Math 444: Midterm Project

Levent Batakci

March 28, 2021

Throughout history, humanity has had a rich tapestry of aspirations. However, one common theme throughout has been the drive to improve the quality of life. Naturally, this makes quality of life a pertinent topic to study.

Since quality of life is inherent to the human experience, we look to data concerning places where people live. Seeing as cities are abundantly populated, they are a natural choice to examine. In the course of this paper, we will explore the quality of life of cities across the United States of America.

All of the code and implementations can be found publicly on the GitHub Repository

City Data Overview

At its core, quality of life is subjective concept. For example, people are likely to differ in how important they believe different aspects of daily life to be in determining quality of life. That being said, quantitative methods are extremely powerful and can be applied with great efficacy to subjective topics. In order to achieve meaningful results when studying a subjective topic, it's imperative to work under reasonable assumptions and claims. Furthermore, all interpretations need be considered in the context of these assumptions and claims.

The data we're working with considers city ratings in 9 different categories - each of which can be argued to exert reasonable influence on quality of life. Specifically, the data consists of $ratings^{(j)} \in \mathbb{N}^9$. The data pertains to 329 cities, so $ratings \in \mathbb{N}^{9\times 329}$.

The 9 categories are: climate, housing, health, crime, transportation, education, arts, recreation, and economics. In each of these categories, a higher score is *better*. That is, a higher crime score indicates less (not more) crime. Similarly, a higher climate score indicates a climate that is generally perceived as better by the people of the city.

Oddly, these scores do not seem to be subject to any form of normalization. Particularly, the range of the arts category consists of considerably higher values than all the other categories. This leads us to believe that this structure is intentional, and that varying weight has been placed on aspects of life in the city.

To further investigate this potentially meaningful weighting, we computed the median of ratings for each category. We chose to use the median as a measure of center since it is resistant to outliers. We sorted these medians and produced a bar graph to visualize the difference in scale between the category ratings. This graph can be seen in **Figure 1** on the next page.

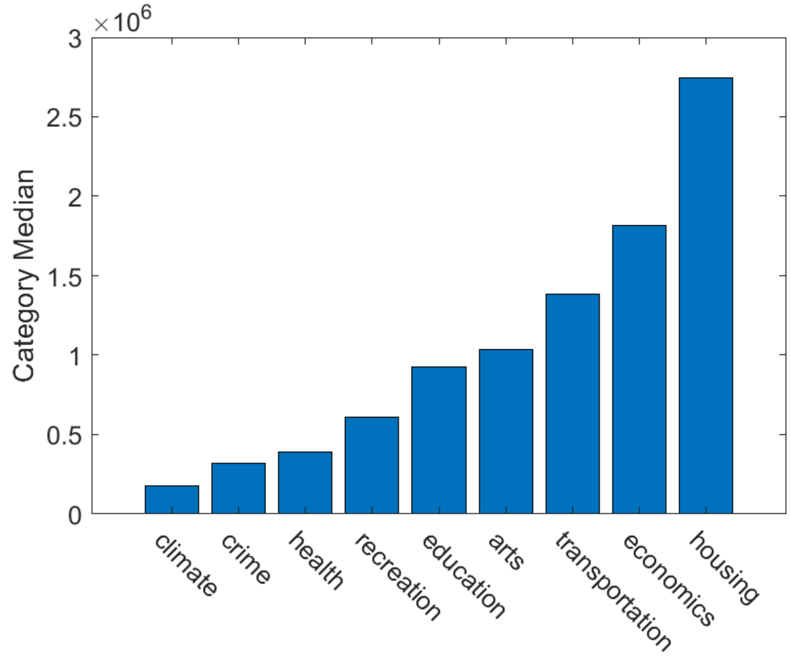


Figure 1: Sorted Category Median Ratings

We find this visual to be surprising overall. While it makes sense for housing ratings to have such a large scale, it is a bit shocking that health and recreation are small comparatively. This leads us to question whether the scaling of the data actually is meaningful. In the appendix we briefly discuss a possible normalization of the data and the corresponding effects on our analysis.

With this discussion in mind, we proceed to analyze the raw data for hidden insight.

Telling Cities Apart

Firstly, we wanted to investigate which categories separate cities the most. That is, what aspects of life vary the most from city to city? Naturally, we chose to perform a principal component analysis (PCA) to answer this question.

Essentially, PCA decomposes data by using singular value decomposition. The columns of U are feature vectors u_j along which spread is maximized. As usual, we center the data when computing the PCA. This step is done expressly so that a feature vector is not wasted trying to describe the location of the data about the origin.

Since spread is maximized along the feature vectors, we can use them to identify which categories best differentiate the data. To discuss this question visually, we created bar graphs showing the relative influences of the 9 categories on the feature vectors. Note that when it comes to differentiating data points, negative feature components are just as meaningful as positive ones. For this reason, we considered the absolute values of the feature vector components.

We looking at the first two feature vectors, aiming to identify the most prominent categories. Bar graphs showing this effect can be seen in **Figures 2 and 3** on the next page.

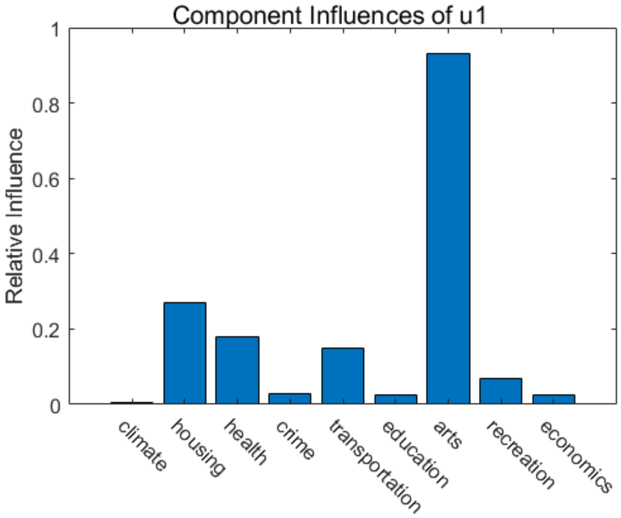


Figure 2: u1's Component Influences

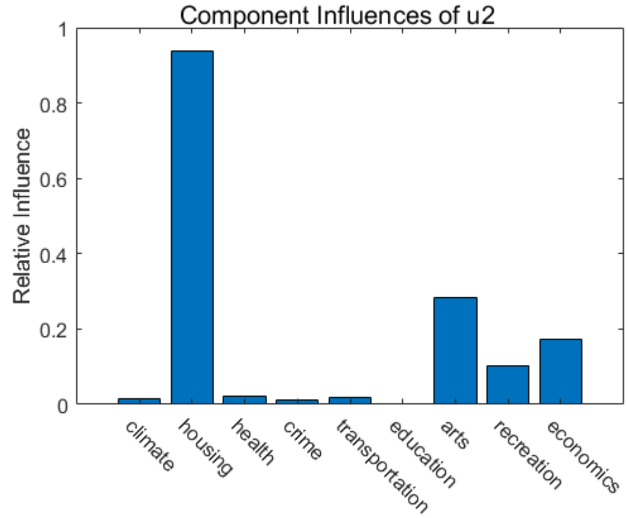


Figure 3: u2's Component Influences

Note that the first feature vector is the one along which spread is maximal. Looking at **Figure 2**, we see that **arts** is by far the most influential category when it comes to differentiating cities. Furthermore, **housing, health, and transportation** also seem to be significant, but much less so.

The second feature vector also represents a high-spread direction. Looking at **Figure 3**, we see that **housing** is by far the most influential category in this direction. Furthermore, **arts and economics** also seem to be significant, but much less so. It could be argued that recreation is also a significant differentiator, but we believe it's influence is too low.

Combining this information, we can conclude that **climate, crime, education, and recreation** are categories which have little utility when it comes to separating the data. We can thus conclude that factors are *relatively consistent* in cities across the United States. Consistency of crime is very surprising.

Likewise, we can conclude that the presence of arts and the quality of housing *vary greatly* across the United States. While the variance of art presence is somewhat unsurprising, we found the variance of housing scores to be both unexpected and concerning. Good housing is such a necessity, and so it is cause for concern that there isn't consistency in this sense across the United States.

Search for Natural Clustering

Next, we chose to explore whether there is a natural clustering in the data. In previous work with clustering, we had a general idea of how many clusters to expect - here, we do not. For this reason, we attempted clustering the data into various numbers of groups. In all cases, we used the k-medoids algorithm and euclidean distance to compute the clusters.

The k-medoids algorithm will always return a labeling which partitions the data into k clusters, so there is not guarantee that these clusters are in any way natural. If they are natural, however, we should be able to separate them somewhat cleanly in space. Specifically, we chose to use the LDA algorithm to visualize the clustering results.

LDA, which stands for Linear Discriminant Analysis, is a technique in which separating directions are computed. Along these seperators, the clusters are as tightly packed within themselves and distanced from the others as they can be. Similarly to PCA, we center the data before performing an LDA in order to not waste a separator. This is an effective way of evaluating how natural a clustering is.

Furthermore, for data with k clusters, LDA computes $k-1$ separators. We tried separating the cities into $k = 2$, 3, and 4 clusters. In the case of $k = 2$, only one separator is computed, so we visualize the clustering as a histogram. The results can be seen in figure **4** below. Of course, the two different colors correspond to the two clusters.

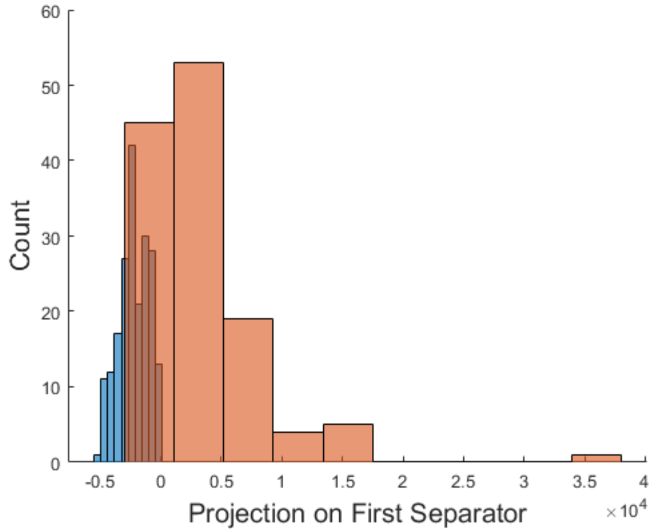


Figure 4: K-Medoids Results with k=2

These results are rather unimpressive, so we decided to look to larger k-values. Next, we tested $k = 3$ and $k = 4$. To assess both of these, we projected the data onto the first two separators and created a 2D scatterplot. The different colors represent different clusters, and the filled black circles represent the computed medoids. The results are shown in **Figures 5 and 6**.

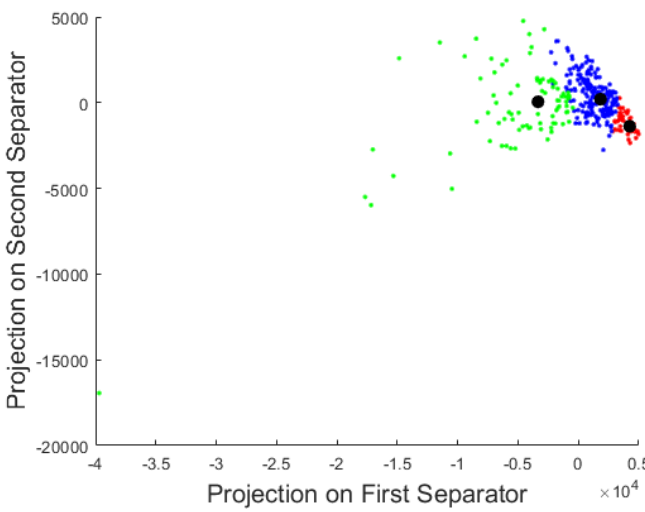


Figure 5: k=3 Results

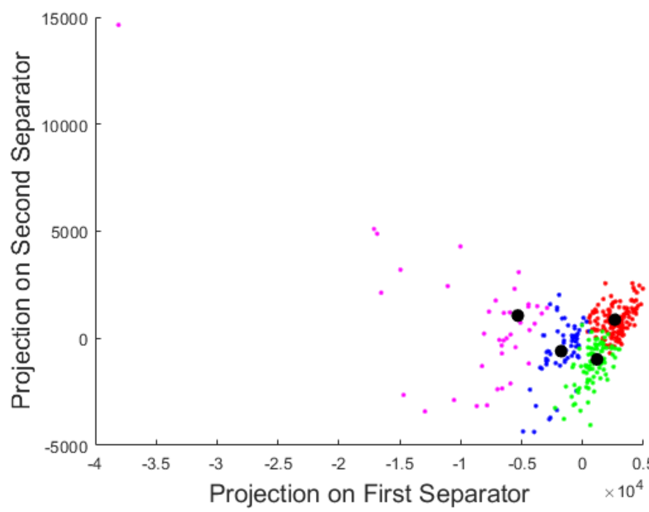


Figure 6: k=4 Results

Immediately, we see that the results are far better! One thing to note is that these results should be taken with a grain of salt, as k-medoids works in a way that makes it so clusters don't overlap much anyway. That being said, we still conclude that there could reasonably be a natural clustering in the data.

We find both of these results to be fairly compelling, though we believe **Figure 5** to be a slightly cleaner result. This is because the points appear less cluttered and seem to condense towards the red group. We also tested $k = 5$, but found the results to be fairly bad - this is shown in the appendix. For this reason, we did not test any $k > 5$.

We also identified which cities the computed medoids correspond to. For $k = 3$, we found the red, blue, and green medoids to be *Alexandria, LA*, *Roanoke, VA*, and *Providence, RI*. A point to note is that none of these cities represent the West side of the United States. That being said, we don't know if quality of life varies according to geographic locaition - so this is not necessarily an issue.

For $k = 4$, we found the red, blue, green, and magenta medoids to be *Montgomery, AL*, *Phoenix, AZ*, *Bismarck, ND*, and *Milwaukee, WI*, respectively. This result is rather interesting for the following reason: except for *Milwaukee*, all of the listed cities are the capitols of their respective states! This perhaps hints that state capitols are in some sense representative of different classes of American city quality of life.

Overall, we conclude that the data does in fact support some kind of natural clustering - perhaps into 3 or 4 groups. These groups likely correspond to different cities with different quality of life aspects. It makes sense that some quality of life aspects could be associated with others. For instance, it's feasible that cities with good economies are likely to have good housing as well. A more in-depth study considering the separators would be required to answer such questions.

Non-negative Matrix Factorization

As discussed in the introduction, the data *ratings* $\in \mathbb{N}^{9\times 329}$. So it follows that the data actually consists only of positive entries. This remark led us to explore the non-negative matrix factorization (NMF) of the data.

An NMF factors a matrix $X \in \mathbb{R}_+^{n\times p}$ of non-negative entries into the product of two matrices $W \in \mathbb{R}^{n\times r}$ and $H \in \mathbb{R}^{r\times p}$. Since such a factorization is not guaranteed, $X \approx WH$ is an approximation. The parameter r is a variable which denotes the rank of the approximation.

In a PCA, the feature vectors generally contain negative entries to enforce mutual orthogonality. This leads to the issue of feature vectors not being faithful to the data they represent (if the data is non-negative). The use of NMF is a way to get around this problem. Since the columns of W are positive, they can sometimes be interpreted as representatives of typical data points.

The NMF algorithm we used computes W and H by using alternating optimization and multiplicative updating. Its details, including parameter choices, can be found in the appendix.

We computed a low-rank ($r = 3$) NMF of the raw data. Note that centering the data would not make any sense at all, since then the data would no longer be non-negative.

Then, we considered the 3 columns of W as feature vectors. We found the cities closest to the 3 features to be *Anderson, IN*, *Florence, AL*, *Anniston, AL*, and *Dayton-Springfield, OH*. We wanted to assess if these cities represent the typical American city in some sense.

We found this question to be rather non-obvious since it is hard to say what a "typical" American is like. With this in mind, we decided to take a visual and quantitative approach. Specifically, we produced a plot of the first two principal components of the data. On this plot, we also indicated the three cities closest to the medoids. The big idea was that if these cities are *typical*, then the data should be densely packed around them. On the next page, **Figure 7** shows the resulting plot.

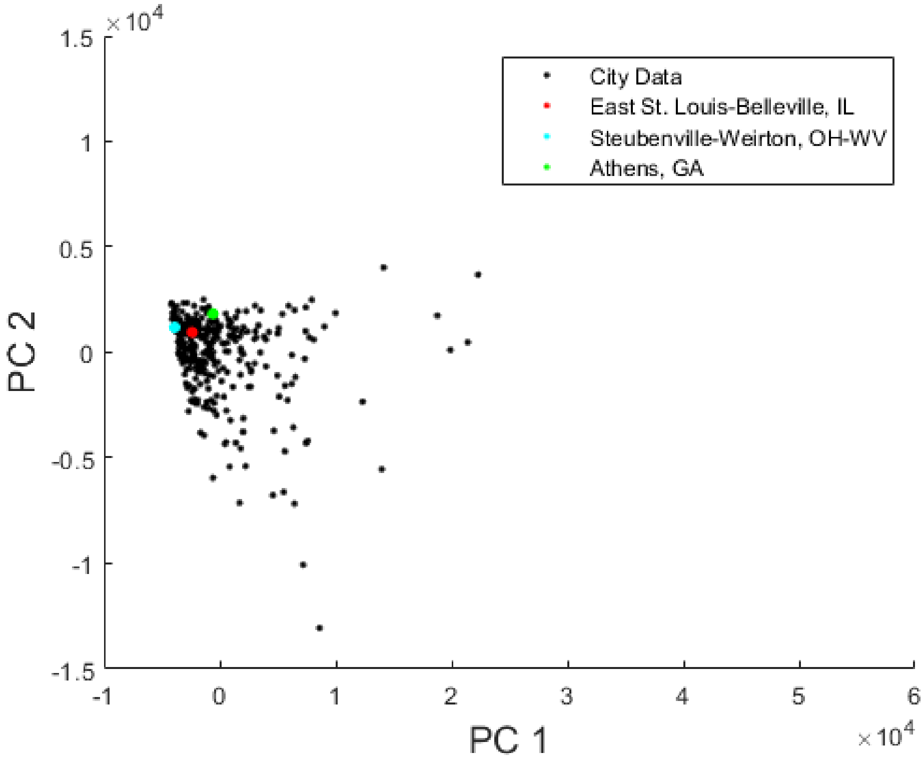


Figure 7: Closest Cities to NMF Features with r=3

Immediately, we notice that all 3 of the representative cities are in the most densely packed region of data points. It makes sense that typical cities will be in densely packed regions, since typicality is essentially a measure of component similarity. We are led to conclude that the 3 chosen cities are fairly representative of the typical American city. It follows that we claim that the feature vectors of the NMF do a good job of representing typical data points.

Self Organizing Maps

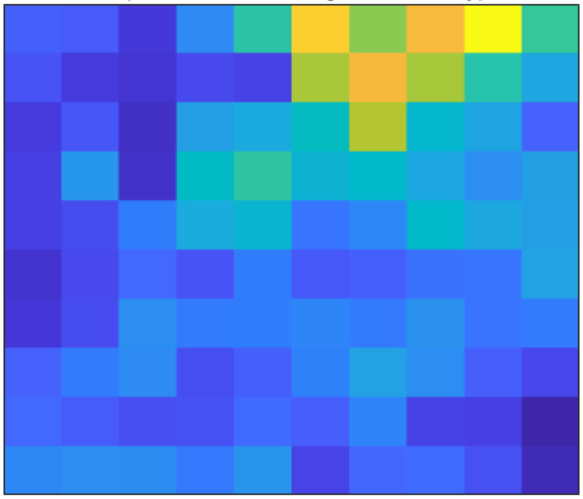
Finally, we considered using a self-organizing map (SOM) to approximate the data in lower dimensions. SOM seeks to approximate data with a low-dimensional manifold. The power of this method is rooted in its ability to approximate data in non-affine subspaces. Methods such as PCA are incapable of this.

If k is the number of dimensions we wish to approximate the data in, then these prototypes have a topology that aligns with the topology of a k-dimensional lattice. Note that a 1-dimensional lattice is essentially just a section of a number line.

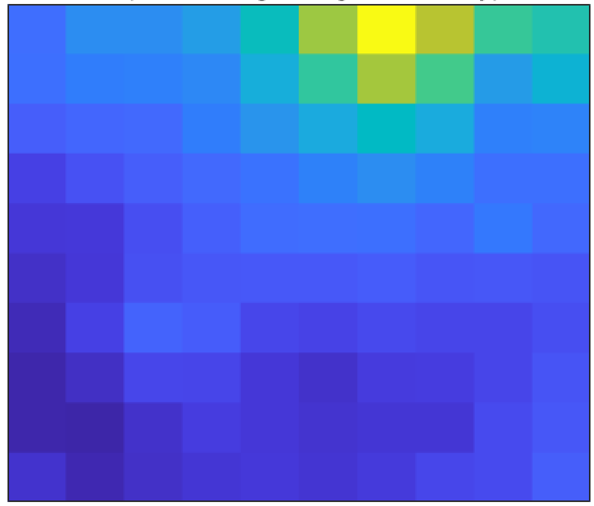
We will attempt to approximate the data with a 2D manifold. The main idea here is that if the data can be well approximated with a 2D manifold, similar cities (in the quality of life sense) should be close to one-another on the corresponding grid (2D-lattice).

First, we decided to approximate the data with 100 prototypes. Since we are seeking a 2 dimensional approximation, the prototypes correspond to points on a 10-by-10 grid. The most immediate obstacle we faced was that of interpreting the SOM results. As a preliminary step in understanding what the prototype grid represents, we decided to use heatmaps. We looked at heatmaps of each component (corresponding to a rating category) of the prototypes. The following page contains all of these visuals. **Colors that are more yellow correspond to higher scores, whereas dark blue corresponds to lower scores**.

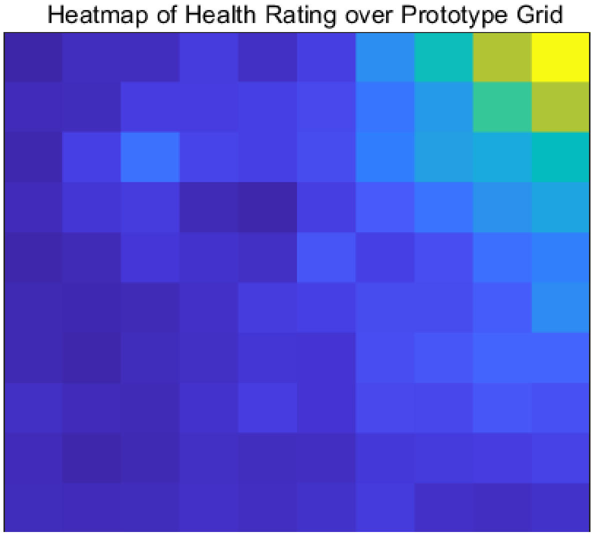
Heatmap of Climate Rating over Prototype Grid



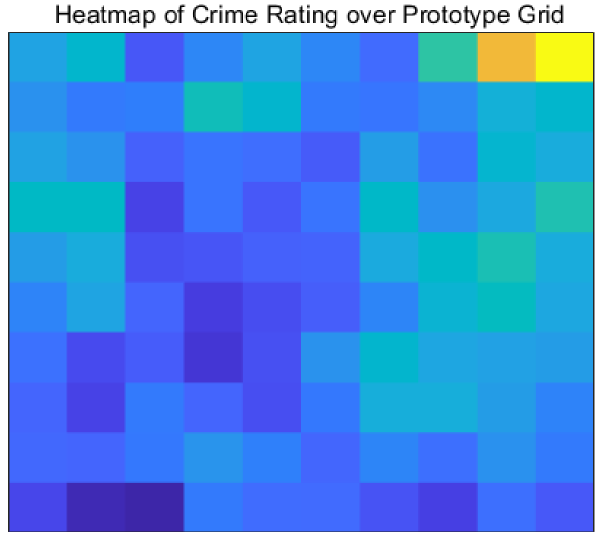
Heatmap of Housing Rating over Prototype Grid



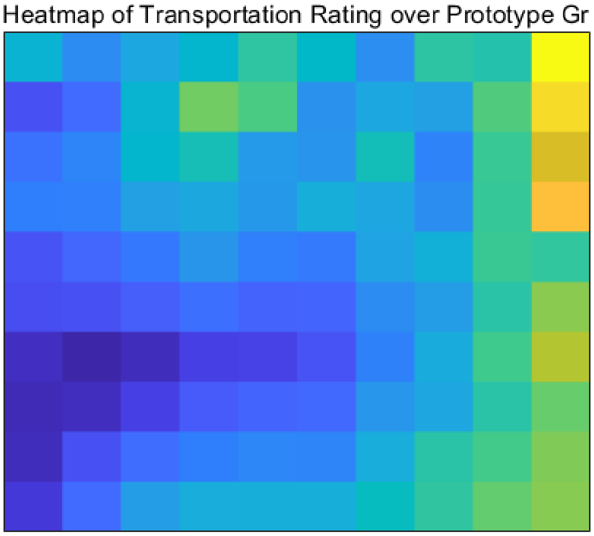
Heatmap of Health Rating over Prototype Grid



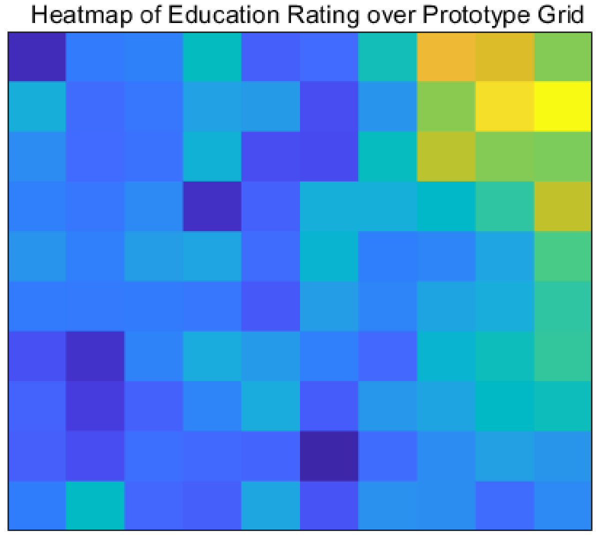
Heatmap of Crime Rating over Prototype Grid

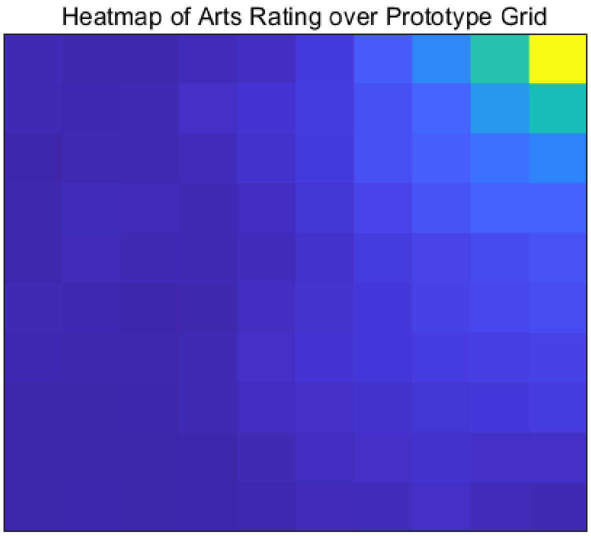


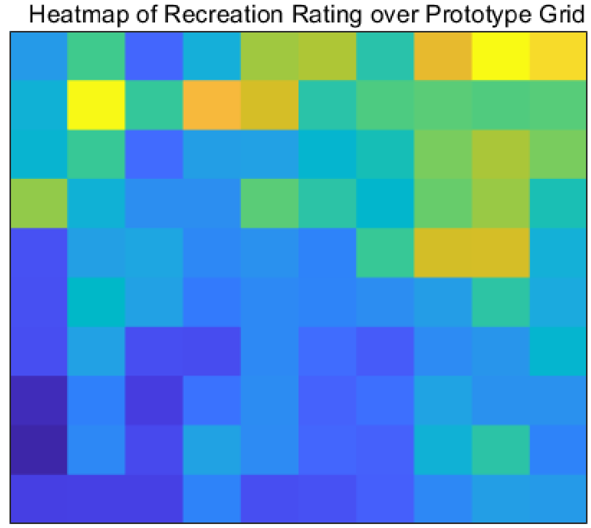
Heatmap of Transportation Rating over Prototype Gr

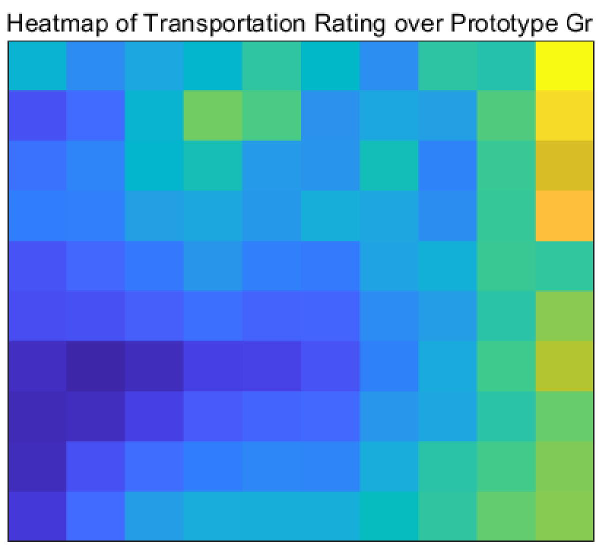


Heatmap of Education Rating over Prototype Grid









Overall, the heatmaps show a strong organization based on ratings. Most of the heatmaps don't have drastic color changes but rather exhibit a gradient between yellow (high scores) and blue (low scores) across the grid. Moreover, except for in the heatmaps corresponding to economics and transportation, the top-right seems to correspond to high scores while the bottom-left corresponds to low scores.

Next, we looked at a heatmap of prototype rating sums. For a given prototype, the rating sum is the sum of all its components. This can be considered a crude way of quantifying the overall quality of life. If the prototypes are organized accordingly to quality of life, the rating sums heatmap should exhibit gradual color changes. The resulting heatmap can be seen in **Figure 8** on the next page.

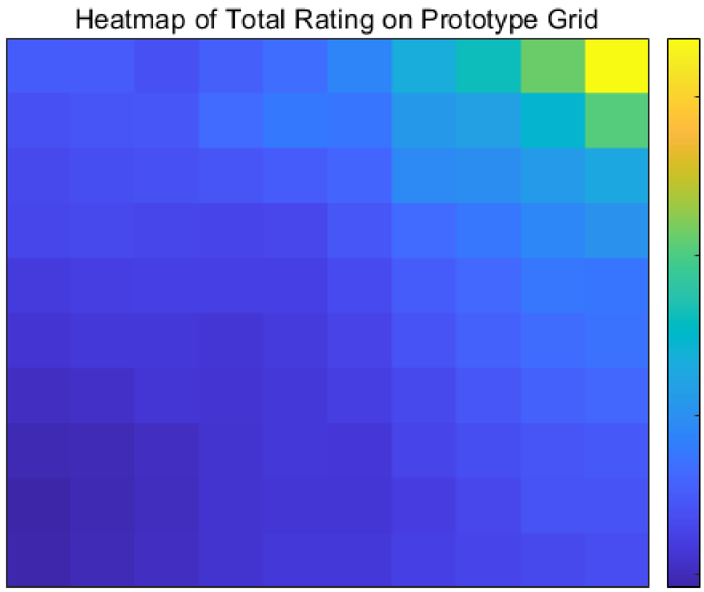


Figure 8: Heatmap of Prototype Rating Sums on 10-by-10 grid

We find the results in this heatmap to be rather amazing! The gradient between the extremes is abundantly clear and even somewhat symmetric about the diagonal connecting the top-right and bottom-left corner. The sum of the scores can be though of as a crude method of quantifying to the overall quality of life in a city. With this in mind, **Figure 8** strongly supports that our prototype grid is organized according to quality of life.

Before we call it a day, however, we need to verify that the prototypes meaningfully span the data. That is, the impressive grid organization above is meaningless is regions of the grid are largely unoccupied by data. To quickly verify the coverage of the prototypes, we produced a button plot placing data points in their best-matching prototype. This plot is shown in **Figure 9**.

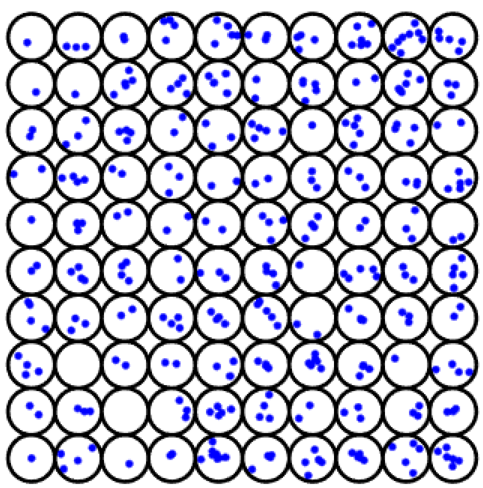


Figure 9: Prototypes on Grid and Data Points Corresponding to them

It's immediately apparent that the entirety of the grid is well-populated with prototypes. This leads us to conclude that the 2D manifold approximates the data both well and meaningfully! That is, the grid is organized according to overall quality of life and the data distributes well over the grid. Cities that are similar in the quality of life sense are close to one another on the grid.

We also computed an SOM with 36 prototypes (6-by-6 grid). Performing a similar assessment, we looked at the heatmaps (for organization) and button plot (for coverage). The rating sums heatmap can be seen in **Figure 10** and the button plot can be seen in **Figure 11**.

Because of the assignment description, we interpreted the SOM results in one more way. For each prototype, we found the best matching data point. Then, we displayed the city names on a grid corresponding to the prototypes. The resulting graphic can be seen in **Figure 10**

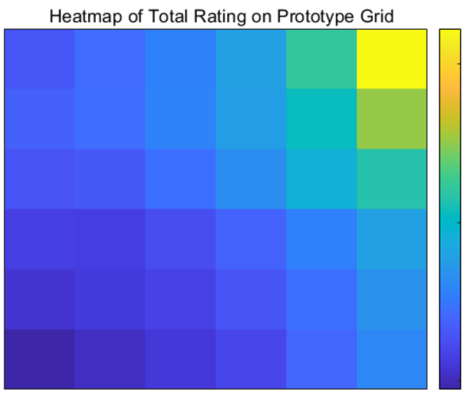


Figure 10: Rating Sums on 6-by-6 grid

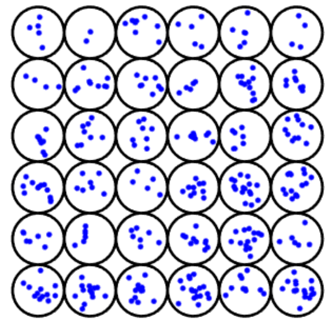


Figure 11: Button plot on 6-by-6 grid

Examining these two visuals, we see very similar results to what we achieved with 100 prototypes. Namely, we see a nice gradient from yellow (top right) to blue (bottom left) and ample coverage of the prototypes by the data.

We also decided to interpret the SOM results in another way. For each prototype, we compute the best matching data point. Then, we plotted the names of those cities on a grid. We first did this interpretation for the 100-prototype SOM. The graphic is a bit cumbersome, but can be seen in **Figure 12**. The size of text does not indicate anything in particular.

Bloomington	Fargo	Columbia	Lansing	Memphis	Atlanta	Baltimore	Cleveland	Boston	New York
Des Moines	Bangor	Tallahassee	Nashville	Albuquerque	Indianapolis	Buffalo	Minneapolis	Newark	San Francisco
Topeka	Fort Wayne	Knoxville	Tulsa	Phoenix	Columbus	Providence	Ann Arbor	Bridgeport	Bergen
Lynchburg	Alton	Wichita	Lubbock	Wilmington	Akron	Portland	Miami	Oakland	Norwalk
Lynchburg	Monroe	Springfield	Vineland	Grand Rapids	Kenosha	Sacramento	Trenton	Santa Cruz	Anaheim
Elkhart	Sioux City	Evansville	Evansville	Lawrence	Hamilton	Joliet	Vallejo	Santa Rosa	Santa Cruz
Elkhart	Terre Haute	Pueblo	Janesville	Saginaw	Medford	Chico	Lawrence	Lawrence	Anchorage
Benton Harbor	Lewiston	Lakeland	Wilmington	Davenport	Fall River	Fall River	Bristol	New London	West Palm Beach
Sharon	Clarksville	Biloxi	Fort Walton Beach	Pittsfield	Hagerstown	Lancaster	Poughkeepsie	Sarasota	Lafayette
Texarkana	Alexandria	Waco	Wichita Falls	Kankakee	Yuba City	York	Bremerton	Fort Pierce	Colorado Springs

Figure 12: 10-by-10 Prototype Grid with Best Matching Cities

There are a few notable features of this graphic. Firstly, we see that some cities are the best match for multiple prototypes. This isn't necessarily concerning, as the resolution of the grid is rather modest.

Secondly, we see that the grid does seem somewhat reasonable. For instance, the top right has similar (and well known) cities like New York, Boston, Philadelphia, and Baltimore. The presence of these cities in the top right, where arts and total score are so concentrated, indicates that this grid does logically match up to the heatmap!

Next, we considered this interpretation for a 36 prototype grid. The results are seen in **Figure 13**.

Sarasota	Santa Rosa	Anaheim	Bergen	Boston	Chicago
New London	Fort Lauderdale	Trenton	Middlesex	Minneapolis	Cleveland
Binghampton	Bloomington	Kalamazoo	Sacramento	Columbus	St. Louis
Brazoria	Davenport	Lynchburg	Bangor	Lansing	Kansas City
Fort Walton Beach	Elmira	Lima	Vineland	Lubbock	Wilmington
Wichita Falls	Alexandria	Lewiston	Evansville	Las Cruces	Lexington

Figure 13: 6-by-6 Prototype Grid with Best Matching Cities

The results are fairly similar to the 100 prototype grid. The organization is the same, and the corners contain many of the same cities. Again, we can tell heuristically that the names-grid matches up well with the heatmap.

As with the heatmap interpretation, these results support that SOM is doing a good job of organizing cities by quality of life. That being said, we find the heatmap argument to be more convincing since it is more quantitatively based. Furthermore, we are not familiar with many of the listed cities and as such cannot gauge similarity between some pairs. Overall, we conclude that the quality of life (in the context of this data) across American cities can be well approximated by a 2D manifold. Moreover, SOMS with 100 and 36 prototypes are both effective.

Conclusion

With PCA, we were able to discern that the arts and housing vary greatly across the country while there's a sense of consistency when it comes to climate, education, crime, and recreation. Through k-medoids, we found that the data does lend itself to a natural clustering into 3 or 4 groups. With the power of NMF, we were able to identify "typical" American cities. Finally, SOM let us approximate the quality of life amazingly well with a 2D manifold. This data set had many hidden insights on life across American cities.

Appendix

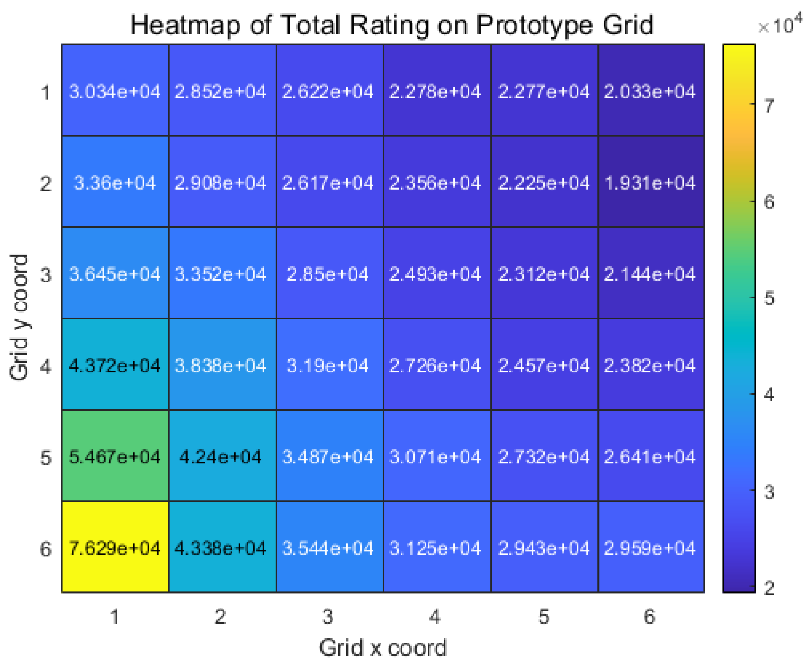
As stated before, please feel free to visit the public GitHub Repository.

1 Previous Heatmap Issue

This section pertains to the mismatch between the city-names grid and heatmap on the previous submission of this project. Essentially, the mismatch was caused by 2 separate factors.

First, a typo caused our city-names grid to swap the x and y coordinates. Fixing the typo resolved this issue.

The next issue was that heatmap labels the axes rather unusually. That is, the y axis coordinates decrease as they go up. **Figure 14** below shows what I mean.



So, putting together the effects of both issues, we see that my names-grid was off from the heat map by a reflection of the x-coordinates. Recalling what the heatmap and names-grid looked like, this makes sense. That is, Chicago would go from the top left corner to the top right corner, where arts have a high rating. All is right with the world!

2 Potential Normalization

As discussed in the introduction, we considered normalizing the data. Particularly, we considered scaling the rows so that every row added up to one. Essentially, this would be a way of equalizing the categories' influence on the analysis, more or less. Ultimately, we chose not to do this for our official analysis because we concluded it unwise to mess with data's scaling without knowing much about the data's origins.

That being said, we did run sum tests on this normalized version of the data, and the results were rather similar to the raw data. Most notably, the total-sum heatmaps had a very similar structure to those computed with the raw data! This is significant since the total sum is highly affected by this scaling. **Figures 14 and 15** show this result. While the yellow region is in a new spot now, the structure is very clearly similar to before.

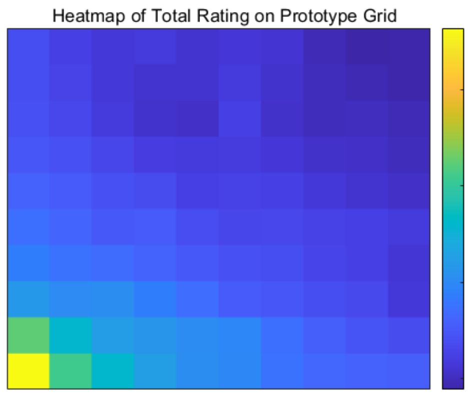


Figure 14: Normalized Rating Sums on 10-by-10 grid

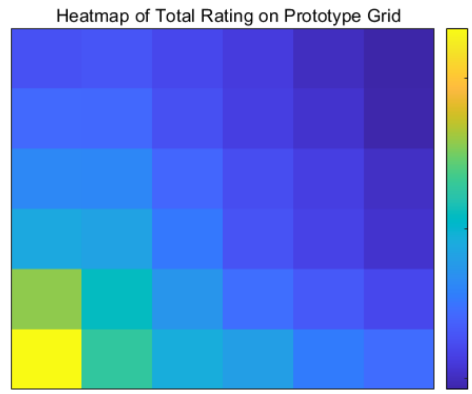


Figure 15: Normalized Rating Sums on 6-by-6 grid

3 Parameters

For K-medoids, we used k =2,3, and 4. We used $\tau = 0.001$.

For NMF, we used $r = 3$ and $\tau = 0.001$.

For SOM, we used $k = 100$ and 36. The other parameters were computed by the defaultParams function.

4 K-medoids with k=5

Figure 16 shows the results of clustering with $k = 5$. Seeing as the results are not good, we decided not to try $k > 5$. The red group is almost nonexistent.

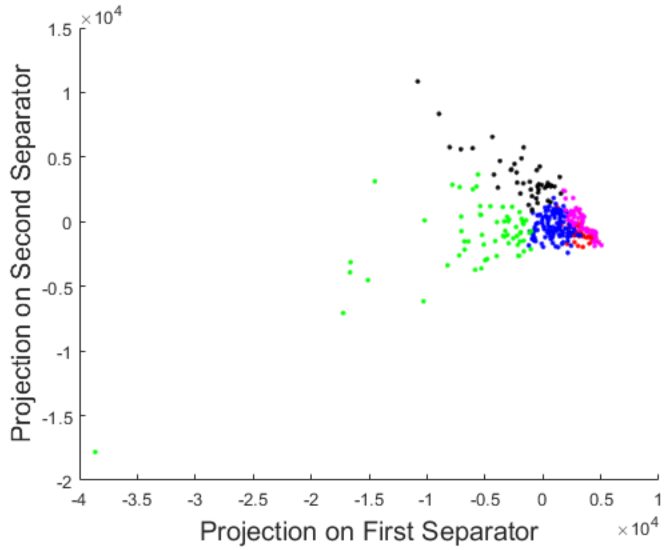


Figure 16: K-Medoids result with k=5

5 Code - NMF

```
    function [W, H] = NMF(X, r, tol)
% X approx= WH

[n,p] = size(X);

t = 0;

Wo = rand(n,r);
Ho = rand(r,p);
W = Wo;
H = Ho;

while t==0 || (norm(W - Wo, 'fro')/norm(Wo, 'fro') + norm(H - Ho, 'fro')/norm(Ho, 'fro') > tol)
    %Move forward!
    Wo = W;
    Ho = H;

    %Wo is W^(t), same for Ho
    %W is W^(t+1), same for H

    %Update W
    Xc = Wo * Ho; %approximation of X
    W = ((X*Ho') ./  (Xc * Ho')) .* Wo;

    Xc = W * Ho; %approximation of X
    H = ((W'*X) ./ (W'*Xc)) .* Ho;

    t=t+1;
end
```

```
end
```

6 Code - Analysis

```
    %Levent Batakci
%3/28/2021
%Math444 Midterm Project
%Load the data
load cities.mat

%Format the data
ratings = ratings';
[n, p] = size(ratings);
%ratings = ratings ./ sum(ratings,2);


%Extract the categories
categories = string(categories)';

%Get the states
states = char(p, 10);
for i = 1:p
    name = names(i,:);
    nonEmpty = find(name ~= ' ');
    stopChar = nonEmpty(end);
    while stopChar > 0 && name(stopChar) ~= ' '
        stopChar = stopChar-1;
    end
    states(i,1:nonEmpty(end)-stopChar) = name(stopChar+1:nonEmpty(end));
end
%This section manipulates the data so that individual scores are more
%representative of how well a city does in an area as compared to other
%cities

relativeRatings = zeros(size(ratings));
for c = 1:n
    [~, I] = sort(ratings(c,:));
    for i = 1:p
        relativeRatings(c,I(i)) = i;
    end
end
%relativeRatings = relativeRatings - sum(relativeRatings,2)/p;
relativeRatings = ratings ./ sum(ratings,2);
%Show medians
medians = median(sum(ratings,2),2);
[sorted, sortIndex] = sort(medians)
bar(sorted)
ylabel("Category Median", "FontSize",15)
set(gca,'xticklabel', categories(sortIndex), 'FontSize', 12)
xtickangle(-45)
[Z, Ur] = PCA_r(ratings, 2); %Compute the PCA
```

```
%First two feature vectors
u1 = Ur(:,1);
u2 = Ur(:,2);
scatter(Z(1,:),Z(2,:))
%Compare the impact of attributes on the feature vectors
figure(1)
bar(abs(u1))%Bar graph of u1's absolute components
ylabel("Relative Influence", "FontSize",15)
set(gca,'xticklabel',categories, 'FontSize', 12)
xtickangle(-45)
sgtitle("Component Influences of u1", "Fontsize", 15)

figure(2)
bar(abs(u2)) %Bar graph of u2's absolute components
ylabel("Relative Influence", "FontSize",15)
set(gca,'xticklabel',categories, 'FontSize', 12)
xtickangle(-45)
sgtitle("Component Influences of u2", "Fontsize", 15)

%Kmedoids
D = dMatrix(ratings, @norm2);
K=[2, 3, 4, 5];
tau = 0.001;

%Define parameters
k = ;
N = k^(1/2);
params = defaultParams(k, 2);

%Compute the SOM
ratings_scaled = ratings ./ sum(ratings,2);
M = SOM(ratings_scaled, k, params);

%Lets look at some heatmaps to understand the grid better
cats = ["Climate", "Housing", "Health", "Crime", "Transportation", "Education", "Arts", "Recreation",

for cat = 1:n
    C = zeros(N,N);

    for i = 1:k
        c = getCoord(i, N);
        x = c(1);
        y = c(2);
        C(x,y) = M(cat, i);
    end

    figure(30+cat);
    h = heatmap(1:N, 1:N, C, 'FontSize', 0.01, 'GridVisible',"off");
    sgtitle("Heatmap of " + cats(cat) + " Rating over Prototype Grid");
    h.Colormap = parula;
end
```

```
C = zeros(N,N);

for i = 1:k
    c = getCoord(i, N);
    x = c(1);
    y = c(2);
    C(x,y) = sum(M(:, i));
end

figure(30+n+1);
h = heatmap(1:N, 1:N, C, 'FontSize', 0.01, 'GridVisible',"off");
sgtitle("Heatmap of Total Rating on Prototype Grid");
h.Colormap = parula;


for i = 1:p
    x = ratings(:,i);
    [~, bmu_I] = min(vecnorm(M - x));
    c = getCoord(bmu_I, N);

    %Place a dot
    theta = RandRange(2*pi,0,1);
    r = RandRange(0.4,0,1);
    x_ = r*cos(theta) + c(1);
    y_ = r*sin(theta) + c(2);
    txt = states(i,:);
    %text(x_,y_,txt, 'FontSize', 10)
    scatter(x_, y_, 150, 'b.');
    hold on
end

%%SET UP CIRCLES!%%
cirlces = zeros(N,N);
q1 = [1:N]'*ones(1,N);
q2 = ones(N,1)*[1:N];
Q = [q1(:) q2(:)];
% Define the distance squared matrix
D2 = zeros(k,k);
for i = 1:k
    for j = 1:i
        D2(i,j) = norm(Q(i,:) - Q(j,:))^2;
        D2(j,i) = D2(i,j);
    end
end

% Plotting the buttons on the map
thplot = linspace(0,2*pi,k);
cc = cos(thplot);
ss = sin(thplot);
for j = 1:k
    plot(Q(j,1)*ones(1,k) + 0.5*cc,Q(j,2)*ones(1,k)+0.5*ss,'k-','LineWidth',2)
    hold on
end
```

```
axis('square')
axis([0,N+1,0,N+1])
set(gca,'FontSize',20)
set(gca, 'visible', 'off')
hold off
%%%%%

%City names interpretation
figure(100)
clf
for i = 1:k
    m = M(:,i);
    [~, bmu_I] = min(vecnorm(ratings - m));
    c = getCoord(i, N);

    %Place a dot
    theta = RandRange(2*pi,0,1);
    r = RandRange(0.4,0,1);
    x_ = c(1)-1;
    y_ = c(2)-0.5;
    name = names(bmu_I,:);
    commas = find(name == ',' | name == '-');
    ind = commas(1)-1;
    name = name(1:ind);
    txt = name;
    fs = 30;
%     if(length(name) <= 8)
%         fs = 30;
%     end
    text(x_,y_,txt, 'FontSize', fs, 'Rotation', 0)
    %scatter(x_, y_, 150, 'b.');
    hold on
end
set(gca,"visible","off")

%Plot squares
for i = 1:N+1
    X = [i-1 i-1];
    Y = [0 N];
    plot(X, Y, 'k');
    plot(Y, X, 'k');
end
```