Economics 821 Duke University
Christopher Timmins

Hedonics and Non-Market Valuation:
Empirical Problem Set

In this problem set, you will complete a series of structured empirical exercises using a variety of methods and techniques from the hedonics literature. This will give you an opportunity to get some hands-on experience with the theories that we are going to be studying during this module. To facilitate this process, I will provide you with data describing housing transactions and violent crime rates for the Los Angeles and San Francisco Metropolitan Areas between 1993 and 2008. These data have been used in a number of projects, and have been found to "work well". They have been cleaned and organized into a variety of samples that will be of particular use for specific exercises. Therefore, your efforts will be primarily directed towards estimation, not data management.

There are two goals for this empirical problem set. (1) To give you first-hand experience with hedonic modeling. (2) To introduce you to computer programming for applied work in microeconomics. You will therefore be required to complete these exercises using one of the following computer languages: Matlab, C++, Fortran. If you have experience programming with another language that you think is comparable (e.g., R may work), you can use it with instructor approval. I will provide support (access to a compiler, example code) for Fortran. There are two reasons for forcing you to do this exercise in one of these languages. (1) While some parts of the exercise could be carried out in a language such as Stata, others will require a language where you can program estimators yourself. If you continue on to take the second module (Sorting Models), there will be many more estimation problems where this will be the case. (2) Becoming familiar with programming in one of these languages opens the door for you to a variety of empirical techniques (besides hedonics) that may prove useful in your dissertation research.

The empirical problem set is ordered into six "tasks" that build upon one another. You might think of each as taking roughly a week to complete, although the early tasks will not take nearly this long, while later ones could take longer.

While I am not qualified to teach a formal class in Fortan, I can provide you with simple programming advice in class before each task is due. After the task is due, I can provide you with Fortran source code so that you can see exactly how to do it before moving on to the next task.

1. Data Loading and Summary Statistics

You will be given two data set: la_data.txt (n=386063) and sf_data.txt (n=378252). Each data set contains the following variables (in order):

- *House ID #*
- *Price (deflated to year 2000 dollars)*
- *County ID # (see below)*
- *Year Built*
- *Square Footage*
- *# Bathrooms*
- *# Bedrooms*
- *# Total Rooms*
- *# Stories*
- *Violent Crime Rate (Cases per 100,000)*
- *Property Crime Rate (Cases Per 100,000)*
- *Year of Sale (1993 – 2008)*

County identifiers refer to California state FIPS codes:

LA:		SF:	
37	Los Angeles	1	Alameda
59	Orange	13	Contra Costa
65	Riverside	41	Marin
71	San Bernadino	55	Napa
111	Ventura	75	San Francisco
		81	San Mateo
		85	Santa Clara
		95	Solano
		97	Sonoma

You should drop SF data for Marin, Napa, Solano and Sonoma counties. Write a separate script for each data set that reads in each variable and calculates its mean and variance.

Report these means and variances (separately for each county, organized by city) in an easy-to-read table.

2. Bootstrapped Hedonic Price Function

Write a separate script for each city that estimates 500 bootstrapped hedonic price functions.[1] Each price function should contain:

- *Constant*
- *# Bathrooms*
- *# Bedrooms*
- *# Stories*
- *Property Crime Rate*
- *(Property Crime Rate)*^{2}
- *Year Built*
- *(Year Built)*^{2}
- *Square Footage*
- *(Square Footage)*^{2}
- *# Total Rooms*
- *(# Total Rooms)*^{2}
- *Violent Crime Rate*
- *(Violent Crime Rate)*^{2}
- *Vector of Year Dummies (omit 1999)*
- *Vector of Dummies for Certain Counties*[2]

Save the 500 pairs of coefficients for *Violent Crime Rate* and *(Violent Crime Rate)*^{2} in a separate file. You will use these in the next task.

Bootstrapping Hint: For a data set size n, create a vector length n of random integers ranging from [1,n]. Random numbers should be generated with replacement (i.e., you may draw the same random number more than once). Call this random vector $\bar{r}$, where *r(i)* is the random draw corresponding to observation *i*. You can then create your bootstrap data set (e.g., for some variable *x*), by simply re-assigning values according to the random number draws. Your code may look something like this:

```
do i = 1,n
   boot_x(i) = x(r(i))
enddo
```

Calculate your point estimates by using the actual data set for each city, re-assigned to observations in the bootstrapping algorithm. Calculate standard errors by taking the standard deviation of your bootstrapped parameter estimates.

Report point estimates and standard errors for each regressor (separately by city) in an easy-to-read table.

[1] In each bootstrap iteration, you need to create a new data set with the same number of observations found in the actual data set, but created by randomly sampling observations with replacement.
[2] For Los Angeles, include explicit dummies for counties 59, 65, and 71. For San Francisco, include dummies for counties 13, 75, 81, and 85.

3. Bootstrapped Multimarket Rosen Estimates

Using the 500 bootstrapped hedonic price gradients from task #2, carry out the same number of bootstrapped estimates of the Rosen model (i.e., for each bootstrapped Rosen procedure, use one of the bootstrapped price gradients from the previous task). In order to do so, you will need to read in an additional data set that contains information about homebuyers in both San Francisco and Los Angeles. "buyer_data_sf_la.txt" (n=659,548) contains the following variables (in order):

- *Buyer ID #*
- *Price (deflated to year 2000 dollars)*
- *Violent Crime Rate (cases per 100,000 residents)*
- *Property Crime Rate (cases per 100,000 residents)*
- *Race (2 = Asian/Pacific Islander, 3 = Black, 4 = Hispanic, 5 = White)*
- *Income*
- *LA Indicator (= 1 if resident lives in LA, = 0 otherwise)*

For each bootstrapped hedonic price gradient found in task #2, you will need to generate a bootstrapped data set of home buyers. Assign individuals implicit prices for crime from the appropriate bootstrapped hedonic price gradient (given their city of residence) and carry-out OLS regression.

For your Rosen specification, regress the implicit price of violent crime on the violent crime rate chosen by each individual, and a vector of individual attributes. You could have alternatively regressed the violent crime rate on the implicit prices of violent crime and property crime, but that will be more difficult to compare to the other results that come later in the problem set.

As above, calculate your point estimates by using the actual data set (i.e., not boostrapped) both for the calculation of the first stage hedonic price function in each city and the second stage (estimation of the MWTP function). Calculate standard errors by taking the standard deviation of your bootstrapped parameter estimates in each stage.

Question: Do the results that you get seem sensible (i.e., do they correspond to your priors given economic theory)? Explain why or why not.

4. Estimating a Non-Parametric Hedonic Price Function

For this task, you will need to use "local-linear" estimation techniques to recover a non-parametric representation of the hedonic gradient. Non-parametrics are attractive for the first stage of a hedonic analysis, as theory does not dictate what the shape of the hedonic price function should be. Any parametric assumptions we impose on the price function are, therefore, arbitrary.

Bajari and Khan (2005) use local-linear techniques to recover their MWTP estimates. With their technique, they demonstrate that MWTP estimates can be obtained with data from a single market – we will follow them in this regard and work with just data from Los Angeles and we will pool data over time (read in "la_data.txt", which you used in task #1).

To keep things simple, you will calculate the hedonic gradient in a single dimension (violent crime rate). You should impute values for the gradient (i.e., the derivative of the hedonic price function) at evenly spaced points on a grid of violent crime rates (χ = 1, 2, 3,…, 2000).

Given a choice of bandwidth, h, the Gaussian kernel is given by:

$$K_h\left(VC_j - \chi\right) = \frac{1}{h\sigma_{VC}} \frac{1}{\sqrt{2\pi}}\, exp\left\{-\frac{1}{2}\left(\frac{VC_j - \chi}{h\sigma_{VC}}\right)^2\right\}$$

This determines the weight placed on a particular observation VC_j when estimating the linear equation that describes the hedonic price function at χ. We are primarily interested in the slope of that function, which can be found by solving the following minimization problem at each point χ:

$$\min_{\{\alpha(\chi),\beta(\chi)\}} \sum_{j=1}^{J} \left(P_j - \alpha(\chi) - \beta(\chi)VC_j\right)^2 K_h(VC_j - \chi)$$

This minimization problem has the following closed form solution:

$$\begin{bmatrix}\alpha(\chi)\\ \beta(\chi)\end{bmatrix} = \left(VC'W_\chi\, VC\right)^{-1} VC'\, W_\chi\, P$$

where

$$W_\chi = diag\{K_h(VC_j - \chi)\}$$

Plot your hedonic price gradient (i.e., $\beta(\chi)$, χ = 1, 2, 3,…, 2000) for the following values of the bandwidth parameter: h = 1, 3, 10, 1000.

5. MWTP Estimates Based on Non-Parametric Gradient (Bajari & Benkard, 2004)

The file "buyer_data_la.txt" (n=55,498) contains the following variables:

- *Buyer ID #*
- *Price (deflated to year 2000 dollars)*
- *Violent Crime Rate (cases per 100,000 residents)*
- *Property Crime Rate (cases per 100,000 residents)*
- *Race (2 = Asian/Pacific Islander, 3 = Black, 4 = Hispanic, 5 = White)*
- *Income*

describing a randomly selected 5% sample of all the homebuyers in Los Angeles. Use these data along with the estimated hedonic price gradient from task #4 (h = 1) to estimate a function describing how MWTP varies with income and race:

$$MWTP_i = \beta_0 + \beta_1 INC_i + \beta_2 ASIAN\_PI_i + \beta_3 BLACK_i + \beta_4 HISP_i + \varepsilon_i$$

Calculate standard errors using standard regression asymptotics (i.e., you can ignore the fact that MWTP is a generated from a first-stage regression).

Report point estimates and standard errors in an easy-to-read table.

6. Bishop-Timmins Estimator

Begin this task by repeating task #2, but using a 6^{th} order polynomial in violent crime (this is the poor-man's way of running a non-parametric estimator). Save your 500 bootstrapped coefficient estimates for VC, VC^2, etc… in a separate file.

Using the same data you used in task #3, set up the likelihood maximization problem described in Bishop-Timmins (2017) so as to recover the following MWTP function:

$$MWTP_i = \lambda_0 + \theta VC_i + \lambda_1 INC_i + \lambda_2 ASIAN\_PI_i + \lambda_3 BLACK_i + \lambda_4 HISP_i + \varepsilon_i$$

Report point estimates and bootstrapped standard errors for MWTP function. How does it differ from the MWTP function estimates you found in task #3?

Fortran

Appendix 1: Compiling and running code in Fortran on the Duke Econ server

If you are a student in the economics department, you already have access to a Fortan compiler and IMSL libraries. If you are not a student in the economics department, contact me and we can arrange for you to be given an account.

In order to run a Fortran script (i.e., a text file with a name like "program.f90" that contains a list of commands), carry out the following steps:

(1) Log on to the Duke Econ server (login.econ.duke.edu). You will need an account on the Econ Server in order to do so.

(2) Go to your working directory (i.e., where you are storing your Fortran script and any data sets you want to use).

(3) Type "sourse setup-imsl". This will allow your program to access the IMSL libraries maintained on the Duke Econ server.

(4) Use the following command to compile your script:

```
$F90 -o PROGRAM.EXE $F90FLAGS PROGRAM.f90 $LINK_F90
```

This will create an executable file called PROGRAM.EXE.

(5) Run your script by typing "./PROGRAM.EXE".

Appendix 2: IMSL Documentation

The IMSL MATH/LIBRARY and STAT/LIBRARY are collections of FORTRAN subroutines and functions useful in research and statistical analysis. Each routine is designed and documented to be used in research activities as well as by technical specialists.

To use any of these routines, you must write a program in FORTRAN (or possibly some other language) to call the MATH or STAT/LIBRARY routine. Each routine conforms to established conventions in programming and documentation. IMSL gives first priority in development to efficient algorithms, clear documentation, and accurate results. The uniform design of the routines makes it easy to use more than one routine in a given application. Also, you will find that the design consistency enables you to apply your experience with one STAT/LIBRARY routine to all other IMSL routines that you use.

Documentation for all of the IMSL routines you will use can be found on-line at:

http://www.absoft.com/Support/Documentation/fortran_documentation.html

Look for the following links at the bottom of the page:

IMSL Fortran Library User's Guide Math/Library Volume 1 of 2
IMSL Fortran Library User's Guide Math/Library Volume 2 of 2
IMSL Fortran Library User's Guide Math/Library Special Functions
IMSL Fortran Library User's Guide Stat/Library Volume 1 of 2
IMSL Fortran Library User's Guide Stat/Library Volume 2 of 2

One routine we will use very early is RLSE, which fits a multiple linear regression model using ordinary least squares. You can find the description of this routine in Stat/Library Volume 1, p.110. Other routines can be found by searching the tables of contents and their hyperlinks.

Appendix 3: Web Resources for Fortran Programming

A couple of online manuals where you can go for advice on Fortran coding…

Fortran 77: http://www.star.le.ac.uk/~cgp/prof77.html

Fortran 95: http://www-eio.upc.edu/lceio/manuals/Fortran95-Manual.pdf