CSE6242 Fall 2022 Team 32 Final Report

Harry Li, Di You, Lilin Hu, Sai Wang, Ling Jing, Xiaofei Chen

Introduction and Problem Definition

Choosing a city to live can be a big life decision to many. With exploding information in this digital age, recommender systems to help people with such decisions seem to be lacking. To the best of our knowledge, there are only a few websites/projects that recommend cities (see Appendix). Among them, two recommend cities to visit using collaborative filtering (CF) methods, one assesses the similarity between urban economies by calculating whether they are competitive in the same industries, and the other four use content-based methods to recommend cities to live/visit. We think the four projects that recommend cities to live are not good enough because 1) the city range is small (the largest has 672 US cities). 2) A content-based model can only be as good as the hand engineered features of the cities, but the city features used in these models leave much to be desired. For example, the characteristics of a city when people are considering for travel are certainly different from the ones when they are considering for moving. However, we see some of the existing projects did not make a distinction about that. 3) We think an interactive map is key to success given the geographic nature of this project. However, none of the websites/projects has an interactive user interface.

Here we propose to build an interactive website to recommend cities to live in the US. A user can give a city as input, and then the website will return a list of cities most similar to the input city. We envision our project will be useful for people who want to move to another city but only have experience in the current/previous cities (know input city => explore output cities). It can also be helpful for people who know very little about certain cities but would like to know how they compare with cities they do know (do not know input city => relate to cities more familiar with). Last but not least, it can help businesses to explore potential markets.

Literature Survey

Bidart et al., 2014; Takerngsaksiri et al., 2019 *(a) main idea:* finding similar cities or areas in different cities using Tripadvisor.com network (CF) or Twitter data (content-based). *(b) why useful:* similar projects. *(c) shortcoming:* CF-based methods are better suited for recommending cities to travel, not to live, and the content-based method was limited to Twitter data.

Houle et al., 2020; Sohangir et al., 2017; Xia et al., 2015 *(a) main idea:* alternative approaches to the mainstream cosine similarity measures in order to overcome the curse of dimensionality. *(b) why useful:* As our data can be high-dimensional, we may resort to these alternative methods. *(c) shortcoming:* No consensus on the effectiveness of these methods.

Adomavicius et al., 2005 *(a) main idea:* describes common recommendation approaches such as content-based, CF, and hybrid approaches, and various limitations of each method. *(b) why useful:* helps understand the basic algorithms of common recommendation systems. *(c) shortcoming:* CF methods suffer from data sparsity and new user problems. Content-based methods suffer from overspecialization.

Gulzar et al., 2018 *(a) main idea:* a course recommender using hybrid algorithm by combining the predictions of CF and content-based methods to increase the overall precision. *(b) why useful:* We may consult their approach to solve data sparsity problem by clustering the cities to reduce categories. *(c) shortcoming:* Hybrid approach requires high computational complexity and a large database.

Jin et al., 2012 *(a) main idea:* While the similarities of users can be captured, their rating patterns may not be the same. This paper designed a decoupled model and a preference model to distinguish between user preferences and ratings. *(b) why useful:* The preference model models the orderings of items preferred by a user, rather than the user's numerical ratings of items. This idea can be applied to

our similarity calculation. *(c) shortcoming:* The accuracy of the preference model needs improvement which suggests that the rating information cannot be ignored completely.
Li et al., 2020; Poston et al., 2009; Zhang et al., 2019; Kang et al., 2020 *(a) main idea:* factors that motivate people to move in the US or to attract overseas immigrants, such as the human development index, education grade, climate, cost of living index, house price index, crime rate, tax rates/credits, etc. *(b) why useful:* guidance for us to select important attributes for our city recommendation project. *(c) shortcoming:* difficulties in finding enough data at the city level.
Stephens et al., 2013; Wang et al., 2022 *(a) main idea:* analyzed travel distance and time for US hemodialysis patients and supercommuters in Bay area. *(b) why useful:* method to generate additional attributes from public data. *(c) shortcoming:* narrow scope (only on dialysis patients, only in Bay area).
Naylor et al., 2019 *(a) main idea:* uses geospatial techniques to quantify healthcare accessibility within the US using provider, Medicare, and Census data. *(b) why useful:* analysis to designate accessibility of healthcare due to travel time. *(c) shortcoming:* simply looks at spatial accessibility and does not factor in affordability and acceptability of healthcare.
Lan et al., 2021 *(a) main idea:* a web mapping system that allow users to experiment with different schemes in the geographic analysis and provide users analytical interactivity by integrating maps with geodemographic charts. *(b) why useful:* method to integrate the mapping and statistical data. *(c) shortcoming:* The analysis tool is not flexible and the data update is time consuming.
MacEachren et al., 1998; Lu et al., 2017 *(a) main idea:* geographic visualization techniques. *(b) why useful:* design of the interactive map. *(c) shortcoming:* The option of statistical variables is fixed and undivided. The temporal navigation is complicated for users to understand.

Methodology

Since there are no platforms that track where people have lived like tripadvisor.com tracks where people have visited, we think a content-based method is more appropriate for recommending cities to live. The project is split into three parts: 1) data collection: scrape, combine, and clean city attributes data needed for computation and visualization; 2) computation: develop algorithms to compute similarities between cities; 3) visualization: build a user-friendly interactive interface to visualize and dynamically update recommendations. As stated in *Introduction and Problem Definition*, current city recommenders are not only lacking but also flawed. Our project aims to improve with the following **innovations**:
1. Data: cover much larger city range than any existing websites/projects and combine diverse data into one dataset which includes a wide range of attributes. The final dataset has 27964 cities x 75 attributes.
2. Computation: group attributes into categories and assign weights to categories based on users' input.
3. Visualization: develop an interactive user interface for direct visualization of the recommendations.

Data Collection

We expanded our city range to 27964 cities with demographic and socio-economic attributes scraped from datausa.io supplemented with weather data for 5940 cities from usclimatedata.com. To impute the missing weather data, we applied k-nearest neighbor (KNN) regression using longitude and latitude information scraped from openweathermap.org as the feature space and Euclidean distance as the distance metric. We set k=1, essentially imputing the missing weather data for a city as its geographically closest city's values. For the rest of the attributes no more than 10% of the data was missing, and we filled the missing values with the median of the state the city is in. Then we scaled the data with a min-max scaler (min = 0, max = 1).

Computation

The algorithm we used to recommend cities most similar to an input city is weighted cosine similarity, which is built upon vanilla cosine similarity. Cities with larger cosine similarity values are more similar to the input city. Mathematically, if the vector of the input city is x, the vector of the city to be compared is y, then the cosine similarity value is:

$$\text{Cosine Similarity (x, y)} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

To calculate the similarity score more accurately reflecting the user's interest, we assign weights to each city attribute based on the user's input. As we have a large range of attributes, we group them into categories (e.g., economy, weather, living & housing, etc.) and give the user options to rank the importance of each category of attributes. Then the weighted cosine similarity can be calculated:

$$\text{Weighted Cosine Similarity (x, y, w)} = \frac{\sum_{i=1}^{n} w_i x_i y_i}{\sqrt{\sum_{i=1}^{n} w_i x_i^2}\sqrt{\sum_{i=1}^{n} w_i y_i^2}}$$

After the weighted cosine similarity values are computed, we order them in descending order.

Visualization

Our project has an interactive user interface primarily built with HTML, CSS, and JavaScript. The front end was developed using React, Bootstrap, and JavaScript, and utilized the MapBox API and D3.js for visualization. React allows for the establishment of hooks and re-rendering of different portions of the website, while Bootstrap provides a framework for the layout of the website. The backend was developed using Express.js with parameters being passed to query the MySQL database. These parameters produce a pre-calculated dataset for the score which is then converted into GeoJSON before passing the corresponding values to the front end through the properties attribute. This reduces the number of fetches as loading the initial dataset produces the most delay for the user. These values are then re-calculated real time based on the current weights provided on the input slider. Search was implemented by utilizing the Boolean

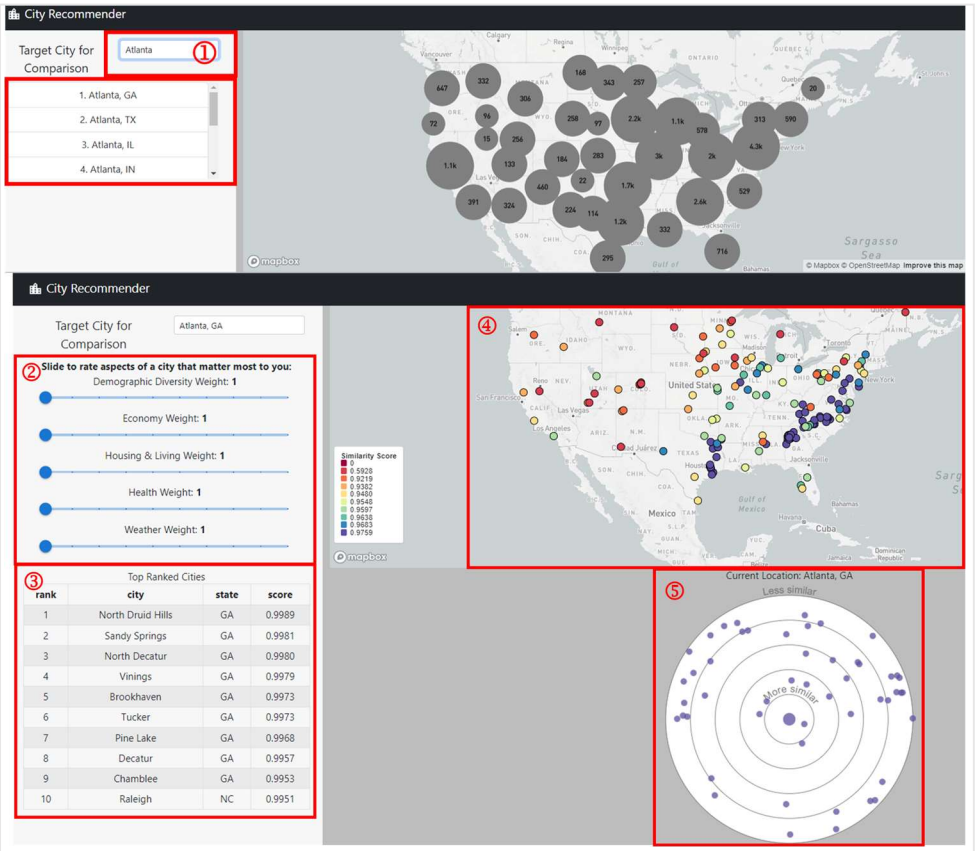


Figure 1. Screenshots of the website interface.

Full-Text Search implementation in MySQL using the columns of city name, full state name, and abbreviated state name. The results from this query are loaded in descending order and the top 10 results are returned to the front end. For hosting, the front end utilizes Netlify, backend uses Azure App services, and the Database is hosted on Amazon Web Services RDS.

A demonstration of our website interface is shown in **Figure 1**. The user can search from a search box to select the input city **(Figure 1, Box 1)**. The website also includes a slider for the user to rank categories of city attributes based on their importance **(Figure 1, Box 2)**. Once the inputs from the user are received, similarity scores for all cities will be calculated and the color attributes of the cities on the map will transition to new values reflecting the calculated similarity scores. To avoid clutter, we only display the top 50 and bottom 50 similar cities as well as a random set of 100 cities not in the top or bottom cities **(Figure 1, Box 4)**. To better showcase the most similar cities, we include a ring chart of the top 50 similar cities with the input city at the center of the circle and the rest of the cities on concentric circles with different radii representing their similarity scores to the input city. The higher the similarity score, the smaller the radius **(Figure 1, Box 5)**. The tooltip for each city displays the city name and state name, as well as the similarity score and rank (after the input city has been chosen). Last but not least, we display a list of the top 10 most similar cities with their similarity scores and ranks **(Figure 1, Box 3)**.

Experiments and Evaluation

Evaluation of the imputed weather data

Since we only acquired ~20% of the weather data and imputed the rest 80%, we performed a clustering analysis to evaluate the usefulness of the weather data after the imputation. For this purpose, we partitioned all cities into five clusters using k-means clustering with all the weather attributes. As shown in **Figure 2**, our five clusters (left) closely resemble the pattern of the five US climate types (right; marine, cold/very cold, hot-dry/mixed-dry, hot-humid, and mixed-humid), giving us more confidence on the imputed data.

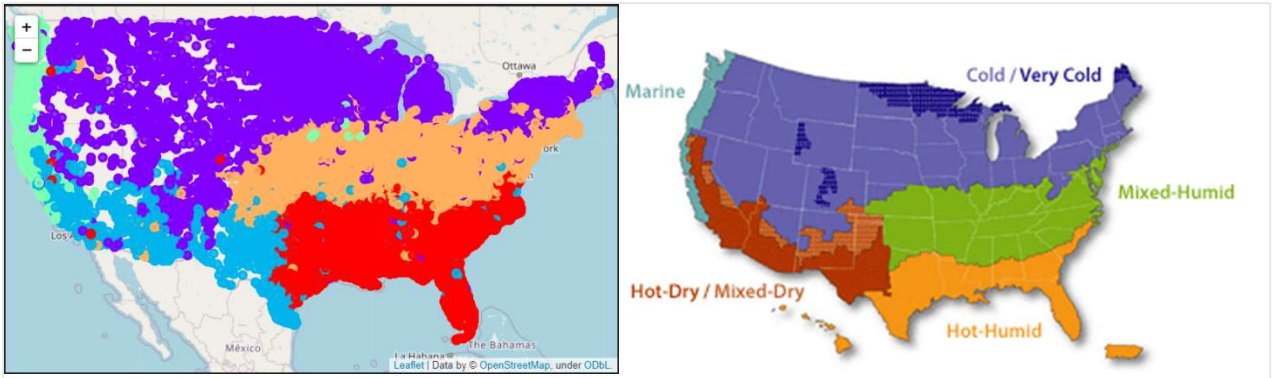


Figure 2. (left) k-means clustering of all cities using weather attributes (k=5). (right) US climate zones from energy.gov.

Experiment with an alternative similarity measure

As mentioned in *Literature Survey*, the mainstream cosine similarity measure is based on L2 norm (Euclidean distance) and may suffer from the curse of dimensionality, a phenomenon where the concept of proximity becomes less meaningful as dimensionality increases, e.g., the ratio between the nearest and farthest points approaches 1 as dimensionality approaches infinity. Some research proposed to use L1 norm or even fractional norm in a high dimensional data situation to alleviate this problem (Aggarwal et al., 2001). However, this is an active research area and a consensus has not been

reached (Mirkes et al., 2020). Nevertheless, we tested an alternative similarity measure and compared with the mainstream cosine similarity measure.

As our dataset is large, calculating similarity matrices is computationally taxing. For this experiment we randomly sampled 100 cities from the 27964 cities and calculated the cosine similarity matrix. As shown in **Figure 3**, the similarity scores range from ~0.7 to 1 and the distribution is left skewed, with most scores in the 0.9-1.0 range. This means that the cosine similarity measure provided a poor contrast. Thus, we sought alternative similarity measures using a lower norm metric, specifically the improved sqrt-cosine similarity (ISC) proposed by Sohangir et al. The ISC similarity is defined as

$$\mathrm{ISC}(\mathrm{x},\mathrm{y}) = \frac{\sum_{i=1}^{n}\sqrt{x_i y_i}}{\sqrt{\sum_{i=1}^{n} x_i}\sqrt{\sum_{i=1}^{n} y_i}}$$

Because it is based on Hellinger distance (L1 norm), it is thought to be more favorable than cosine similarity (L2 norm) for high-dimensional applications. However, when we calculated ISC similarity scores for the same randomly chosen 100 cities, the distribution of the scores showed an even poorer contrast, with a range of 0.825-1.0 and most scores in the 0.95-1.0 range (**Figure 4**). Therefore, we did not find any merit in using ISC similarity and still used weighted cosine similarity in building our city recommender website.

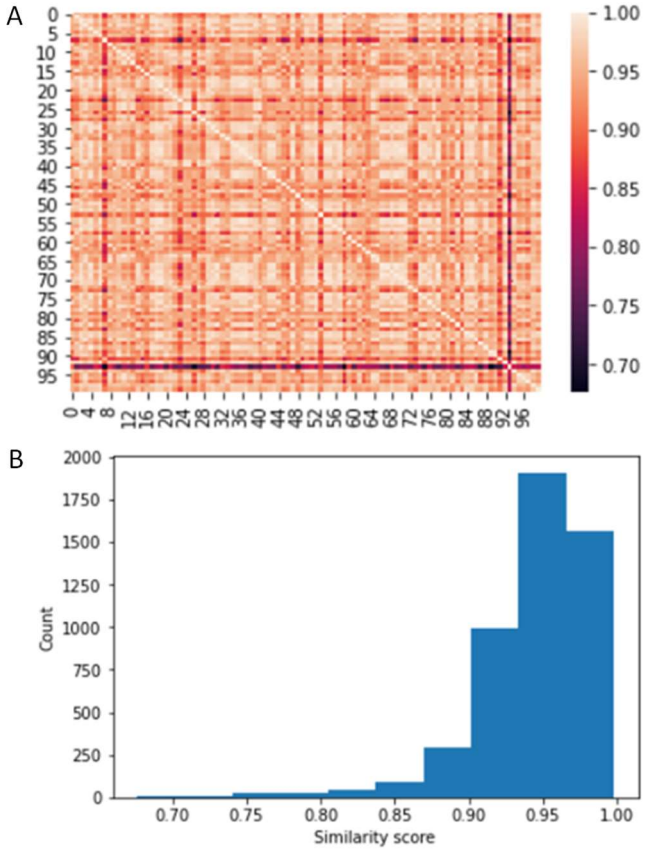


Figure 3. (A) Heatmap of the cosine similarity matrix of the randomly chosen 100 cities. (B) Histogram of the pairwise cosine similarity scores for the same 100 cities.

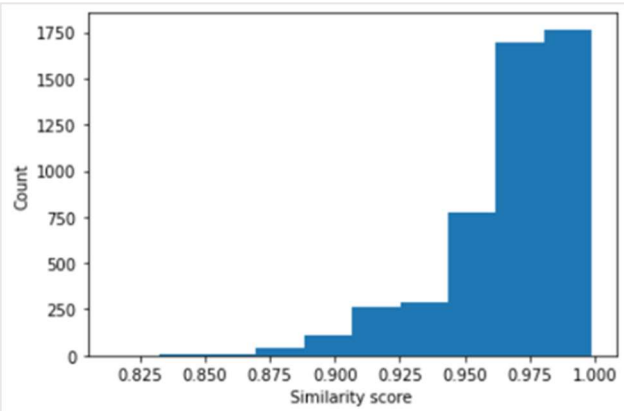
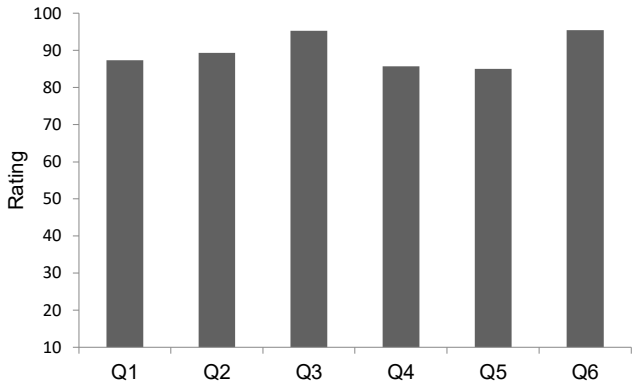


Figure 4. Histogram of the pairwise ISC similarity scores for the same 100 cities used in Figure 3.

User study

We designed a questionnaire using Google Form to collect feedback from users after they explore our interface. The questionnaire has 6-7 questions asking users to rate their experience with our website and should usually take less than 2 minutes to complete. We collected 30 responses, most of which showed favorable results toward our website (**Figure 5**). For example, over 95% of the users thought our recommended cities were reasonable (Q3), and our website was better than another city recommender (Q6). All other metrics were averaged between 8 and 9 on a 1-10 scale. We also got valuable feedbacks from the optional question asking for areas to improve. Many of them suggested providing more explanations to describe what each component of the layout does. We agree

with this suggestion and have added some instructions to guide the user. However, due to the time limit, there is still large room for improvement in this area. Other suggestions include doing a worldwide version and increasing the loading speed. While we agree these are good suggestions, we think there is not much we can do at this point given the large dataset requirement of the project. Finally, there is a suggestion that it would be better to have a tier list of all similar cities instead of top 10 cities. We think the filter functions discussed below in *Discussion and Conclusions* could partially address this need.



Q1 - How likely are you to recommend our website to a friend or colleague?
Q2 - How satisfied were you with your experience on our website?
Q3 - Do you think our recommended cities based on your input are reasonable?
Q4 - How satisfied were you with our displaying of recommended cities?
Q5 - How likely will you return to our website?
Q6 - Compared with the other city recommender website (https://cityrecommender.herokuapp.com/), do you think our website is better?

Figure 5. User study questionnaire results

Discussion and Conclusions

To summarize, in this project we developed an interactive website to recommend cities to live in the US with a comprehensive and relevant city database. It saves users a lot of time to look over the information about cities all over the places. Compared with other similar projects, an advantage of our website involves the interactive map when displaying the similar/dissimilar cities, in addition to a plain text table. Not only can users get the most similar cities from their input cities, but they can also see on the map the trend on where these cities are located. For example, the cities most similar to Atlanta, GA are primarily located on the coast of Georgia, South Carolina, North Carolina, Virginia, and Texas (**Figure 1** purple dots), and the cities most dissimilar are mostly in Alaska. This result makes complete sense. We tested many input cities and the results all seem reasonable (and the user study proved it, too).

Because our database has almost 28000 cities, it would be too much clutter to display all the cities on the map. Therefore, we utilized Mapbox GL JS clustering to render cities as clusters when the website is first loaded. After an input city is given, only 200 cities are shown including the top 50 and bottom 50 most similar cities. It would be ideal to add elaborate filter functions to enable direct filtering of the results. For example, users can choose to show only cities that have population in certain ranges they desire or exclude Alaskan cities from the results. Additional future endeavors could consider adding more attributes to the city database, for example, venue information from Foursquares' points of interest data platform (https://location.foursquare.com/products/places/) and public school information from Niche (https://www.niche.com/about/data/).

Team member contribution

X.C. wrote the final report and performed the clustering analysis and the similarity algorithm comparison. H.L. recorded the proposal presentation and built the website. S.W. scraped datausa.io and openweathermap.org data. D.Y. and X.C. cleaned the data. L.H. wrote the progress report and conducted the user study. L.J. and X.C. made the final slides. All team members worked on the proposal and the proposal slides.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734-749. https://doi.org/10.1109/TKDE.2005.99

Aggarwal, C.C., Hinneburg, A., Keim, D.A. (2001). On the surprising behavior of distance metrics in high dimensional space. In Van den Bussche, J., Vianu, V. (eds) *Database Theory — ICDT 2001. Notes in Computer Science*, 1973, 420–434. Springer. https://doi.org/10.1007/3-540-44503-X_27

Bidart, R., Pereira, A.C.M., Almeida, J.M., & Lacerda, A. (2014). Where should I go? City recommendation based on user communities. *2014 9th Latin American Web Congress,* 2014, 50-58. https://doi.org/10.1109/LAWeb.2014.15

Gulzar, Z., Leema, A. A., & Deepak, G. (2018). PCRS: Personalized course recommender system based on hybrid approach. *Procedia Computer Science, 125*, 518-524. https://doi.org/10.1016/j.procs.2017.12.067

Houle, M. E., Kriegel, H. P., Kröger, P., Schubert, E., & Zimek, A. (2010). Can shared-neighbor distances defeat the curse of dimensionality? In M. Gertz & B. Ludäscher (Eds.), *Scientific and Statistical Database Management* (SSDBM 2010, pp. 482–500). Springer. https://doi.org/10.1007/978-3-642-13818-8_34

Jin, R., Si, L., & Zhai, C. (2012). Preference-based graphic models for collaborative filtering. *arXiv preprint arXiv:1212.2478*. https://doi.org/10.48550/arXiv.1212.2478

Kang, S., Kim, J., Hong, S., & Ko, D. (2020). Determinants of the intention to recommend overseas migration to a city: The perspective of foreign residents. *The Social Science Journal (Fort Collins), ahead-of-print(ahead-of-print)*, 1–12. https://doi.org/10.1080/03623319.2020.1728511

Lan, T., & Longley, P. (2021). Interactive web mapping of geodemographics through user-specified regionalisations. *Journal of Maps,17*(1), 71-78. https://doi.org/10.1080/17445647.2021.1912667

Li, X., Huang, S., Chen, J., & Chen, Q. (2020). Analysis of the driving factors of U.S. domestic population mobility. *Physica A: Statistical Mechanics and its Applications, 539,* 122984-122996. https://doi.org/10.1016/j.physa.2019.122984

Lu, W., Ai, T., Zhang, X., & He, Y. (2017). An interactive web mapping visualization of urban air quality monitoring data of China. *Atmosphere, 8*(12), 148. https://doi.org/10.3390/atmos8080148

MacEachren, A. M., Boscoe, F. P., Haug, D., & Pickle, L., W. (1998). Geographic visualization: designing manipulable maps for exploring temporally varying georeferenced statistics. *Proceedings IEEE Symposium on Information Visualization*, 1998, 87-94. 10.1109/INFVIS.1998.729563

Mirkes, E.M., Allohibi, J., & Gorban, A. (2020). Fractional norms and quasinorms do not help to overcome the curse of dimensionality. *Entropy, 22*(10), 1105. https://doi.org/10.3390/e22101105

Naylor, K. B., Tootoo, J., Yakusheva, O., Shipman, S. A., Bynum, J. P. W., & Davis, M. A. (2019). Geographic variation in spatial accessibility of U.S. healthcare providers. *PLOS ONE*,*14*(4), e0215016. https://doi.org/10.1371/journal.pone.0215016

Poston, D., Zhang, L., Gotcher, D., & Gu, Y. (2009). The effect of climate on migration: United States, 1995-2000. *Social Science Research, 38*(3),743-753. https://doi.org/10.1016/j.ssresearch.2008.10.003

Sohangir, S., Wang, D. (2017). Improved sqrt-cosine similarity measurement. *Journal of Big Data*,*4*, 25. https://doi.org/10.1186/s40537-017-0083-6

Stephens, J. M., Brotherton,S., Dunning, S. C., Emerson, L. C., Gilbertson D. T., Harrison D. J., Kochevar J. J., McClellan A. C., McClellan W. M., Wan S., &Gitlin M. (2013). Geographic disparities in patient travel for dialysis in the United States. *The Journal of Rural Health,29*(4), 339-348. https://doi.org/10.1111/jrh.12022

Takerngsaksiri, W., Wakamiya, S., &Aramaki, E. (2019). City link: Finding similar areas in two cities using twitter data. *Web and Wireless Geographical Information Systems* (pp. 13–27). Springer. https://doi.org/10.1007/978-3-030-17246-6_2

Wang, B. S., Rodnyansky, S., Comandon, A., & Boarnet, M. (2022). Drive until you qualify: exploring long commutes in a high housing cost region. *SSRN Electronic Journal.* https://doi.org/10.2139/ssrn.4074809

Xia, P., Zhang, L., & Li, F. (2015). Learning similarity with cosine similarity ensemble. *Information Sciences, 307*, 39–52. https://doi.org/10.1016/j.ins.2015.02.024

Zhang, Y., & Hewings, G. (2019). Nonlinear tax-included migration: an overlooked tale. *The Annals of Regional Science, 63*(3)*, 425-438.* https://doi.org/10.1007/s00168-019-00902-5

Appendix. Known city recommenders

Projects	Purpose	Method	City range	User interface
Bidart et al., 2014	To visit	CF	World-wide 85,505 cities	No map
www.datarevenue.com/en-blog/building-a-city-recommender-for-nomads	To visit	CF	World-wide 4,247 cities	No map
github.com/anitaokoh/City-Recommender-Web-App	To visit, live	Content-based	World-wide 111 cities	No map
devpost.com/software/the-global-aggie-city-recommender	To live	Content-based	World-wide 216 cities	With map
github.com/eliasmelul/finding_schitts	To live	Content-based	USA 672 cities	No map
teleport.org	To live	Content-based	World-wide 266 cities	No map
metroverse.cid.harvard.edu	For economic research	Not disclosed	World-wide >1000 cities	With Map