Finding a neighbourhood in a new city that is similar to where you used to live

New Neighbourhood Search

A capstone project for the Coursera IBM course: Applied Data Science

Leon Smith

A. Introduction

What if you secured a great new job in another city but you have come to love your current neighbourhood? Would moving to a similar neighbourhood in the new city make your decision easier? But what if you are not familiar with the new city and have no idea what neighbourhoods are similar to where you live now?

We live in a time where the workforce is increasingly mobile, not only within countries but also internationally. Moving to an unfamiliar city has many challenges, with finding the best place to live high on that list. A tool that makes this search easier will help users to settle into their new city quicker, rather than spend valuable weekends trudging from one neighbourhood to another to look for a pace to live.

Is there a way to narrow down the neighbourhoods to explore? If so, what criteria should be used to compare neighbourhoods?

Average income, average education, average household size, average age, average cost of accommodation? All these might be relevant, but it could be argued that one of the main reasons people grow to love their neighbourhoods is because of the services that the neighbourhood gives one access to: the parks, the coffee shops, the gyms, the bars and restaurants, etc.

The purpose of this project is to take as an input a user's current address and then suggest similar neighbourhoods in a target city. These suggestions could then serve as a starting point to narrow down a preferred location based on a users' own observations on the ground.

B. Data

Two data sets are used:

a) Details of the venues in the user's current neighbourhood
b) Details of facilities in neighbourhoods in the target city

For this project, we assume that the user is looking to move from New York City, USA to Toronto, Canada. Both data sets are generated with venue data accessed through Foursquare's API.

The Toronto data is structured based on neighbourhood definitions scraped from a Wikipedia website on Toronto's postal codes (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M). The New York data is taken from a json file with New York neighbourhood data. This file was made available as part of an assignment for the IBM "Applied Data Science Capstone" course on Coursera.

Neighbourhood coordinates used to query the Foursquare API are sourced by querying Google's Geocoding API.

Processed Toronto neighbourhood data used is in the following format:

	Neighbourhood	**Neighbourhood Latitude**	**Neighbourhood Longitude**	**Venue**	**Venue Latitude**	**Venue Longitude**
0	Parkwoods	43.753259	-79.329656	Brookbanks Park	43.751976	-79.332140
1	Parkwoods	43.753259	-79.329656	Variety Store	43.751974	-79.333114
2	Victoria Village	43.725882	-79.315572	Victoria Village Arena	43.723481	-79.315635
3	Victoria Village	43.725882	-79.315572	Portugril	43.725819	-79.312785
4	Victoria Village	43.725882	-79.315572	Tim Hortons	43.725517	-79.313103

where the data sources for the entries are as follows:

- Neighbourhood: Scraped from Wikipedia page
- Neighbourhood Latitude: Google's Geocoding API, based on Neighbourhood
- Neighbourhood Longitude: Google's Geocoding API, based on Neighbourhood
- Venue: Foursquare API, providing venues close to neighbourhood coordinates
- Venue Latitude: Foursquare API, latitude for venue
- Venue Longitude: Foursquare API, longitude for venue

C. Methodology

The most significant data issue to consider when comparing venue profiles across neighbourhoods is a basic one: the significant variance in the venue profiles of neighbourhoods.

The two most significant attributes that affect this analysis are:

a) Variability in venue types
b) Venue density

Cultural, geographical, and other differences lead to differences in the venue types available in reference and target cities. One might for example reasonably expect to come across coffee shops in both New York and Toronto neighbourhoods but American restaurants are much more likely to be found in New York than Toronto.

The result is that data sets may need to be aligned and backfilled to ensure meaningful comparability.

Venue density is a further consideration. A Manhattan neighbourhood is likely to have significantly more venues than a neighbourhood on the outskirts of Toronto. The algorithm used to compare neighbourhoods should take this into account to avoid spurious results.

To illustrate this, lets consider two ways to calculate the distance between the venue profiles of two neighbourhoods from a reference neighbourhood.

For our example, let us say that we have a reference neighbourhood REFERENCE that has an equal number of coffee shops and bars as venues within easy walking distance. We want to decide which of two candidate neighbourhoods, CANDIDATE A and CANDIDATE B, are most like the reference neighbourhood. The venue profile of the neighbourhoods are as follows:

Neighbourhood	Coffee shops	Bars
REFERENCE	7	7
CANDIDATE A	2	2
CANDIDATE B	7	0

Intuitively, CANDIDATE A is more like the REFERENCE neighbourhood as it also has an even split in coffee shops vs bars, whereas the CANDIDATE B neighbourhood has no coffee shops at all.

Euclidean distance

The first method investigated to compare neighbourhoods was to calculate the Euclidean distance:

Euclidean distance between REFERENCE and CANDIDATE A, the distance from (2,2) to (7,7):

$$\sqrt{(7-2)^2+(7-2)^2}$$

$$= \sqrt{5^2+5^2}$$

$$= \sqrt{50}$$

Euclidean distance between REFERENCE and CANDIDATE B, the distance from (7,0) to (7,7):

$$\sqrt{(7-7)^2+(7-0)^2}$$

$$= \sqrt{0+7^2}$$

$$= \sqrt{49}$$

Thus, measured by Euclidean distance, CANDIDATE B is closer to REFERENCE than CANDIDATE A – an unexpected result. This is partly due to the venue density issue highlighted earlier. The similarity in magnitude in the number of bars between CANDIDATE B and REFERENCE is overriding the similarity in composition between CANDIDATE A and REFERENCE.

Cosine distance

A method that is more sensitive to composition, rather than magnitude is required. Let us consider the Cosine distance method. This method (also referred to as the angular distance) measures the difference in angle between two vectors, ignoring magnitude.

The SciPy library documentation defines the cosine distance between two vectors *u* and *v* as:

$$1-\frac{u\cdot v}{\|u\|_2\|v\|_2}$$

Repeating the distance calculation using this method, we see:

Cosine distance between REFERENCE and CANDIDATE A, the distance between (2,2) and (7,7):

$$1 - \frac{(7\times2)+(7\times2)}{\sqrt{7^2+7^2}\times\sqrt{2^2+2^2}}$$

$$= 1 - \frac{28}{\sqrt{784}}$$

$$= 0$$

Cosine distance between REFERENCE and CANDIDATE B, the distance between (7,0) and (7,7):

$$1 - \frac{(7\times7)+(7\times0)}{\sqrt{7^2+7^2}\times\sqrt{7^2+0^2}}$$

$$= 1 - \frac{49}{\sqrt{4802}}$$

$$= 0.292893$$

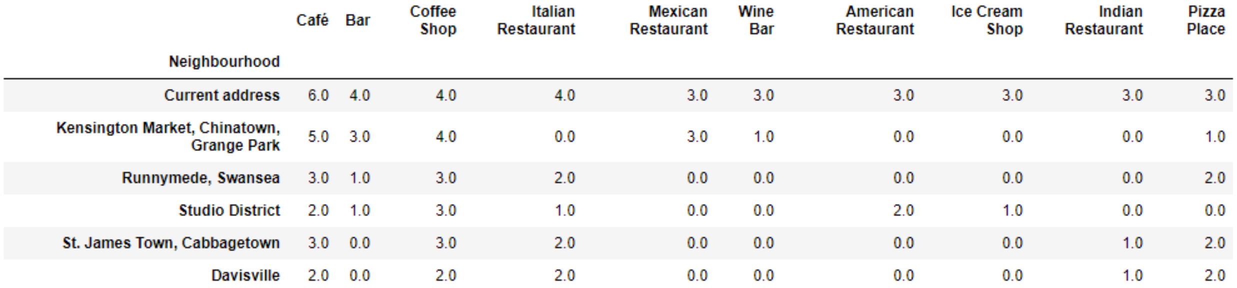
Measured by the cosine distance method, there is no difference in the composition of the REFERENCE neighbourhood and CANDIDATE A, with a non-zero difference between REFERENCE and CANDIDATE B.

In this implementation, it was decided to make use of the cosine distance method as the most appropriate way to determine neighbourhood similarity.

D. Results

Now let us turn our attention to a real-life example. As an input, we will use an address in the Upper West Side of New York City and output recommended neighbourhoods in Toronto that have similar venue profiles.

Table 1 below shows the neighbourhood recommendations based on the cosine distance method. The first line shows the profile of the New York input (current) address, followed by recommended Toronto neighbourhoods with the nearest match at the top (Kensington Market, Chinatown, Grange Park in this instance). The columns display the top 10 venue categories around the current address, while the values in the table represent the count of each venue category in the respective neighbourhoods.



	Café	Bar	Coffee Shop	Italian Restaurant	Mexican Restaurant	Wine Bar	American Restaurant	Ice Cream Shop	Indian Restaurant	Pizza Place
Neighbourhood										
Current address	6.0	4.0	4.0	4.0	3.0	3.0	3.0	3.0	3.0	3.0
Kensington Market, Chinatown, Grange Park	5.0	3.0	4.0	0.0	3.0	1.0	0.0	0.0	0.0	1.0
Runnymede, Swansea	3.0	1.0	3.0	2.0	0.0	0.0	0.0	0.0	0.0	2.0
Studio District	2.0	1.0	3.0	1.0	0.0	0.0	2.0	1.0	0.0	0.0
St. James Town, Cabbagetown	3.0	0.0	3.0	2.0	0.0	0.0	0.0	0.0	1.0	2.0
Davisville	2.0	0.0	2.0	2.0	0.0	0.0	0.0	0.0	1.0	2.0

Table 1: Count of venue categories by neighbourhood, based on top 10 venue categories around "Current address"

The cosine distances calculated are:

Neighbourhood	Cosine distance
Kensington Market, Chinatown, Grange Park	0.204357
Runnymede, Swansea	0.213643
Studio District	0.219578
St. James Town, Cabbagetown	0.230025
Davisville	0.236098

The Kensington Market/Chinatown/Grange Park neighbourhood is the top match. The additional neighbourhoods are presented for consideration in case the user wants to explore alternatives and to allow for instances where the top match may not be suitable due to practical considerations such as commuting distance and/or accommodation cost.

E. Discussion

The relatively good match in venue profile between the input address and the recommended neighbourhood indicates that the cosine distance approach has good potential to serve as a basis for recommending neighbourhoods in new cities.

A limitation of this approach is the potential variability in venue types, as mentioned in the Methodology section (see section C). As can be seen in the Results section (section D), there are number of instances where only one in the top five neighbourhoods have instances of a venue category. Examples of these are Mexican restaurants (only present in Kensington Market/Chinatown/Grange Park) and American restaurants (only present in the Studio District). To mitigate against distortion in the results, the number of venue categories used in the calculation should be set sufficiently high to allow for sparsely populated venue categories while avoiding over fitting. In this implementation, the comparison is based on ten venue category types.

Enhancements that end users might find helpful include comparisons of:

- commuting times
- demographics
- average accommodation costs

F. Conclusion

This project set out to generate suggestions for users looking for neighbourhoods in a new city that has a similar venue profile to where they currently live.

It has been demonstrated that using readily available data, appropriate data structuring and methods provided in Python libraries, this is readily achievable. In particular, recommendations were generated of Toronto neighbourhoods that has a similar venue profile to a New York address.