Fall
2018

Library Usage Patterns in UC Berkeley

INFORMATION ORGANIZATION AND RETREIVAL FINAL REPORT

NITHYA RAMGOPAL | SRIKAR VARANASI | NICK DEKUTOSKI

Contents

1. Problem Statement

One of the most frequent complaints heard from the UC Berkeley Library staff is that the libraries are too busy and students do not find space to study. UC Berkeley has a number of libraries in the campus space. However, most students only visit few of the libraries such as Moffit and Doe, and the vast majority of space on campus is underutilized. The library staff have also found through interviews[1] with students that this is a major problem. One of the student talked about the Moffit library and quoted it was *"often very crowded, and it's cut throat finding seats."* Both the students and the library staff would like to get the optimal usage of the library spaces and ensure students are assisted in finding places where they can study.

We built an information retrieval system around spaces in libraries with more personalized recommendations and filters based on usage patterns and library attributes. This system will help students get ranked recommendations that suggest times and libraries based on preferences. E.g., 'I wish for a quieter study time in the evening' would return best the timings to visit and which library to visit.

2. Data & Resources

UC Berkeley Library staff installed gate sensors in 2017 in order to collect data on library traffic and better understand the usage patterns and potential solutions. We were granted access to this data, which included hourly library gate traffic, and included information such as:

- Date & Time
- Library & Entrance
- Entry (No. of entries in the hour)
- Exit (No. of exits in the hour)

We have also collected the data on libraries and descriptions from the UC Berkeley website and the additional manual categorization. We created attributes that describe each library building, such as the following:

- Open weekends
- Open late
- Study rooms available
- Quiet study

We have organized library and time as the resources which we help the users retrieve through the recommendations. We solved problems such as :

- Given the time period, we provide the best recommendations to user for the library that they can visit based on their preferences
- Given the time period and library, we provide the best recommendations to user for the time slots that they can visit based on their preferences

[1] https://docs.google.com/document/d/1OmBh0KX-VdR0MFiUFVKL6WEKXYt-MNnrDMP5dX0xxFk/edit

3. Organizing System Design

We designed our organization system to retrieve two types of resources; **time** to visit the library; and which **library** to visit. Design choices were made in organizing each resource and translating the original dataset into our organization retrieval system that affected the granularity of the dataset and the information the user received. We also looked into the trade-off between recall and precision in a users' query result.

3.1 Traffic Data

We used the sensor data at the entrance and exits in the libraries to organize the **time.**
In organizing this resource, we had to make design decisions based on the interactions we wanted to facilitate, information available to us and limitations of that information.

1. Organizing the data required a judgement on what the structure of the dataset should be, and what the relationships among the data were. Our initial dataset included .csv files for each day; that needed to be consolidated before information could be used further. We designed an automated script that loaded all data and stored it in a common table, so that we could update or handle new data in the future without much effort.

2. We used the existing structure of the dataset to create additional features and further organize the data for our retrieval system. For example, grouping entrances by their library name and building and filling missing data of the library names. This process involved judgement of granularity and determining what is a single resource based on the intended use. Because each building can contain multiple interconnected libraries; it's possible for a person to enter one library entrance and exit through another. For example, someone entering Bancroft Library in order to reach Stacks Library may exit through the West Doe entrance. We decided to treat "Library Buildings" as one thing, and save information on additional levels of granularity such as the specific entrance/exit, or interconnected libraries.

   We decided to aggregate data on the individual entrances, and treat them as one "part" of a library building. This decision choice was made with the recommendation of Library staff, and their use-case in analyzing traffic at the building level which would be most interpretable.

3. The aggregation of "time" was also a decision in designing our system. We determined to use different levels of granularity in our back-end database, compared to the query result for the user. Our raw time data was at a granularity of per hour and per entrance. However, this information was too precise and prone to day of week and day of month fluctuation(*Appendix Fig 6.2.1*) . We noticed that storing data at a week level could be problematic since there was quite a bit of variation in the traffic between different days of the week, and there was more traffic during weekdays than weekends. This could affect the weekly averages (*Appendix Fig 6.2.2*).There was a lot of fluctation. For example,

first 2 weeks of December are very busy and last 2 weeks have almost no traffic. So the average hourly traffic for December could be very misleading.

We decided to use the week number and day number combination to store our traffic data. For example, every year second week of December (i.e week 50) tends to be the Finals week where all days are very busy including weekends. This, methodology worked quite well in enabling the user to find specific times because UC Berkeley's exam and holiday schedule has been quite consistent over the years. This granularity was also chosen with the idea that users are desiring to choose a specific date to study rather than a temporal trend such as Day of Week. We imagine users looking for a place to study at the current date & time, or within 48 hours of it as the most probable use-case; and wanted to be sure- to have data for all dates of the year; not only days of the week.

4. We chose to return time to a user at a different granularity, based on our judgement of what unit of time would be most useful for the interaction of determining when to visit a library. We made our categories less granular by creating categories that grouped each hour of day into a bucket of times; for example "Afternoon" was defined as 12-4 pm. The choice of creating buckets of time at a broader granularity reduced the precision of the query result returned to the user as users would no longer be able to access information for each hour of day; but improved the recall- as users would be able to see libraries that were available for a broader range of time (e.g. libraries that open at 6am would also be shown with libraries that are open at 8am).

   This choice is based on the assumption that users does not have an exact hourly preference in mind when they search for study space; and that users prefer a broader time block for a sustained trip to the library. This assumption would not hold true if users are looking for an exact time period at which they wish to study; for example- if a user is looking for a study spot between 1-2pm to have a group meeting in between their classes. In the future; different levels of granularity of time could be offered to allow for increased precision if desired.

3.2 Library data

To organize Library descriptions and features; we manually scrapped the UC Berkeley website to extract the important features of every library. Each library had a lot of features such as: group_study_allowed, quiet_study, food_allowed, reservable_rooms, projecter_available etc. We could also get information such as open time, close time, and the GPS coordinates.

1. Feature selection: We needed to decide which features we wanted to include in our system. Based on our own experiences, and talking to a few of our classmates that the quiet study, group study, reservable room, GPS coordinates and the timings of the library were the most important things that influence a students' decision to visit a library. Adding these attributes to our library made it easy to organize our data to

respond to user queries such as 'Which libraries have reservable rooms and open at 7pm ?'.

2. Feature encoding: Most of the features we had could not be used in the raw format. For example, the GPS coordinates alone are not useful unless they are used to inform the user how far each library is. We created a distance function that could calculate the distance between a user and each library. This is a dynamic feature. We also encoded variables such as group_study and reservable_rooms to binary to facilitate easier filtering and search operations for the users.

3. Categorization: We had to group all the libraries which were interconnected into categories which denoted the building because the hourly traffic information was aggregated at the building level. In order to make meaningful comparisons, we had to ensure both the datasets were at the same level of granularity

4. Retrieval System Design

The guiding principle for the information retrieval system was enabling quick and easy access to the information on what time to visit a library and which libraries to visit. The retrieval system we designed can support the following queries:

4.1 Best time

Users can select a library of their choice and choose when they would like to study (morning/ afternoon/ evening). The system will return the most suitable time-slots to visit the library ranked from best to worst (*Appendix Fig 6.3.1*) . This interaction was designed to facilitate finding a resource that you already know exists; as we are assuming that the user is familiar with at least one library on campus which they desire to visit.

As a result, we then enable user to select a resource (time) from a list of recommendations. We decided to allow the user to choose morning/ afternoon/ evening in order to return a more generalized responses to the query that suits the users' preference. For example, if the best time to study at DOE is 9am -11am, but the user has a class during that time, the recommendation is not very useful. Hence, we ask the user for a rough time range when they plan to study such as morning/afternoon/evening which helps us return more useful recommendations that are more generalized. Sometimes the library the user wants to go to might be very busy. So, the system also lists other libraries which have similar features to the one the user looked up to give them more options.

4.2 Best Library

Our system recommends libraries a user can visit; based on selection of their preferences for library attributes(*Appendix Fig 6.3.2*). For example, users can select attributes of a library they

would like to visit, such as whether the library allows quiet study, group study, or has reservable rooms and receive recommendations of libraries that are similar to this query. Based on the selection, users can see the list of recommended libraries and a graph which shows the hourly traffic in each library as the day progresses.

In designing our retrieval system; we are assuming that the determining factor for which library a user would like to visit is attributes of the library. Our system is designed with the interaction of a student "studying" in mind. We are implicitly assuming that the user has some familiarity with the attributes of libraries; and that access to information on library attributes will influence their decision of where to study. Additionally, our resource descriptions were limited by convenience of information available; as well as subjective judgement on which attributes would be important to UC Berkeley students for the purpose of finding a place to study. We recognize there could be other factors that determine which libraries are preferred by users that are not captured in our data; such as where a users friends are, how long they wish to study for, or an infinite number of extrinsic attributes and we plan to extend this in the future.

Additionally, our our organization of the resources facilitates specific interactions, but is limited in its generalization to other types of search and discovery. There are other interactions that may occur with libraries other than our intended use case which may be less suited with our retrieval system. For example, someone looking to check out a particular book, in which case library resources may not be as interchangeable. Checking out a specialized Mathematics textbook may require you to visit the Math Library.

The system also returns the libraries that are geographically closest to the user. This feature uses the GPS coordinates of the user (which was assumed to be south hall for the demo) and calculates the aerial distance between the user and every other library. While this gives us the rough estimate of the distance of each library it is not the ideal distance metric because we are not considering the practical travelling distance based on campus paths, on-campus roads, and traffic.

4.3 Similar Libraries Calculation

As a potential solution to the issue of congestion in the most popular libraries, we wanted to help users find libraries that had similar attributes to the library they are interested to visit; but the user might not be aware of. Most users are familiar with the Doe and Moffitt Libraries, and may want to go there. However, these libraries tend to be very busy and can't accommodate all students. Based on the users' preference of the library, the system will recommend other similar libraries which the user can choose.

To achieve this, we calculated a similarity matrix for the libraries in our data based on a subset of metrics. Our current implementation used the metrics of opening hours, quiet study, and group study (*Appendix Fig 6.1.1*) . Our choice of which metrics to use in calculating similarity was a design decision based on our judgement of what we expect to be important to the end-user; and what interactions we want to enable. However, this is approach is naive, and makes significant assumptions about a subset of features that we hope to iterate on in the future.

There may be other factors that influence a users preference; such as travel distance to the library or a multitude of other features not captured in our dataset.

We decided to use cosine similarity as the distance metric for our similarity calculation. An advantage of cosine similarity is that variables are scaled so the magnitude of a value does not have an effect. This is useful for metrics such as boolean variables that are already scaled between 1 or 0, or words where we're less concerned about the difference between a word that occurs once compared to 3 times. Our recommendations identifies libraries with similar features and closing times (see appendix). However, we think using a different distance metric, such as Euclidean distance would improve our recommendation system and allow us to more flexibly incorporate additional features in the future. For example, if a user selects Moffit because of its geographic distance from their location, the magnitude of the distance feature is important to the user's decision. Additionally, our recommendations assume that all features are weighted equally. However, the closing time could be more important than group study. We would like to address this in the future through weighting attributes, or exploring collaborative filtering to better understand feature importance to a variety of users.

5. Next Steps

We believe the following steps would improve our system:

1. Having an index based on utilization rates for each library. We plan to show an index ranging from 1 - 5 (with 5 being the highest utilization) would help the users easily understand the libraries that they need to use.
2. Integrating with the real time data. We want to integrate historical insights with the real time data to give more accurate predictions. Currently, we only use the historical data to make future predictions
3. Geo-locate the user using app and give better recommendations for similar libraries
4. Providing error-bands for the predictions can improve our understanding of the model accuracy

6. Appendix

6.1 Similarity

Fig 6.1.1 Similarity table for CHEM Library

```
feats = ['close','quiet_study','group_study']

summarize_similarity('CHEM', feats)
```

```
Similar Libraries to CHEM
```

	Building	close	quiet_study	group_study	Cosine_Similarity
5	CHEM	19	1.0	1.0	1.000000
17	OPTO	19	1.0	1.0	1.000000
8	EART	19	1.0	1.0	1.000000
0	AHC	21	1.0	1.0	0.999044
2	DOE	17	1.0	1.0	0.998943

The most similar libraries are the Optomitry Library, and Earth Sciences & Map Library. Both libararies are open during the same hours, and share the same group and quiet study attributes.

Fig 6.1.2 Similarity table for MATH Library

```
summarize_similarity('MATH', feats)
```

```
Similar Libraries to MATH
```

	Building	close	quiet_study	group_study	Cosine_Similarity
12	MATH	17	1.0	0.0	1.000000
14	MORR	17	1.0	0.0	1.000000
19	SOCR	18	1.0	0.0	0.999630
18	PHYS	19	1.0	0.0	0.998578
16	NEWS	19	1.0	0.0	0.998578

The most similar libraries are Morrison Library, and Social Science Research Libary. These libraries have similar hours, allow for quiet study time, but do not allow group study.

6.2 Traffic Analysis

Fig 6.2.1 Plot for day vs total entries for Moffit library(November)

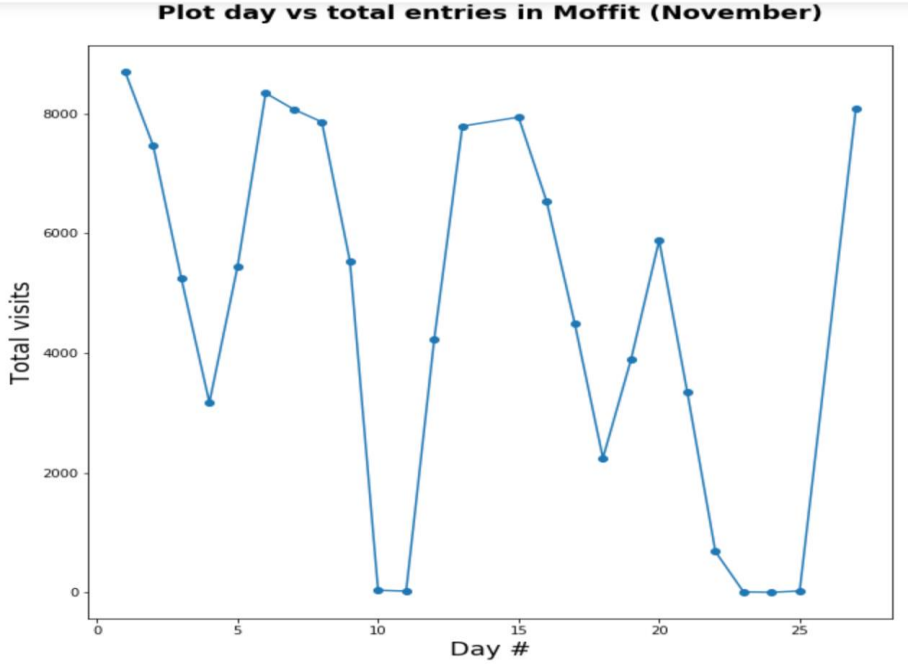
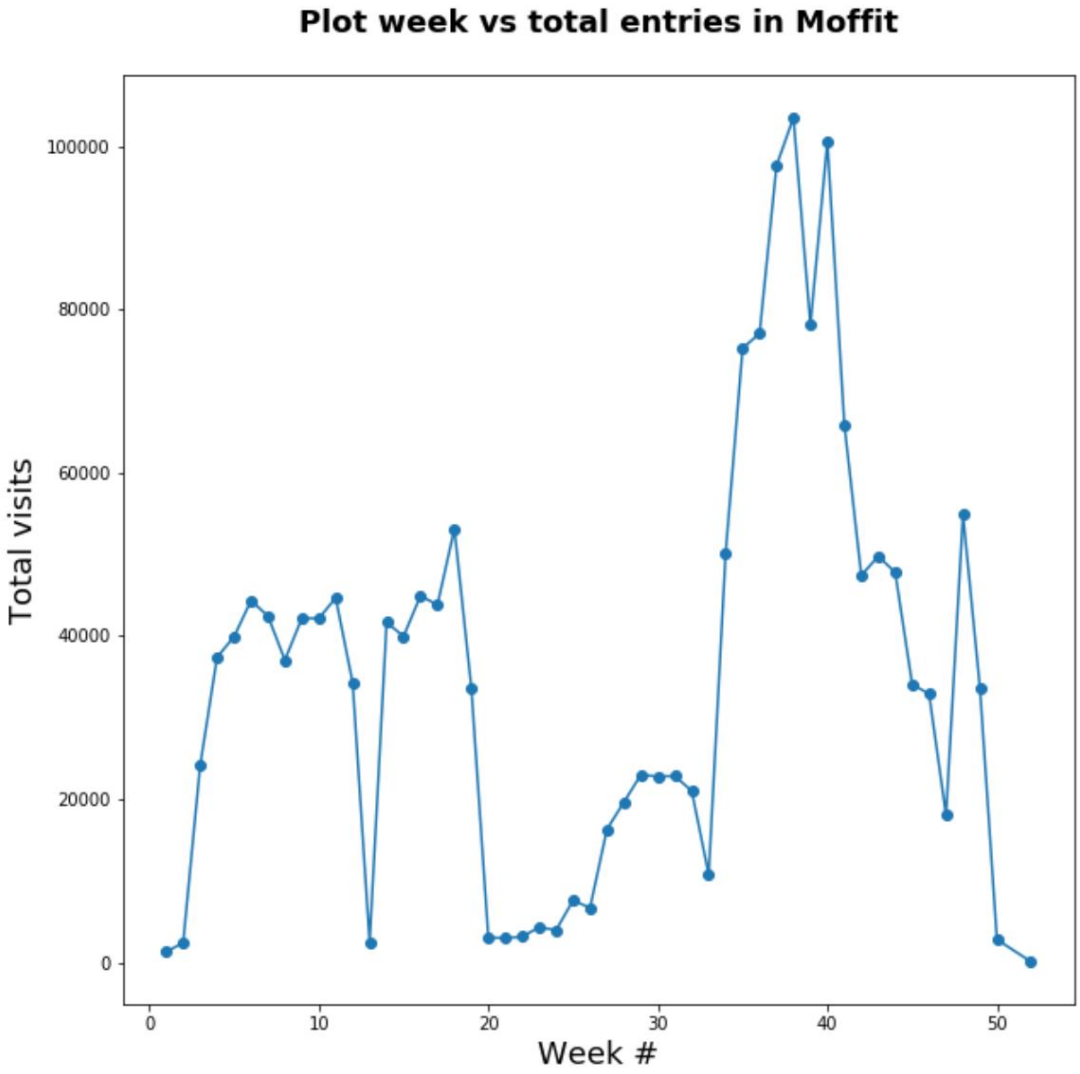
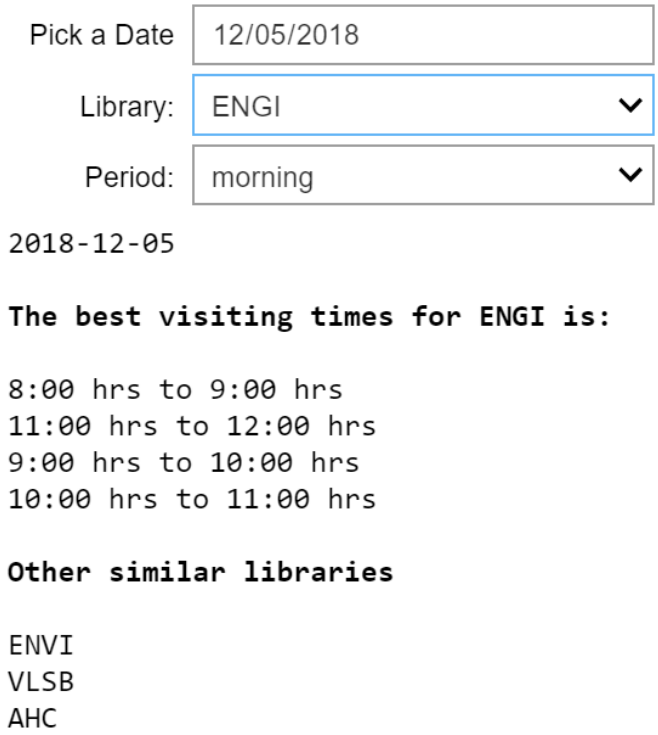


Fig 6.2.2 Plot for week number vs total entries for Moffit



6.3 Retrieval System

Fig 6.3.1 Prototype that recommends best times to visit for a library



If the user wants to know which is the best time to visit a library, they can select the library and the time period they plan to study (eg. morning) and see what is the best time to visit. However, sometimes the library the user wants to go to might be very busy. So, the system lists other libraries which are similar to the one the user looked up to give them more options.

Fig 6.3.2 Prototype that recommends best times and libraries closest to the user preferences