AHSEplorer

James Lampton, Sonia Ng, Swetha Reddy, Di-Wei Huang
University of Maryland

Abstract
AHSEplorer is an application developed to explore the information within the American Housing Survey (AHS). In particular, AHSEplorer provides a means of describing transition patterns for housing states and provides three means of exploring the data. AHSEplorer is used to show the patterns of how the houses change over time, especially the dynamics of how affordable rental houses are formed based on the data from the AHS.

Introduction
Where does affordable rental housing come from? Where does it go? What policies are effective in maintaining affordable rental housing? Researchers at George Washington University are currently examining information from the American Housing Survey to find new insights on affordable housing.

The American Housing Survey is the main source for characteristics of American housing. It is conducted by the U. S. Census Bureau to provide up-to-date housing statistics for the Department of Housing and Urban Development [1]. The AHS is actually two surveys one national and one metropolitan. Since the year it was founded in 1973, the AHS has interviewed three samples of around 50,000 to 60,000 housing units each. The summarized data set we’re using was created to provide longitudinal data on the affordability of rental housing in the United States. It utilizes information from three sources: the Area Median Income Files (AMI), the Housing Affordability Data System (HADS) of the Department of Housing and Urban Development (HUD) and the Public Use Files of the American Housing Survey.

Our focus has been on analyzing the trends in affordable rental housing. Initially we explored the transitional probabilities of housing units between years using statistical methods like forward and backward analysis using Spotfire [2]. We then developed a custom application, AHSEplorer, which allows for further exploration of this data set. There are three main components of AHSEplorer. The first is an overview that displays the transition probabilities using a stack bar chart and provides access to the other views. Our next view allows for exploration of the correlation of behaviors to additional attributes (age, number of units, number of rooms, etc). The final view is a time series plot that displays the paths a house takes over years showing trends and pattern that houses follow in the market.
Related work

Spotfire was used to examine this data set [3], this lead to the development of the time series graph for better control over selection. The transition patterns were introduces to help focus the analysis and visualizations around affordable rentals. Much of the original work with the time series graph was inspired by application of motif extraction and clustering on time series data, [4] and [5], as well as the selection schemes provided by TimeSearcher [6]. Due to prioritization from the customer, most of this was shelved to focus on the transitional visualizations. Different means of expressing the transition probabilities along with their attributes were considered, [7] and [8], but ultimately a simple stacked bar chart was used.

Data Description

The way the affordability of rental housing units has been measured and classified is relative to HUD AMI. For the year 2005, the HUD produced estimates of the income of the median family for 365 metropolitan areas and 2,302 non-metropolitan areas [9]. Privacy issues have also been taken into consideration. Therefore, the AHS has reduced geographical details for housing units found in areas with a population of less than 100,000. These actually compose about 50% of the data sample. However, with the information provided by the HADS, we can still have a consistent way to classify housing units in terms of affordability.

We made use of the AHS data sets from 1985 to 2005. Although there is a 2007 dataset available, it is not as complete as the 2005 data. The reason for this is because the Census Bureau did not include about 5,000 housing units in the 2007 data for budgetary reasons. Therefore, to avoid problems in terms of dataset comparisons, we are working with data up to 2005.

The data from AHS was converted to a set of time series represented as a series of encoded categories. Each category encodes the state of a house in a 2-year period from 1985 to 2005. There are 8 possible states (0 - 8) ranging from not build to destroyed. The sample also includes information about other attributes of the unit like its age, number of rooms, locality etc. The data is for ten cities in the US.

To further simplify the information for visualization, the information is summarized further using transition patterns. AHSEXplorer gives user the flexibility of defining the transitions, but the default transitions are: Constant (from affordable to affordable), Gentrify (from affordable to moderate or expensive rental), Loss (from affordable to uninhabitable or destroyed), and Tenure change (from affordable to owner occupied).

Spotfire

For the first task of visualizing the transition probabilities we began with using Spotfire to plot the transitional probabilities of forward and backward analysis between 1985 and 2005.
Figure 1 Scatter plot showing forward analysis of housing data. Note that Los Angeles has more affordable housing being gentrified than remaining constant. This can be compared to a later figure from AHSExplorer displaying similar information.

Figure 1 shows the four transition (constant, gentrify, tenure change and loss) of forward analysis represented by different shapes and colors for the ten cities ordered by increasing percentage of gentrify.

Figure 2 Scatter plot representation of backward analysis of the housing data.
Figure 2 shows the five transitions (constant, new construction, tenure change, restore and filter down) of backward analysis represented by different shapes and colors for the ten cities ordered by increasing percentage of filter down. Our customer initially preferred the scatter plots from Spotfire instead of the stacked bar chart. We now believe the scatter plot is probably better for placement in print. However, we believe that the stack chart provides a better means of interactive visualization.

**AHSEExplorer**

After exploring the data using existing tools, we set out to develop and application to focus specifically on the features in the AHS data set. Our tool, called AHSE Explorer is a Java application written using Processing [10] to draw custom widgets and Swing for control and layout. It was engineered using the Model View Controller [11] design pattern and is primarily intended for embedding via a Java Applet. AHSE Explorer is comprised of three major views: a stacked bar chart overview screen, an attribute explorer, and a time series display.

The initial screen presented to the user is the stacked overview. Our client was interested in examining transition probabilities between different years. Because they were studying affordable housing, all named transitions were centered on affordable housing. To provide flexibility, the tool allows for the creation of new transition patterns. This screen provides a very quick overview of the behavior of housing in each city. It also provides sufficient ordering criteria to allow users to develop insights about the data. Consider figure 3 which shows such an ordering. In this image, we can see that Los Angeles demonstrates some unique properties. Over the 20 years studied, the highest percentage of sampled affordable housing was lost to gentrification (the only city to show this behavior). Furthermore, we can see that it lost the least amount of housing to loss.
Figure 3 Overview screen from AHSExplorer showing cities ranked by percent of housing remaining constant. Each column is sorted in an ascending order by transition probability. This can be compared to the scatter plot from figure 1.

Examination of cities by percentage removes the ability to reason about the sample size of each city. In figure 4 we can see how the cities compare by instance count. In this image, we can see that New York had the highest number of instances, 300. We can also see a specific ordering of the individual cells within the stacks. This provides a means of doing a direct comparison between cells within each city by providing a consistent ordering across the chart.
Figure 4 Overview screen showing stacks sized by count. The cities are ranked by losses. The stack cells use a fixed order. This view provides a better comparison of the number of instances in each category.

In our initial implementation, we are able to examine all cities for two given years. This provides a very quick and useful means of examining long term effects. However, it tends to hide behaviors caused by temporal events such as a housing boom. Users could explore different timeframes by altering one of the endpoints, but we believe this dataset and application would benefit from a rank by feature [12] capability. It would allow users to easily explore new questions such as: “between which timeframes did the most affordable housing undergo a tenure change?” As we show later, a basic time series of this data doesn’t lend well to analysis.

Once the user finds an interesting data point, they’re able to select the specific city and examine the attributes for all the instances. Our original design called for insertion of visual artifacts to represent some of these attributes (inspired by [7]), but we were unable to come up a satisfactory representation. Instead, we provided a separate view to allow users to explore the data set further. Our current implementation makes use of the same frame to display the attribute view (temporarily hiding the overview). This is most likely detrimental to a users work flow, but was deemed necessary for hosting this application on a web page. Each component was developed in such a way that they could be viewed in separate frames but a window switch would still be necessary. Scaling these widgets down would require further and careful design to improve the information density while still maintaining readability.
In figures 5 and 6 we can see the breakdown of attributes for Washington, DC. For each attribute and transition pattern, there is a bar graph showing the number of instances with each attribute value. Figure 5 shows the information sized by the percentage within each category. As you can see, buildings from the 1930’s contribute quite significantly to the houses lost to gentrification. As with the overview chart, switching to a count based view allows for further analysis. In figure 6, we can see that older houses tend to make up a large portion of the affordable housing that remained so between 1985 and 2005. We can also observe that houses from the 1960’s appear to stay within the affordable range the most and contribute very little in loss and gentrification. In fact, they seem to be less popular than houses from the 30’s, 50’s and 70’s for gentrification; whereas houses from the 40’s contribute the most to loss (uninhabitable or destroyed).
Figure 6 Attribute view of Washington, DC data now sized and ordered by count.

The information in this view is fairly static aside from the controls on the screen. Tooltips are used to provide full information for each cell should the user mouse over them. In a later iteration, attribute values could be used as a means to select and filter housing instances (updating the other views in the process).

The final view can be seen in figures 7 and 8. Given a selected city (or the whole data set), the user can examine the paths of all housing instances. The list on the right provides a means of selecting instances to be highlighted. There is a search mechanism via regular expressions (see figure 8). This search mechanism could provide the basis for mouse generated search patterns much like TimeSearcher [6].

The information in this screen is not summarized. This screen would be used for more advanced users that are interested in finding new patterns or do deeper analysis of housing data across time. In that respect, this view is certainly incomplete. In the overview screen, a user is able to make discoveries without advanced capabilities such as rank by feature. Unfortunately, due to the nature of the housing data, a simple time series graph is not as helpful. In later versions of AHSExplorer it would make sense to add motif extraction [4] and visualization capabilities [5]. The transition patterns themselves could be considered very simple motifs (of highest importance to our customer).
Figure 7 Time series plot of Washington, DC data showing highlighting from selection. Note that all data starts from affordable due to the filtering of the transition probabilities.

Figure 8 Time series plot showing the result of a query (... affordable ... moderate ... expensive ...) against the entire data set.
Future work

As we’ve mentioned throughout the text, we believe there are significant features that could be added to improve AHSEplorer. These include the addition of attribute summaries in stack cells (space permitting). Allowing for each stack to be aligned vertically at the base of a specific transition (currently, they’re all aligned at the bottom), this would be similar to the alignment feature in Lifelines [13]. Addition of a scatter plot capability to provide an alternative view to the stack chart.

For the time series graph, we believe there are numerous improvements that could be made. These include the generation of regular expression searches from user mouse actions. Further summarization of instances based on motif extraction would probably lead to the discovery of unique and interesting transition patterns within the data.

We believe that the addition of a rank by feature [12] capability would be beneficial in discovering interesting data points without having to manually search through all the start/end pairs. Additional selection and ordering could be introduced that makes use of the additional attributes.

Switching to a different toolkit (such as Piccolo2D [14] or prefuse [15]) may be in order. Embedding Processing sketches into larger applications proved to be quite problematic and consumed far too much development time. AHSEplorer is most likely outside of the design considerations of Processing (providing rapid development of visualizations to non-programmers).

Conclusion

Though our scope reduced as we progressed, we believe we have developed a useful tool for the exploration of affordable housing. We look forward to working further with our sponsor to host our application so that researchers and policy makers may gain insight to such an important problem.

We may be able to cherry pick a future work from above (rank by feature) to improve the chances of insight generation? What do our kind reviewers think?

Acknowledgements

We would like to thank our customer and project sponsor Daniel Broxterman for allowing us to work on such an interesting project. We would like to thank Professor Shneiderman for direction and instilling us with Information Visualization knowledge. We would like to thank the other students of CMSC734 for feedback during our review. We would also like to thank FILL_ME_IN for reviewing our draft and providing (great/meh/rotten tomatoes;) feedback and recommendations. We would also like to thank TIBCO for providing licenses for Spotfire.

References


