consider a g vector of moments. The consumers are assumed to be Cobb-Douglas $u(c, \alpha) = c_1^\alpha + c_2^{(1-\alpha)}$. The first -order-conditions (FOC) of this problem given a lagrange multiplier $\lambda = (\lambda_t)_{t\in\mathcal{T}}$ supported on $\mathbb{R}_{++}$ are

From the Lagrangian

$$\mathcal{L} = u(c_1, c_2, \alpha) + \lambda(p_1c_1 + p_2c_2 - y)$$

$$\nabla u(c, \alpha) = \begin{bmatrix} \partial_1 u(c_1, c_2, \alpha) \\ \partial_2 u(c_1, c_2, \alpha) \end{bmatrix}$$

$$\nabla u(c, \alpha) = \lambda p = \lambda \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$$

$$g_{A,t,1}(x, e) = 1(c_{t,1} - w_{t,1} = (\frac{\lambda_t p_{t,1}}{\alpha})^{\frac{1}{\alpha-1}}) - 1 \forall t$$

$$g_{A,t,2}(x,e) = 1(c_{t,2} - w_{t,2} = (\frac{\lambda_t p_{t,2}}{1-\alpha})^{-\frac{1}{\alpha}}) - 1 \forall t$$

$$g_{M,k}(x,e) = w_{t,k} \forall k$$

This means that there is a joint distribution $\mu \times \pi_0$, for some measure $\mu \in \mathcal{P}_{E|X}$ and $\pi_0 \in \mathcal{P}_X$ the observed measure, over the support of $X \times E$ such that

$$\mu \times \pi_0(\omega \in \Omega : c_{t,1}(\omega) - w_{t,1}(\omega) = (\frac{\lambda_t(\omega) p_{t,1}(\omega)}{\alpha(\omega)})^{\frac{1}{\alpha(\omega)-1}}) = 1 \forall t$$

$$\mu \times \pi_0(\omega \in \Omega : c_{t,1}(\omega) - w_{t,1}(\omega) = (\frac{\lambda_t(\omega) p_{t,2}(\omega)}{1-\alpha(\omega)})^{-\frac{1}{\alpha(\omega)}}) = 1 \forall t$$

Then the measurement error moment is known and centered around zero:

$$E_{\mu \times \pi_0}(w_{t,k}) = 0 \forall t, k.$$

Collect all these previous moments into the vector of moments $g(x,e)$.

Theorem 1. ***(Schennach, 2014 ECMA and Aguiar and Kashaev 2022 Restud)*** *The following are equivalent.*

1. A random vector $\mathbf{x} = (\mathbf{p_t}, \mathbf{c_t})_{t \in \mathcal{T}}$ *almost surely satisfies the FOC of CES utility maximization problem under centered measurement error under centered measurement error.*

2.

$$\inf_{\mu \in \mathcal{P}_{E|X}} ||E_{\mu \times \pi_0}[g(x,e)]|| = 0.$$

where $\pi_0 \in \mathcal{P}_X$ *is the observed distribution of* $\mathbf{x}$*.*

We now define a way to check (2) in a way that is computationally feasible.

Definition 1. (Maximum-entropy moment)

$$h(x;\gamma) = \frac{\int_{e \in E|X} g(x,e) \exp(\gamma' g(x,e)) d\eta(e|x)}{\int_{e \in E|X} \exp(\gamma' g(x,e) d\eta(e|x)},$$

where $\gamma \in \mathbb{R}^{k+q}$ is a nuisance parameter, and $\eta \in \mathcal{P}_{E|X}$ is an arbitrary user-input distribution supported on $E|X$ such that $E_{\pi_0}[logE_\eta[\exp(\gamma' g(x,e))|x]]$ exists and twice continuously differentiable in γ for all $\gamma \in \mathbb{R}^{k+q}$.

Note that

$$\{d\eta^*(\cdot|x;\gamma) = \frac{\exp(\gamma' g(x,\cdot))d\eta(\cdot|x)}{\int_{e\in E|X}\exp(\gamma' g(x,e))d\eta(e|x)}, \gamma \in \mathbb{R}^{k+q}\}$$

is a family of exponential conditional probability measures. Thus the maximum entropy moment h is the marginal moment of the function g, at which the latent variable has been integrated out using one of the members from the above exponential family. We call this the Elvis distribution.

The importance of this distribution is:

Theorem 2. ***Theorem.*** *The following are equivalent:*

1. A random vector $\mathbf{x} = (\mathbf{p_t}, \mathbf{c_t})_{t\in\mathcal{T}}$ *almost surely satisfies the FOC of CES utility maximization problem under centered measurement error.*

2.

$$\inf_{\gamma\in\mathbb{R}^{k+q}} ||E_{\pi_0}[h(x;\gamma)]|| = 0$$

where $\pi_0 \in \mathcal{P}_X$ *is the observed distribution over* x*.*

The idea or intuition behind this theorem is that there exists a distribution that satisfies the moment conditions, then there must be a distribution in the family of Elvis distributions, and satisfies the same moment conditions. This is a finite problem after we have computed the integral with respect to the Elvis distribution for a fixed γ parameter.

In Aguiar and Kashaev (AK 2022) we use the following η that satisfies all the requirements for the theorem to work:

$$\eta(e|x) \propto \exp(-||g_M(x,e)||^2)[1(c_{t,1} - w_{t,1} = (\frac{\lambda_t p_{t,1}}{\alpha})^{\frac{1}{\alpha-1}}) - 1\forall t][1(c_{t,2} - w_{t,2} = (\frac{\lambda_t p_{t,2}}{1-\alpha})^{-\frac{1}{\alpha}}) - 1\forall t].$$

Note that we have included the moments of the model g_A as a support constraint on η, and the only the moment g_M is used in the density.

3.1 Simulated GMM and Testing

The data is $\{x_i\}_{i=1}^n = \{(p_{t,i}, c_{t,i})_{t\in\mathcal{T}}\}_{i=1}^n$ where n is the sample size. The sample analogue of h is

$$\hat{h}_M(\gamma) = \frac{1}{n}\sum_{i=1}^{n} h_M(x_i, \gamma)$$

$$\hat{\Omega}(\gamma) = \frac{1}{n}\sum_{i=1}^{n} h_M(x_i, \gamma) h_M(x_i, \gamma)' - \hat{h}_M(\gamma)\hat{h}_M(\gamma)'.$$

We let Ω^- be the generalized inverse of matrix Ω.

$$TS_n = n \inf_{\gamma \in \mathbb{R}^q} \hat{h}_M(\gamma)\hat{\Omega}^-(\gamma)\hat{h}_M(\gamma).$$

We assume that $\{x_i\}_{i=1}^n$ is i.i.d.

Theorem 3. *Suppose $\{x_i\}_{i=1}^n$ is i.i.d. and the previous assumptions hold then under the null that the data is rationalizable it follows that*

$$lim_{n\to\infty} Pr(TS_n > \chi^2_{q,1-\alpha}) \leq \alpha$$

for every $\alpha \in (0,1)$.

If moreover the minimal eigenvalue of the variance matrix $V[h_M(x,\gamma)]$ is uniformly, in γ, bounded away from zero and its maximal eigenvalue is uniformly, in γ, bounded from above, then under the alternative hypothesis that the data is not approximately consistent with rationalizability, it follows that

$$lim_{n\to\infty} P(TS_n > \chi_{q,1-\alpha}) = 1.$$

Now we can do some recoverability of parameters of interests by test-inversion.

The $(1-\alpha)$-confidence set for θ_0 is

$$\{\theta_0 \in \Theta : TS_n(\theta_0) \leq \chi^2_{q_{ext},1-\alpha}\}.$$

Note that $q_{ext} = q + d$ where d is the number of new parameters introduced in the problem.

3.2 Alternatives to ELVIS: Support approaches (Li and Potoms and DeMuynck: Testing revealed preference models with unobserved randomness: a column generation approach).

Observables in y with support $\mathcal{Y} \subseteq \mathbb{R}^{d_Y}$. Unobservables are collected in u with support $\mathcal{U} \subseteq \mathbb{R}^{d_U}$. The unknown joint distribution over observables and unobservables (y, u) is μ on some measure space $(\mathcal{Y} \times \mathcal{U}, \mathcal{B})$ where $\mathcal{B}$ is the Borel σ-algebra on $\mathcal{Y} \times \mathcal{U} \subseteq \mathbb{R}^{d_U + d_Y}$.

Definition 2. A model consists of a tuple $(\mu_Y, \Gamma, f, \alpha)$, where

- μ_Y is the marginal probability measure with respect to observables,
- $\Gamma \subseteq \mathcal{Y} \times \mathcal{U}$ is a $\mathcal{B}$-measurable set that gives all combinations $(y, u) \in \mathcal{Y} \times \mathcal{U}$ that are consistent with the economic model.
- $f : \mathcal{Y} \times \mathcal{U} \to \mathbb{R}^K$ gives a vector of measurable functions $f = (f^1, \cdots, f^K)$ that govern the moment conditions imposed by the economic model
- $\alpha = (\alpha^1, \cdots, \alpha^K) \in \mathbb{R}^K$ is a K-dimensional vector of moment values for these functions.

The marginal distribution μ_Y corresponds with the joint

$$\mu_Y(A) = \mu(A \times \mathcal{U}). \tag{3.1}$$

We know that Γ encompasses all possible values of (y, u) that can arise in the model

$$\mu(\Gamma) = 1. \tag{3.2}$$

Also the moment conditions are taken with respect to μ and the support Γ

$$E_\mu f^k = \int_\Gamma f^k(y, u) d\mu = \alpha^k.$$

$$E_\mu f = \alpha. \tag{3.3}$$

The problem is to find a μ is any such that 1-3 hold.

Now define $\mathcal{H}(\mu_Y, \Gamma)$ as the collection of all feasible distributions μ such that 1-2. Define the correspondence $F : \mathcal{Y} \rightrightarrows \mathbb{R}^K$:

$$F(y) = \{f(y,u) : (y,u) \in \Gamma\}.$$

Note that since y is random $F(y)$ is a random set with support on subsets of $\mathbb{R}^K$.

Assumption. *(PD-2) (i) The sets $F(y)$ are closed $\mu_Y - a.s.$ (ii) There is a measurable function $g(y)$ that only depends on y such that $E_\mu g(y) < \infty$ and: $g(y) \geq \sup_{(y,u)\in\Gamma} ||f(y,u)||$ $\mu_Y - a.s.$.*

The problem can be rewritten as

$$\min_{\mu \in \mathcal{H}(\mu_Y, \Gamma)} E_\mu ||f(y,u) - \alpha|| = 0,$$

where $||\cdot||$ is the Euclidean norm in $\mathbb{R}^K$.

We define a support function. For a compact set $A \subseteq \mathbb{R}^K$, the support function of A, $h_A : \mathbb{R}^K \to \mathbb{R}$ is given by:

$$h_A(\lambda) = \sup_{x \in A} < \lambda, x >,$$

where $< \cdot, \cdot >$ denotes the inner product. Let $co(A)$ be convex hull of A and $\overline{co}(A)$ the convex, closed closure of A.

$$\overline{co}(A) = \{x \in \mathbb{R}^K : \forall \lambda \in \mathbb{S}^K, < \lambda, x > \leq h_A(\lambda)\},$$

where $\mathbb{S}^K = \{\lambda \in \mathbb{R}^K : ||\lambda|| = 1\}$is the $K-1$ dimensional unit simplex.

If (3) is satisfied then by linearity of the expectation:

$$E_\mu < \lambda, f(y,u) > = < \lambda, \alpha > \forall \lambda \in \mathbb{S}^K.$$

Now by (2)

$$\mu(\{(y,u) \in \Gamma : < \lambda, f(y,u) > \leq h_{F(y)}(\lambda)\}) = 1,$$

combining the conditions above we obtain

$$E_\mu[h_{F(y)}(\lambda)] \geq E_\mu < \lambda, f(y,u) > = < \lambda, \alpha > \forall \lambda \in \mathbb{S}^K.$$

Notice that

$$E_{\mu}[h_{F(y)}(\lambda)] = E_{\mu_Y}[h_{F(y)}(\lambda)],$$

then

$$E_{\mu_Y}[h_{F(y)}(\lambda)] \geq < \lambda, \alpha > \forall \lambda \in \mathbb{S}^K.$$

Theorem. *(Li-3) If Assumption PT-2 holds, then* $\min_{\mu \in \mathcal{H}(\mu_Y, \Gamma)} E_{\mu}||f(y,u) - \alpha|| = 0$, *holds iff*

$$\inf_{\lambda \in \mathbb{S}^K} E_{\mu_Y}[h_{F(y)}(\lambda) - < \lambda, \alpha >] \geq 0.$$

This is an alternative to ELVIS and characterizes the sharp identified regions for partially identified econometric models.

Chapter 4

Finite Mixture Models, Random Utility Model, Bootstrapping

Let X be an abstract grand choice set. Let $\mathcal{B}$ a collection of measurable Borel subsets of X. Let $j \in J$ index one of all choice $|J|$ sets $C_j \in 2^X \setminus \emptyset$. Let P_j be a probability measure over $\mathcal{B}$. Many models of choice in economics can be written as

$$P_j(A) = \int f(A|\alpha) d\mu(\alpha) \quad \forall j \in J.$$

Where $f(A|\alpha)$ is a prediction of choice conditional on a behavioral model α and μ is a measure of all behavioral models.

Random Utility Model.

$$\alpha = u, u : X \to \mathbb{R}$$

$$f(A|u) = 1(argmax_{y \in C_j} u(y) \in A)$$

$$P_j(A) = \int 1(argmax_{y \in C_j} u(y) \in A) d\mu(u).$$

This model is complex because the space of utilities is infinite dimensional. Also we are considering all elements in $\mathcal{B}$ which is a large collections of sets. We have to simplify the problem.

4.0.1 Finite X and no indifference.

If X is finite then we can let $\mathcal{B}$ to be all singletons. For instance, $X = \{a, b, c\}$ then $\mathcal{B} = \{\{a\}, \{b\}, \{c\}\}$. Also, notice that the collection of utilities will map to $|X|!$ linear rankings if we rule out indifference. In terms of utilities we require $\mu(u : u \quad \text{is injective}) = 1$. For the running example we have:

$$\begin{array}{l} \succ^1 : a \succ^1 b \succ^1 c \\ \succ^2 : a \succ^2 c \succ^2 b \\ \succ^3 : b \succ^3 a \succ^3 c \\ \succ^4 : b \succ^4 c \succ^4 a \\ \succ^5 : c \succ^5 a \succ^5 b \\ \succ^6 : c \succ^6 b \succ^6 a \end{array} .$$

This means that we have mapped the infinite collection of utilities to 6 rankings over $X = \{a, b, c\}$. There is no loss of generality here!

Now we can write the mixture problem as:

$$\rho_j(a) = P_j(\{a\}) = \sum_{\succ \in \mathcal{R}} \mu^*(\succ) 1(a \succ b \forall b \in C_j \setminus \{a\}),$$

where $\mathcal{R} \subseteq X \times X$ is the set of linear orders/strict preferences on X, $\mu^* \in \Delta(\mathcal{R})$, $\mu^*(\succ) \geq 0$ and $\sum_{\succ \in \mathcal{R}} \mu^*(\succ) = 1$.

After discretization, without loss of generality, we can write $\rho_j \in \Delta(C_j)$ and $\rho = (\rho_j)_{j \in J}$. Then we can write down

$$\rho = A\mu^*,$$

where

$$A = \begin{array}{c} \\ a,\{a,b\} \\ b,\{a,b\} \\ b,\{b,c\} \\ c,\{b,c\} \\ a,\{a,c\} \\ c,\{a,c\} \\ a,\{a,b,c\} \\ b,\{a,b,c\} \\ c,\{a,b,c\} \end{array} \begin{bmatrix} \succ^1 & \succ^2 & \succ^3 & \succ^4 & \succ^5 & \succ^6 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

4.0.2 (Kitamura Stoye) $X = \mathbb{R}^L_+$ and linear budgets.

We let $X = \mathbb{R}^L_+$ and $C_j = B_j = \{x \in X : p_j' x \leq w_j\}$ for simplicity we can set $w_j = 1$ for all $j \in J$. We also let U be the set of all $u : X \to \mathbb{R}$ that are strictly (quasi)concave, continuous and monotone. We let O be an element of the set of all Borel measurable sets on X. That means that μ a measure on U produces choice

$$P_j(O) = \int 1(argmax_{y \in B_j} u(y) \in O) d\mu(u).$$

We will discretize the problem. First notice that because of monotonicity choices will be on the budget lines with probability 1.

Patches are the **coarsest partition** of intersecting budgets in a time period (I_j collects those patches):

$$\rho(x_{i|j}) = P_j(x_{i|j}).$$

We then let

$$\rho = (\rho(x_{i|j})_{i \in I_j, j \in J}).$$

We then need to obtain the rational demand types:

WLG, we focus on representative elements of patches (Kitamura and Stoye, 2018; henceforth KS). ($x^*_{i|j} \in x_{i|j}$). Discretization with-

		B_1	B_2	rational
Type 1	$\theta(1,1)$	$x_{1\|1}$	$x_{1\|2}$	yes
Type 2	$\theta(1,2)$	$x_{1\|1}$	$x_{2\|2}$	yes
Type 3	$\theta(2,2)$	$x_{2\|1}$	$x_{2\|2}$	yes
Type 4	$\theta(2,1)$	$x_{2\|1}$	$x_{1\|2}$	no

Table 4.1: Rational Types.

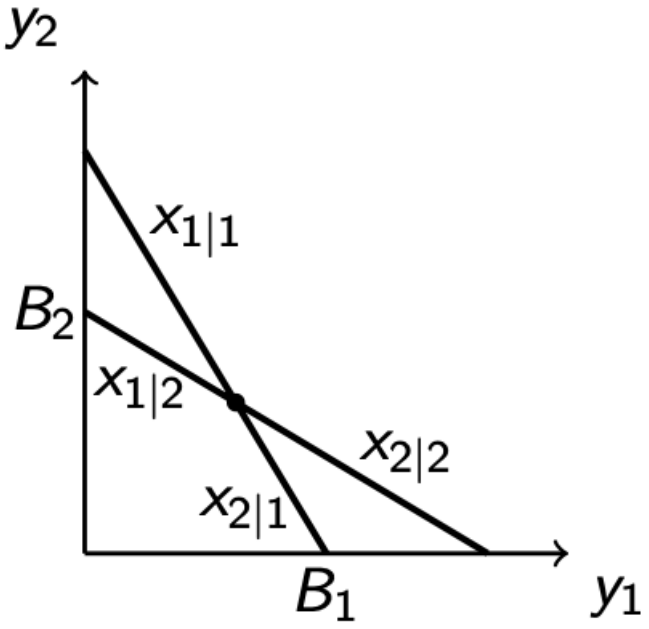

Figure 4.1: Patches

out loss! This has to be proven but we won't
do it here.

We then define

$$\rho = A\nu$$

for $\nu \in \Delta(\Theta)$ where Θ is the set of demand types.

$$A = \begin{array}{c} \\ x_{1|1} \\ x_{2|1} \\ x_{1|2} \\ x_{2|2} \end{array} \begin{bmatrix} \theta(1,1) & \theta(1,2) & \theta(2,2) \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

4.0.3 Statistical Assumptions

$$\mu_j(\alpha) = \mu(\alpha)$$

for all $j \in J$. This is an exclusion-restriction that says that the distribution of α is independent of the choice situation. In the case of RUM this assumption was imposed by McFadden-Richter and essentially says that the distribution of utilities is independent from budgets:

$$\mu_j(u) = \mu(u).$$

In experimental datasets this assumptions can hold by design. The reason is that budgets can be exogenously varied and they will be independent of preferences. In reality this assumption may be broken in survey data where different individuals may have preferences (e.g., risk aversion parameter) that is correlated with income. We will have to deal with this using techniques that can "fix" this form of endogeneity. After discretization we can write down a discrete type space $\mathcal{R}$ that is finite and a discrete number of outcomes $x_{k|j}$ for $k = 1, \cdots, K_j$, such that $a_{k|j} \in C_j$.

$$\rho_j(x_{k|j}) = \rho(x_{k|j}).$$

Then let $\rho = (\rho(x_{k|j})_{k \in K_j, j \in J})$ and we define

$$\rho(x_{k|j}) = \sum_{r \in \mathcal{R}} \nu(r) f(x_{k|j}|\phi(r)).$$

The key is that for each $\alpha \in \mathcal{A}$ there exists an $r \in \mathcal{R}$ such that $f(x_{k|j}|\alpha) = f(x_{k|j}|\phi(r))$. Many α's map to a single r and there is no α that cannot be mapped to some $r \in \mathcal{R}$. We can then always build a matrix as in the previous examples $A_{k|j;r} = f(x_{k|j}|\phi(r))$.

Theorem 4. *The following are equivalent:*

(i) P is consistent with a mixture $P_j(A) = \int f(A|\alpha) d\mu(\alpha)$.

(ii) P with vector representation ρ is such that $\rho = A\nu$ for some $\nu \in \Delta(\mathcal{R})$.

4.0.4 Testing

In reality we do not observe P_j, in fact we can only estimate ρ the discrete counterpart in most situations. As we observed in the previous part, we can focus in many instances on the discretized problem without loss of generality. We collect choices from budgets from a population then

$$\hat{\rho}_j(a) = \hat{\rho}_{j,n}(a) = \frac{1}{N_j} \sum_{i=1}^{N_j} 1(a = c_i(C_j)),$$

where $c_i(C_j)$ is the observed choice of individual i from C_j.

In general we are assuming $c_i(C_j)$ are i.i.d. Under that we can apply the law of large numbers to conclude that:

$$\lim_{n \to \infty} \hat{\rho}_{j,n}(a) = \rho_j(a).$$

Notice that even if ρ is consistent with a finite mixture its finite sample analogue $\hat{\rho}$ may be such that there is no $\nu \in \Delta(\mathcal{R})$ such that

$$\hat{\rho} = A\nu.$$

The reason is that $\rho - \hat{\rho} = \epsilon$ that is not zero in general.

The null hypothesis is stated formally as:

$$H_0 : \exists \nu \geq 0, A\nu = \rho.$$

Notice that without loss of generality we have not searched over $\nu \in \Delta(\mathcal{R})$ but only over the convex cone $\mathcal{C} = \{z : A\nu = z, \nu \geq 0\}$. Alternatively, we can write down the null as:

$$H_0 : \rho \in \mathcal{C}.$$

In particular, this is equivalent to testing:

$$H_0 : \min_{\eta \in \mathcal{C}} [\rho - \eta]' \Omega [\rho - \eta] = 0,$$

for Ω a deterministic positive definite matrix.

The sample counterpart of H_0 is

$$\min_{\eta \in \mathcal{C}} [\hat{\rho} - \eta]' \Omega [\hat{\rho} - \eta].$$

The test statistic $\mathcal{J}_N$

$$\mathcal{J}_N = N \min_{\eta \in \mathcal{C}} [\hat{\rho} - \eta]' \Omega [\hat{\rho} - \eta]$$

$$= N \min_{\nu \in \mathbb{R}_+^{|\mathcal{R}|}} [\hat{\rho} - A\nu]' \Omega [\hat{\rho} - A\nu].$$

If $J_N = 0$ then the null is accepted but what happens when it is above 0. We have to do statistical hypothesis testing using a simulated critical value.

4.0.4.1 Simulating a Critical Value

We need to obtain $\hat{\rho}^{*(b)}$ for $b = 1, \cdots, B$, bootstrap sample.

We need a tuning parameter $\tau_N = \sqrt{\frac{log(min_j N_j)}{(min_j N_j)}}$, in reality it can be picked such that $\tau_N \to 0$ as $\sqrt{N}\tau_N \to \infty$. Restrict Ω to be diagonal and positive definite and $1_{|\mathcal{R}|}$ is a vector of ones of size $|\mathcal{R}|$.

(i) Obtain τ_N-tightened restricted estimator $\hat{\eta}_{\tau_n}$ which solves:

$$\min_{\eta \in \mathcal{C}_{\tau_\mathcal{N}}} N[\hat{\rho} - \eta]'\Omega[\hat{\rho} - \eta] = \min_{[\nu - \tau_N 1_{|\mathcal{R}|}/|\mathcal{R}|] \in \mathbb{R}_+^{|\mathcal{R}|}} N[\hat{\rho} - A\nu]'\Omega[\hat{\rho} - A\nu].$$

(ii) Define the τ_N-tightened recentered boostrap estimator

$$\hat{\rho}^{*(b)}_{\tau_N} := \hat{\rho}^{*(b)} - \hat{\rho} + \hat{\eta}_{\tau_n}, \quad \forall b \in 1, \cdots, B.$$

(iii) The bootstrap test statistic is

$$\mathcal{J}_N^{*(b)} = min_{[\nu - \tau_N 1_{|\mathcal{R}|}/|\mathcal{R}|] \in \mathbb{R}_+^{|\mathcal{R}|}} N[\hat{\rho}^{*(b)}_{\tau_N} - A\nu]'\Omega[\hat{\rho}^{*(b)}_{\tau_N} - A\nu],$$

for all $b = 1, \cdots, B$.

(iv) Use the empirical distribution of $\mathcal{J}_N^{*(b)}$, for $b = 1, \cdots, B$, to obtain the critical value for $\mathcal{J}_N$.

4.0.5 Dealing with Endogeneity in the Demand Setup (Paper Reference: Revealed Price Preference: Theory and Empirical Analysis).

This paper introduces a new model of consumption.

Definition 3. Augmented Utility Functions: Consider $\mathcal{D} = \{p^t, x^t\}_{t=1}^T$, from a consumer, each observation consists of the prices $p^t \in \mathbb{R}^L_{++}$ of the L goods, and the consumer demands $x^t \in \mathbb{R}^L_+$ at those prices. We have an augmented utility $U : X \times \mathbb{R}_- \to \mathbb{R}$, that $(X = \mathbb{R}^L_+)$ rationalizes $\mathcal{D}$ in the following sense:

$$x^t \in argmax_{x \in X} U(x, -p^{t\prime}x)$$

for all $t = 1, \cdots, T$. We assume that this U is monotone on the arguments.

Note that we can define the value function of the problem above that is indirect utility over prices: $V : \mathbb{R}^L_{++} \to \mathbb{R}$,

$$V(p^t) = max_{x \in X} U(x, -p^{t\prime}x).$$

This is an expenditure-augmented utility function, where $U(x, -e)$ is the consumer's utility when she acquires x at the cost e. This means that expenditure is endogenous and dependent on prices.

Special case is the quasilinear model:

$$U(x, -p^{t\prime}x) = u(x) - p'x.$$

We can have also nonseparable models:

$$U(x, -p^{t\prime}x) = u(x)/p'x$$

Consider the new price preference revelation:

We say that p^s is directly (strictly) revealed preferred to p^t ($p^s \succeq_p (\succ_p) p^t$) if $p^{s\prime}x^t \leq (<) p^{t\prime}x^t = e^t$ (note that $p^s \succeq_p (\succ_p) p^t) \implies V(p^s) \geq (>) V(p^t)$ when $\mathcal{D}$ are rationalized by a Augmented Utility). Then we define indirect price preference $p^s \succeq_p^* p^t$ if there is a finite sequence $k, r, l, \cdots, n$, $p^s \succeq_p p^k \succeq_p p^r \succeq_p p^l \succeq_p \cdots p^n \succeq_p p^t$.

Definition 4. Generalized Axiom of Price Revealed Preference (GAPP). We say that $\mathcal{D} = \{p^t, x^t\}_{t=1}^T$ satisfies GAPP if there is no s, t such that $p^s \succeq_p^* p^t$ and $p^t \succ_p p^s$.

Theorem 5. *Given a data set $\mathcal{D} = \{(p^t, x^t)\}_{t=1}^T$, the following are equivalent:*

1. $\mathcal{D}$ is rationalized by an augmented utility function.
2. $\mathcal{D}$ satisfies GAPP.
3. $\mathcal{D}$ is rationalized by an augmented utility function that is strictly increasing, continuous, and concave. Moreover, U is such that there is always a maximum for all $p \in \mathbb{R}_{++}^L$.

We that the Generalized Axiom of Revealed Preference (GARP). We say that x^t is directly (strictly) revealed preferred to x^s ($x^t \succeq_x x^s$) whenever $p^{t\prime}x^t \geq (>) p^{t\prime}x^s$. We define the indirect commodity revealed preference completely analogous to the case of the price preference such that we have $x^t \succeq_x^* x^s$ when there is a chain of direct revelation.

Definition 5. GARP. We say that $\mathcal{D} = \{p^t, x^t\}_{t=1}^T$ satisfies GARP if there is no s, t such that $x^t \succeq_p^* x^s$ and $p^s \succ_p p^t$.

Proposition 1. *Let* $\mathcal{D} = \{p^t, x^t\}_{t=1}^T$ *be a data set and let* $\mathcal{D}^* = \{p^t, x^{t*}\}_{t=1}^T$ *such that* $x^{t*} = \frac{x^t}{p^{t\prime}x^t}$. *Then*

$\mathcal{D}$ *satisfies GAPP if and only if* $\mathcal{D}^*$ *satisfies GARP.*

This is powerful because we can apply the mixture techniques here using the normalized patches, we can essentially do RUM for demand with endogenous expenditure.

Part III

Machine Learning

Chapter 5

Deep Feedforward Networks

Deep forward networks, or feedforward neural networks (DFNN), are the **quintessential** model of **deep learning**.

- Objective: Approximate a function f^*.

For discrete choice, $y = f^*(x)$ where an input x (features/attributes) is mapped into a category or choice y.

- A DFNN is a model, defined by a mapping:

$$y = f(x, \theta),$$

where θ is a parameter. The objective of DFNN is to learn the value of θ that results in the best approximation of the observed instances of f^* (dataset).

- Defining features:

    1. DFNN are called **feedforward** because information flows through the function being evaluated from x, through the intermediate computations used to define f, and finally the output y. No feedback is allowed. Counterexample: $y_t = f(y_{t-1}, x_t, \theta)$ for $t \in \{1, 2\}$.
    2. DFNN are called **neural** because they are loosely inspired by biological neurons. Modernly, they do not try to model a biological brain.
    3. DFNN are called **networks** because they are typically represented by composing many different functions. The model is associated with a **directed acyclical graph** describing how functions are composed together:

$$f(x) = (f^d \circ f^{d-1} \cdots f^3 \circ f^2 \circ f^1)(x),$$

where $\circ$ denotes function composition, f^1 is called the **first layer** of the network, f^2 is called the **second layer** and so on. The overall length of the chain, d, denotes the **depth** of the model. (Deep learning!). The final layer f^d is called the **output layer.**

- Neural network training tries to match $f(x)$ to $f^*(x)$. The training data provides us with **noisy** instances of $f^*(x)$. Namely,

$$y \simeq f^*(x).$$

- There is no information about the behavior of each **hidden layer.** The dimensionality of each these hidden layers determines the **width** of the model.
- Each entry of the vector $x = (x_k)_{k=1}^K \in X \subseteq \mathbb{R}^K$ can be thought as a **neuron.**
- Each layer is a vector valued function $f^1 : X \to X^2$, and $f^n : X^{n-1} \to X^n = Y$, where $y \in Y$. Then the entry $f_l^1(x)$ can be thought as a neuron, that is **activated** by receiving information. Each entry represents a neuron processing one aspect of this information.

Example 2. Linear Probability Model. $y \in \{0, 1\}$.

$$y = f^1(x) = x'w + b,$$

where $\theta = (w, b)$, where w are called **weights,** and b are called **biases.**

Example 3. Logit Probability Model. $y \in \{0, 1\}$

$$y = (f^2 \circ f^1)(x) = \frac{1}{1 + exp(x'w + b)},$$

where $\theta = (w, b)$, where w are called **weights,** and b are called **biases,** and $f^2(z) = \frac{1}{1+exp(z)}$ is the logit **activation function.**

Example 4. Mixed Logit Model.

$$y = f^3 \circ f^2 \circ f^1(x) = \sum_{w \in W} w^2(w) \frac{1}{1 + exp(x'w + b)}$$

Example 5. Closed form solution to the XOR function. Consider the XOR function that is defined as $f(1, 1) = f(0, 0) = 0$ and $f(1, 0) = f(0, 1) = 1$. The vector of features $x = (x_1, x_2)$ is

a vector of binary variables. We have the following four points $\mathbb{X} = \{[0,0]', [0,1]', [1,0]', [1,1]'\}$. We will train a DFNN on this points to learn f.

We define a loss function MSE:

$$J(\theta) = \frac{1}{4}\sum_{x\in\mathbb{X}}(f^*(x) - f(x,\theta))^2.$$

Now we must **choose the model** $f(x,\theta)$**.** Suppose that we choose first a linear model

$$f(x; w, b) = x'w + b.$$

We can minimize $J(\theta)$ to obtain the solution and it will be $w = 0, b = \frac{1}{2}$. This linear model is such that $f(x; 0, \frac{1}{2}) = \frac{1}{2}$ for all x.

Now let's augment the depth of this DFNN, so let's create a first layer with two hidden components this is captured by

$$h = f^1(x; W, c) = WX + c,$$

where $f^1 : X \to \mathbb{R}^2$, and h represents the hidden units. Then we have an output layer

$$y = f^2(h; w, b),$$

$f^2 : \mathbb{R}^2 \to \mathbb{R}_+$ we need h^2 to be nonlinear to learn XOR so we use an activation function. A general recommendation is the rectifier linear unit ReLU such that

$$g(z) = \max\{0, z\},$$

that is applied elementwise to a vector and denoted for simplicity with this notation. Then allowing also weights and biases will give us the full neural net:

$$f(x; W, c, w, b) = f^2 \circ f^1(x, \theta) = w' \max\{0, W'x + c\} + b.$$

The solution to this problem is:

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$c = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

$$w = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

and $b = 0$.

We can now get the points

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix},$$

$$XW = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}.$$

Next we add the bias vector:

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}.$$

We apply Relu:

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix},$$

this nonlinear transformation is key as it allows the outter linear model to fit the nonlinear XOR, finally multiplying by w

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Backpropagation and Chain Rule (based on Roger Grosse notes lecture 2)

Chain Rule

We assume we have one input and out training example (x, t).

We use a squared error loss function.

$$z = wx + b$$

$$y = \sigma(z)$$

$$\mathcal{L} = \frac{1}{2}(y - t)^2$$

Se will consider now a regularizer, that will penalize weights that are far from zero.

$$\mathcal{R} = \frac{1}{2}w^2 \tag{5.1}$$

$$\mathcal{L}_r = \mathcal{L} + \lambda\mathcal{R}. \tag{5.2}$$

We want to perform gradient descent, we want to compute the partial derivatives $\partial\mathcal{L}\backslash\partial w$ and $\partial\mathcal{L}\backslash\partial b$.

We will work now on the important idea of backpropagation or **backprop.**

Calculus class

$$\mathcal{L}_r = \frac{1}{2}\left(\sigma(wx + b) - t\right)^2 + \frac{\lambda}{2}w^2$$

$$\frac{\partial\mathcal{L}_r}{\partial w} = \frac{\partial}{\partial w}\left[\frac{1}{2}\left(\sigma(wx + b) - t\right)^2 + \frac{\lambda}{2}w^2\right]$$

$$= \frac{1}{2}\frac{\partial}{\partial w}\left(\sigma(wx+b)-t\right)^2 + \frac{\lambda}{2}\frac{\partial}{\partial w}w^2$$

$$= \left(\sigma(wx+b)-t\right)\sigma'(wx+b)\frac{\partial}{\partial w}(wx+b) + \lambda w$$

$$= \left(\sigma(wx+b)-t\right)\sigma'(wx+b)x + \lambda w$$

Similar calculations for b:

$$\frac{\partial \mathcal{L}_r}{\partial b} = \frac{\partial}{\partial b}\left[\frac{1}{2}\left(\sigma(wx+b)-t\right)^2 + \frac{\lambda}{2}w^2\right].$$

$$= \frac{1}{2}\frac{\partial}{\partial b}\left(\sigma(wx+b)-t\right)^2 + \frac{\lambda}{2}\frac{\partial}{\partial b}w^2$$

$$= \left(\sigma(wx+b)-t\right)\sigma'(wx+b)\frac{\partial}{\partial w}(wx+b)$$

$$= (\sigma(wx+b)-t)\sigma'(wx+b).$$

This gives a correct answer about the gradient that can be used for optimization but:

1. Calculations are cumbersome. It is error prone, and increasingly so with a realistic neural net.
2. A lot of redundancies. For instance, the first 3 steps in the two derivations are almost identical.
3. Both expressions have lots of repeated terms.
4. Backprop shares the repeated computations whenever possible. It is clean and modular.

Multivariate chain rule 2 variables

Recall the univariate chain rule:

$$\frac{d}{dt}f(g(t)) = f'(g(t))g'(t).$$

The multivariate chain rule is similar:

$$\frac{d}{dt}f(x(t), y(t)) = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}.$$

Convenient notation

We simplify notation a bit here let

$$\overline{v} \equiv \frac{\partial \mathcal{L}}{\partial v}.$$

We can rewrite the multivariable Chain rule

$$\overline{t} = \overline{x}\frac{dx}{dt} + \overline{y}\frac{dy}{dt}.$$

Here we use dx/dt to mean we should actually evaluate the derivative algebraically in order to determine the formula for $\overline{t}$, where $\overline{x}$ and $\overline{y}$ are values already computed in previous steps of the algorithm.

Using the computation graph

The technique of backprop of automatic differentiation (autodiff), using a computation graph:

$$z = wx + b$$

$$y = \sigma(z)$$

$$\mathcal{L} = \frac{1}{2}(y - t)^2$$

$$\mathcal{R} = \frac{1}{2}w^2$$

$$\mathcal{L}_r = \mathcal{L} + \lambda\mathcal{R}.$$

The computation graph has nodes that correspond to all the values that are computed with edges indicating which values are computed from which other values.

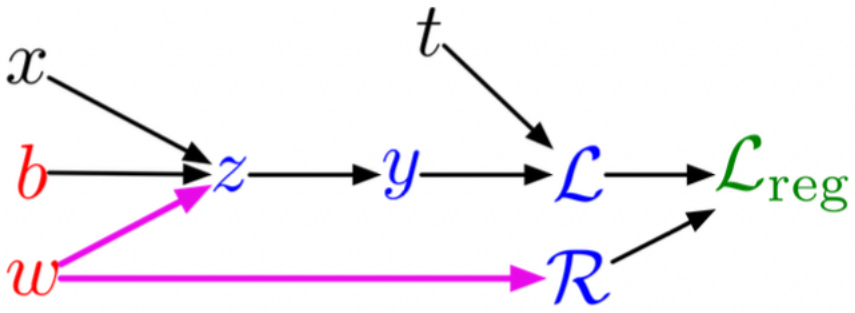

Figure 5.1: Computation Graph

The goal of backdrop is to compute derivatives $\overline{w}$ and $\overline{b}$. We do this applying repeatedly the chain rule.

We have to start with $\mathcal{L}_{reg} = \mathcal{L}_r$, and walk backwards trough the graph. This explains the name.

Let $v_1, \cdots, v_N$ be the notes of the computation graph, in topological order. (Topological ordering is any ordering where parents come before children.)

We want to compute all derivatives $\overline{v}_i$, even when we only want a subset.

We first compute all of the values in a **forward pass,** and then compute the derivatives in **backward pass.** So we let for this case, $v_N = \mathcal{L}_r$ or the result of the computation, we let $\overline{v}_N = 1$ by convention.

$$\overline{\mathcal{L}}_r = 1$$

$$\overline{\mathcal{R}} = \overline{\mathcal{L}}_r \frac{d\mathcal{L}_r}{d\mathcal{R}}$$

$$= \overline{\mathcal{L}}_r \lambda$$

$$\overline{\mathcal{L}} = \overline{\mathcal{L}}_r \frac{d\mathcal{L}_r}{d\mathcal{L}}$$

$$= \overline{\mathcal{L}}_r$$

$$\overline{y} = \overline{\mathcal{L}} \frac{d\mathcal{L}}{dy}$$

$$= \overline{\mathcal{L}}(y - t)$$

$$\overline{z} = \overline{y} \frac{dy}{dz}$$

$$= \overline{y} \sigma'(z)$$

$$\overline{w} = \overline{z} \frac{dz}{dw} + \overline{\mathcal{R}} \frac{dR}{dw}$$

$$= \overline{z} x + \overline{\mathcal{R}} w$$

$$\overline{b} = \overline{z} \frac{dz}{db}$$

$$= \overline{z}.$$

Now let's apply the method

$$\overline{\mathcal{L}_r} = 1$$

$$\overline{\mathcal{R}} = \overline{\mathcal{L}_r} \lambda$$

$$\overline{\mathcal{L}} = \overline{\mathcal{L}_r}$$

$$\overline{y} = \overline{\mathcal{L}}(y - t)$$

$$\overline{z} = \overline{y} \sigma'(z)$$

$$\overline{w} = \overline{z}x + \overline{\mathcal{R}}w$$

$$\overline{b} = \overline{z}.$$

The procedure has no redundant computations so the procedure is modular, it is broken into small chunks that can be reused for other computations.

For instance, if we wanted to change the loss function we just had to modify the formula for $\overline{y}$.

Chapter 6

Generative Adversarial Networks (GANs)

6.1 Introduction to GANs

Generative Adversarial Networks (GANs) are a groundbreaking approach in the field of artificial intelligence and machine learning, particularly within the realm of unsupervised learning. Proposed by Ian Goodfellow and his colleagues in 2014, GANs have revolutionized the way we think about generating new, synthetic data that is indistinguishable from real data.

6.2 What are GANs?

GANs consist of two main components: a Generator (G) and a Discriminator (D). The Generator aims to produce data that is similar to some training set, while the Discriminator evaluates the authenticity of the generated data. Through their adversarial training, both components improve their performance, with the Generator producing increasingly realistic data and the Discriminator getting better at distinguishing real from fake data.

6.3 Generative Models

The core idea behind generative models is to learn the underlying distribution of a dataset. This knowledge allows the generation of new data points that follow the same distribution. GANs, alongside Variational AutoEncoders (VAEs), are prominent examples of generative models, aim-

ing to replicate the statistical properties of input data.

6.4 Adversarial Training

The unique aspect of GANs is their use of adversarial training, where the Generator and Discriminator are trained simultaneously in a competitive setting. The Generator's goal is to produce data that the Discriminator cannot distinguish from real data, effectively "fooling" the Discriminator. Meanwhile, the Discriminator's objective is to accurately classify data as real or generated. This process leads to a dynamic competition, driving both networks towards improved performance.

6.5 Training a GAN

Training a GAN involves several key steps:

1. The Discriminator is trained first, using both real data and fake data generated by the Generator. It learns to classify real data as real and fake data as fake.

2. The Generator is then trained to produce data that the Discriminator will classify as real. This step is crucial for improving the Generator's ability to create realistic data.

3. The process is repeated, with both the Generator and Discriminator being iteratively trained to improve their capabilities.

6.6 Applications

GANs have found applications in a wide range of fields, from image and video generation to more complex tasks like unsupervised translation between different domains. Their ability to generate new, realistic data from learned distributions makes them an invaluable tool in areas requiring creative data synthesis and augmentation.

6.6.1 Random Number Generation in GANs

A crucial component in the training of Generative Adversarial Networks (GANs) is the generation of random numbers, particularly from a normal distribution. This process is essential for initializing the GAN's generator network, allowing it to produce diverse and complex data samples.

6.6.2 Importance of Randomness

Randomness introduces variability and novelty in the generated data. By sampling from a normal distribution, the GAN's generator learns to produce outputs that mirror the statistical properties of the input data distribution. This is pivotal for the generative model's ability to create realistic and varied data.

6.6.3 Implementation in Julia

In the provided Julia code, random number generation plays a key role in training a GAN to simulate data from a normal distribution. The script outlines the setup of a GAN environment and utilizes specific Julia packages for neural network modeling and optimization. Here are some highlights of the code's approach to random number generation:

- **Environment Setup:** The script begins by activating a new environment for the GAN project, ensuring that all necessary packages are available and version-controlled.
- **Random Seed:** A fixed random seed is set using the 'Random' package, ensuring reproducibility in the random number generation process. This is crucial for scientific experiments where consistency is key.
- **Sampling:** The GAN utilizes random numbers sampled from a normal distribution to feed into the generator network. This allows the generator to learn and adapt, producing data that is statistically similar to the target distribution.

6.6.4 Benefits for GAN Training

Using random numbers from a normal distribution enhances the GAN training process by:

1. Providing a foundation for generating diverse and complex data samples.
2. Facilitating the learning process of the generator to capture the essence of the target data distribution.
3. Ensuring reproducibility and consistency across training sessions through the use of a fixed random seed.

This approach underlines the significance of randomness in generative modeling, particularly for GANs aimed at producing realistic synthetic data.

This approach underlines the significance of randomness in generative modeling, particularly for GANs aimed at producing realistic synthetic data. Generative Adversarial Networks (GANs) present several compelling advantages over traditional Markov Chain Monte Carlo (MCMC) techniques for random number simulation, particularly in the context of efficiency and flexibility. One of the most notable benefits is GANs' ability to generate samples in parallel, as opposed to the sequential nature of MCMC methods, leading to significant improvements in computational efficiency. This parallel generation capability allows GANs to produce large volumes of samples quickly, making them highly suitable for applications requiring substantial amounts of synthetic data. Additionally, GANs are not constrained by the need to meticulously design proposal distributions, a critical step in MCMC that often requires domain expertise and can limit the flexibility of the sampling process. GANs learn to simulate the target distribution directly from data through adversarial training, automatically capturing complex dependencies without explicit modeling. Furthermore, GANs can generate high-dimensional data with relative ease, whereas MCMC methods might struggle with the curse of dimensionality, making GANs particularly advantageous for tasks in image and video generation, where the data is inherently high-dimensional. Overall, GANs offer a powerful and versatile alternative to MCMC techniques, with their ability to efficiently produce diverse and complex samples directly from data, bypassing many of the limitations inherent in traditional sampling methods.

Chapter 7

K-means and K-medoids

(Material adapted from Lester Mackey's slides).

7.1 Unsupervised learning

The world is filled with apparent high dimensional complexity however it may be that much of the underlying structure is low-dimensional. How do we uncover the hidden structure of categories underlying our data?

Unsupervised learning given covariates $x_1, \cdots, x_n$, we can infer the underlying structure.

Clustering: Group these unlabeled images into three clusters or groups.

We humans seem to be able to easily categorize a set of complex objects to simplify understanding, we can apply the same principle to machine learning.

Unsupervised learning is useful because most datasets are in fact unlabeled. Also they could be use to obtain compressed representations to save storage and computation.

They reduce noise, missing data, and irrelevant attributes in high-dimensional data.

It can be used also as a pre-processing step for supervised learning.

7.2 K-means

The objective of K-means is to assign each datapoint to one of k clusters so that no average is closer to its cluster mean.

Datapoints $x_i \in \mathbb{R}^p$, cluster mean is $m_j \in \mathbb{R}^p$, cluster assignment is $z_i \in \{1, \cdots, k\}$.

Objective: $J(z_{1:n}, m_{1:k}) = \sum_{i=1}^{n} ||x_i - m_{z_i}||_2^2$, where $|| \cdot ||_2^2$ is the squared Euclidean norm.

Goal: Minimize J over $z_{1:n}$ and $m_{1:k}$.

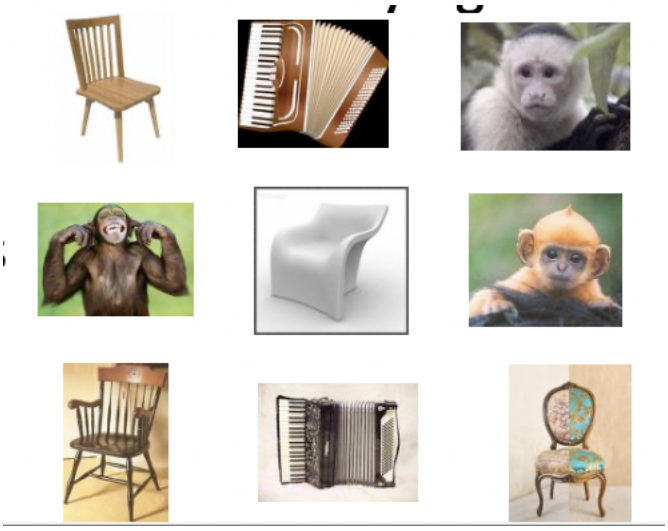

Figure 7.1: Example of image clustering for the students

7.3 Standard k-means algorithm/ Lloyd's algorithm

-Initialize cluster means arbitrarily (e.g. sample from datapoints)

-Alternate until convergence

- Update cluster assignments: $z_{1:n} \leftarrow argmin_{z_{1:n}} J(z_{1:n} m_{1:k})$, i.e., assign each point to the cluster with the closest mean.
- Update cluster means: $m_{1:k} \leftarrow argmin_{m_{1:k}} J(z_{1:n}, m_{1:k})$, i.e., $m_j = \frac{\sum_{i=1}^{n} 1(z_i=j) x_i}{\sum_{i=1}^{n} 1(z_i=j)}$, the mean of points in cluster j.

The objective J **always converges.**

Lloyd's algorithm is a coordinate descent procedure. Each step monotonically decreases the objective.

Only finite number of partitions of data, so the objective must converge in finite number of steps.

Technically the algorithm could cycle if ties arise (multiple centroids equidistant from point).

We can avoid ties by assigning always the point to the smallest centroid under some total ordering of vectors.

7.4 K-means limitations and Categorical Data

The Euclidean distance is restrictive for certain dataset like categorical features. Also the Euclidean distance is sensitive to outliers, and ill-suited for datasets with very different units/scales.

Also the K-means optimization problem is NP-hard. Lloyd's algorithm usually finds suboptimal solutions. Many random restarts often needed to obtain good performance.

Most important **the user must choose** k.

Running time: #features $\times$ datapoints $\times k$ per iteration.

Generalized problem:

Minimize $J_d(z_{1:n}, m_{1:k}) = \sum_{i=1}^{n} d(x_i, m_{z_i})$.

Arbitrary dissimilarity/discrepancy measure $d(x, m)$.

Optimize via coordinate descent as in Lloyd's algorithm.

The great advantage is that it applies to all data types and dissimilarity measures.

Updating cluster representative $m_{1:k}$ may be expensive.

7.5 K-medoids algorithm

Minimize J_d above but constrain each cluster representative to be a datapoint i.e., $m_j \in \{x_1, \cdots, x_n\}$.

Don't need to store datapoint, only pairwise discrepancies. $d(x_i, x_j)$.

7.6 Choosing the number of clusters k

Best case known beforehand.

Elbow method.

Gap statistic.