A Prediction-based Collaborative-Filtering Event Recommendation System

Gregory Wyatt Slominski
College of Charleston / Citadel
Computer Science – Software Engineering
Charleston, South Carolina
slominskiw@g.cofc.edu

***Abstract*—This paper aims to illustrate the understanding of human movement patterns, using this information to create a theoretical event recommendation system. Machine learning techniques will be evaluated to predict the next location a user would likely go. Taking into consideration that humans move in patterns, this paper will examine collaborative filtering methods that implements the understanding of why individuals transfer from one location to another and create a theoretical recommendation service for each individual. The recommendation service aims to provide physical event entertainment recommendations based on user's history features, those that move in a similar fashion, and places that resemble a high subliminal correlation.**

Keywords— Global Positioning System (GPS), Support Vector Machine (SVM), collaborative filtering, events, recommendation system.

I. Introduction

Studies have been conducted that illustrate the ability to predict where humans will move to based on the time of day, their current location, and their previous movement patterns [1, 2, 3]. A team from Da Nang, Vietnam released a study in 2017 at the International Conference of Information Science and Technology that implemented a Support Vector Machine (SVM) that was able to predict where someone would move to with 92% accuracy [1]. Later, this model, as well as other prediction models will be studied to show the resemblance in individuals unique movement patterns. To gain an understanding that there are those that move in a comparable manor, key features from individuals will be examined to cluster together similar nodes. Users that do not know each other have the potential to have a similar transition history and in return, could value the same events. For this study, events are categorized as series of time intervals held at unique locations. Events can be placed in one or many categories, which is important for distinguishing the psychology behind why users attend.

An example of an event could be Taco Tuesday at Taco-Boy every Tuesday from 4PM – 6PM (where all tacos are half-off). This event would fall into multiple categories, such as food, discount, and Mexican. Implementing a collaborative filtering-based approach, users that have appreciate similar category features would be clustered together. What separates this location-based recommendation service (LBRS) from others are the unique features that are focused on to create the clusters.

The idea is to show that is possible to group individuals together to create recommendations based on their historical movement patterns, as well as grouping events together based on their similarity. To do so, users are evaluated based on how they move, ie are they going to work early and working long days, or do they sleep in and work short part-time shifts; when going out in their spare time, how long are the users staying at each event, and how long does the event usually retain its guest; what kind of events do users find themselves attending. The results provided in this study convey the idea of further exploration in this topic with real user interaction, as seen in figure 1.

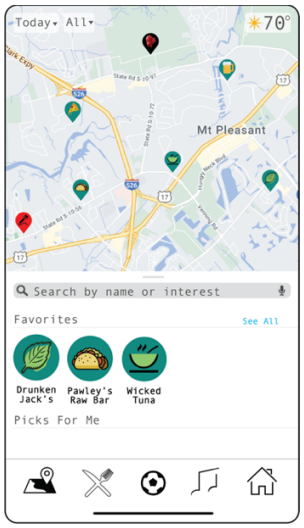


Figure 1: A mobile application interface illustrating the ability for an event recommendation service.

The rest of this paper is organized as follows. In Section II, three next place prediction models will be studied and evaluated. Section III will examine industry recommendation services implemented by Netflix and Amazon, and how these practices can be evaluated for a location recommendation service. Section

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

IV will be the conclusion and Section V will focus on moving forward.

II. Next Place Prediction

A. Prelimenary

Being able to accurately predict the next place a user would go is a valuable feature in deciding which kind of event to recommend. If a model indicates an event that a user would naturally head towards, then an accurate alternative event recommendation could be provided. For example, if a model predicts that a user is going to Taco Tuesday on Tuesday at five, and that user is open to another recommendation, then a model could provide another similar event that would entice that user (ie., half off wings at Westly's). When predicting where a user would go, understanding why they would go there can provide insight for a future recommendation. Due to the importance of being able to predict where a user will go, the following Section II subsections will evaluate three scholarly papers geared towards next place prediction.

The first discussed paper will be, *A Potential Approach for Mobility Prediction Using GPS Data,* and the SVM model that provided the strongest results. Second, *Next Place Prediction using Mobility Markov Chains* and mobility model, "coined as *n*-MMC," to predict the next location based on the previous n destinations [2]. Last to be evaluated will be, *When and Where: Predicting Human Movements Based on Social Spatial-Temporal Events*, and in particular, the density based temporal clustering algorithm to learn patterns and make predictions in terms of time. The goal of this is to closely evaluate three accurate prediction models to decide which would provide optimal results, in terms of speed and accuracy.

B. A Potential Approach for Mobility Prediction Using GPS Data [1]

A supervised learning approach was taken to generate the outcome based on previous results. Three major procedures that generate a next place projection model are data collection, feature extraction, and model learning. The four authors, B. Nguyen, N. Nguyen, N. Nguyen, and M. Tran developed a mobile application that kept track of roughly 200 GPS information for the course of 5 years. Associated with each user was their location, including the address, latitude, longitude, city and country code, tracked every five minutes. Over time, as distinct destinations became apparent, location clusters were created using a K-mean algorithm. With a set number of locations chosen, the algorithm calculated the centroid of each cluster adjusting the mean vector based on the new values added or removed to and from their respected cluster. Once specific historic locations for each user were identified, the prediction model had output data to choose from when deciding which cluster each location belonged to. The features focused on to generate next place prediction were *current location, data and time, frequency of appearance, distance,* and *final feature.* The *current location* was an important feature to extract because based on the historical data provided by from a user, if they proceeded point A by being at point B, then with high probability, this feature alone could provide accurate results. A set of K current locations was kept, based on the K-mean algorithm previously mentioned. The *date and time* that a user visited a location was extracted and utilized to evaluate the patterns throughout the week. It was able to collected that, as expected, most users traveled to a place of work during morning hours of the weekdays, whereas spent time going to the movies or the mall in the afternoon on weekends. The time range the authors kept track of fell between 0AM-6AM, 6AM-12PM, 12PM-6PM, and 6PM-Midnight, and numbers 1 – 4 (h_range) were respectively associated with each time frame. The *frequency of appearance* during a period time was found to be a valuable feature when predicting where a user would be during a range of time. With K locations, a vector was created to keep track of when a user was likely to be at each location. The K location would fall into one of the time ranges numbered 1 – 4. The given *distance* between two notable locations was evaluated to decide whether a user would likely go to a predicted location next. The authors calculated the ratio between the distance of the edge travelled to and the possible next immediate edges that the user could travel, based on their history. The *final feature* used was a collection of the other features combined into one. For each entry recorded, the date of the location, week, h_range, frequency of appearance vector and distance range were all compiled into one feature [1].

The approach that was taken by the authors challenged three models: SVM, decision tree and Markov models; training and testing with five cases. Splitting the data into 70% training and 30% testing, the two highest scoring cases came when the *final feature* (merging of all features) and *date and time* (location vector) alone were used. The results from the *final feature* case provided a high of 92% accuracy from the SVM model and low of 80% from the Markov model, as seen in figure 3. The *date and time* case had its best results generated between the hours 0AM – 6AM (95% accuracy by decision tree) and 6PM – Midnight (88% accuracy by decision tree); its lowest results were generated by the Markov model between 6AM – 12PM (68% accuracy), as seen in figure 2 [1].

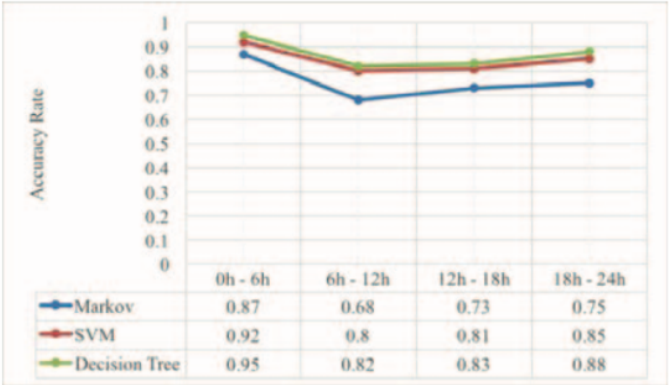


Figure 2: Accuracy results for the three algorithms tested based *date and time* [1]

While the *date and time* case had the highest overall accuracy, it is not considered to be the most reliable, partially in point, because only 4 place predictions are provided. The most valuable outcome provided by this case study in generated from

the *final feature* case when run with the SVM model. Combining the four features together provided the most well-rounded feature to analize when predicting the next location.

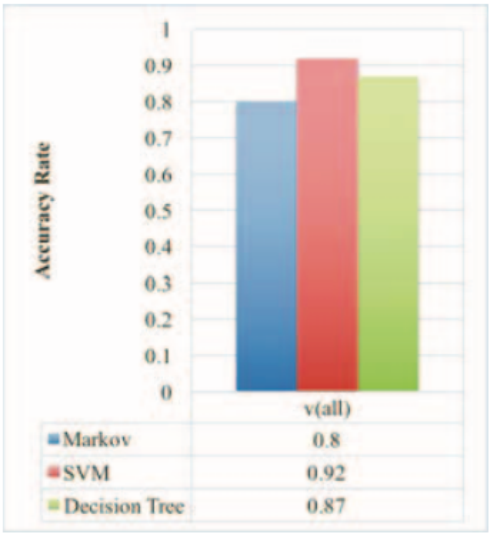
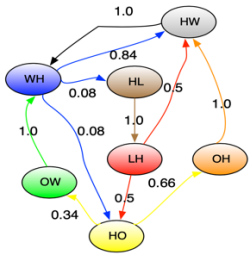


Figure 3: Accuracy rates when utilizing the *final feature* [1].

From this article, value is gained when attempting to recommend an event that an individual would enjoy attending. Understanding where someone would like to go next, with 92% accuracy, could allow for further investigation as to why they would like to go to that location, and in turn provide inciteful suggestions. A con, however, is that using a feature that is comprised of four other features, would have a relatively long compile and runtime, which may not prove to be ideal when calculating an immediate next place recommendation.

Source/Dest.	H	W	L	O
H W	1,00	0,00	0,00	0,00
H L	1,00	0,00	0,00	0,00
H O	0,64	0,34	0,00	0,00
W H	0,00	0,84	0,08	0,08
L H	0,00	0,50	0,00	0,50
O H	0,00	1,00	0,00	0,00
O W	1,00	0,00	0,00	0,00



Figures 4 [2] and 5 [2]: Home (H), Work (W), Lost (L) and Other (O), create a matrix to represent the previous two states and the probability of the next state. Edges create a visual representation of the probabity.

C. Next Place Prediction using Mobility Markov Chains [2]

This paper utilizes the common Mobility Markov Chain (MMC) as foundation and builds upon it. The traditional MMC model takes into consideration a set of predetermined states, that are sorted in descending order based on frequency visited. A set of transitions is maintained which identifies the likelihood of an individual traveling from one state to the next. For instance, an individual at the grocery store would have a high transition probability of going home next. The standard MMCs are referred to as memoryless, in the sense that they only evaluate their current location for predicting the next location (ie, a series of past locations is not utilized to predict where the next location is going to be). The difference with n-MMC is that the algorithm learns a trail of previous locations to predict the future location. The process of learning the mobility traces of an individual is achieved in two steps: a clustering points together based on location, and an algorithm to compute the transitions between locations. The authors, Gambs, Killijian, and Cortez chose to use a Density-Joinable cluster (DJ-Cluster) method to cluster points together. The DJ-Cluster algorithm takes three input parameters: the minimal number of points necessary to for a cluster (MPN), the maximum radius of the cluster, and a merging distance for clusters. DJ-Cluster can be broken down into three steps. The first phase consists of eliminating all irrelevant data. The second phrase consists of clustering the remaining elements to groups whose size is greater than or equal to the MPN. With the clusters created, the final stage involves merging small clusters with at least one shared element together into one larger cluster, and then calculating the centroid of the clusters and merging those who are within the predetermined margining distance set. Once the DJ-Cluster algorithm has created the distinguished groups, the transitions between locations is calculated, sorting the elements into chronological order, and associating them with their cluster group (elements that are found not to be associated with a cluster group are removed). After removal of outlier elements, clusters that are found consecutively are grouped together for a single trace. With the singular dimension of data create and organized in chronological order the next place prediction can take place. To calculate the next destination a node would travel, based on the previous n-locations, a transition matrix is generated, as seen in figure 4. For the example, assume a person has three state clusters they travel to: Home (H), Work (W), and Restaurant (R). If a model was interested in seeing the next projected place that a user would travel, based on the last two locations they registered, calculatations for each of the potential previous states (HW, HR, WH, WR, RH, RW) and the percent chance of the next state, would be derived [2].

The accuracy and predictability were calculated using three datasets, ranging from 1 to 175 users, and the training and test model checked between previous 1-4 locations. Scoring with over 90% accuracy, the highest predictability was generated using the previous two locations but was significantly lower for all other tests.

Understanding that a users' most consistent probability of projected destination can be generated be using their previous two locations, can provide to be useful. However, if a user had many destinations, a finite state machine used to run this model could be computationally expensive, and in result slow when integrated with a recommendation model.

D. *When and Where: Predicting Human Movements Based on Social Spatial-Temporal Events [3]*

By evaluating the "periodicity and the sociality of human movements," the concept of Social Spatial-Temporal Event (SSTE) is used to weigh social interactions between people based on when and where they occur [3]. For each person generating data, a subgraph (represented by a node) is created, containing two spatial-temporal attributes, the time and location that a social interaction is tracked. Edges connecting nodes represent a relationship between two nodes. Integrated with an AutoRegressive Moving Average (ARMA) model, that is incrementally updated with the use of a Kalman Filter based learning algorithm, helps predict the time that movement is going to occur. The goal of the time prediction is to estimate how much time will pass before the next location movement occurs. For the location prediction, a ranking model based on the frequency and the sociality movement patterns is utilized. *Checkin* records, *r,* are a series of 3-tuple (v, τ, σ), where *r.v* is a person that checks in from the set *G.V, r.*τ is the *checkin* time, and *r.* σ is the *checkin* location. Social Spatial-Temporal Event is a 3-tuple (*G*, τ, σ), denoted by *a*. *a.G* represents a subgraph of the friendship graph created by *checkins* (*G), a.*τ is the time the tuple is created (*a*), and *a.*σ is the location where the tuple is created (*a*). To predict the next time that a SSTE tuple will be generated a recursive algorithm based on the estimator of the last time, the new observation, the historical observations, and the covariance of the estimate error at time is calculated and constantly refigured. The periodic behavior is evaluated to predict the location at the future time, which looks at the desired future time correlated to the relevant related past time. For example, when attempting to predict the location for noon next Friday, the previous Friday noon records help make the prediction [3].

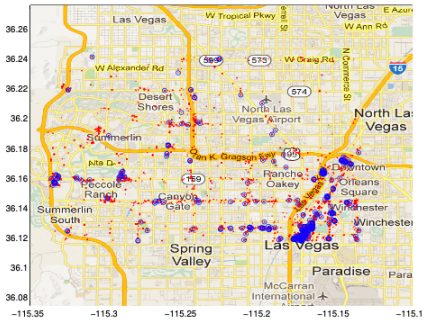


Figure 6: SSTE in Chicago, IL [3].

The test for next time prediction are based on three datasets (Chicago, New York, and Las Vegas) where users check-in, and are evaluated by their mean squared error (MSE) on the training dataset from 20% to 90%. For all datasets there was an inverse relationship between the MSE and training. As the training increased the MSE decreased, which corresponds with the expectation that as more data is passed in the algorithm is able to adapt and learn, as seen in figure 7. The test for location prediction generated a list of size N, for the potential places. As the list size increased and the training sample increased the algorithm became more accurate [3].

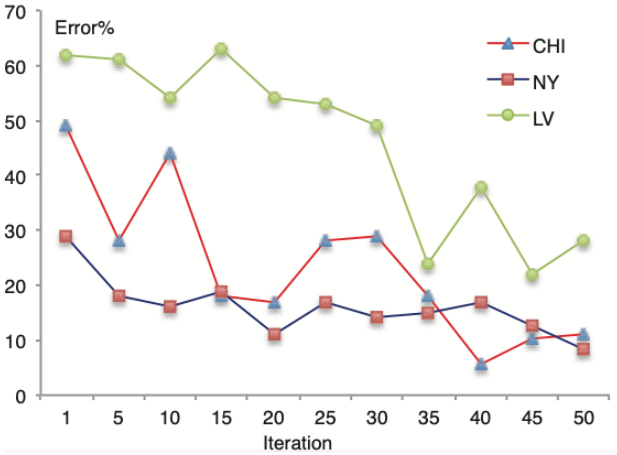


Figure 7: The percent in error is decreased for as the iterations in training increases [3].

When creating a recommendation service, the time that particular events are proposed could have an impact on the user. Integrating the time related service to understand when users are going to move based on the social implications could provide to be valuable from a user point of view. In contrast, given that the recommendation service would often provide events that users have not attended, proposing the idea of going somewhere that a person has not been before would likely need to be introduced beforehand. The marketing rule of 7 states, "a prospect needs to "hear" the advertiser's message at least 7 times before they'll take action" [4]. With that being said, it is unlikely that proposing the idea as a potential prospect is walking out the door would provide much value.

E. *Schoraly Article Recap*

Predicting an individual's next location would provide valuable insight to understand the events that they enjoy attending, allowing for positive recommendations. As illustrated in the *Next Place Prediction using Mobility Markov Chains,* viewing the last two locations a user has attended can accurately help decide where they will attend next. However, as noted in the article, when multiple locations were plotted together, many were discarded and combined into one, which allows for the consideration the amount of time that a user stays in one location. *When and Where: Predicting Human Movement Based on Social Spatial-Temporal Events* demonstrated that it is possible to predict the next time that a user is looking to attend an event and where. In contrast, this method relied heavily on the social structure surrounded by the node, which does assume

that others outside the users' friend group, may have similar patterns. *A Potential Approach for Mobility Prediction using GPS Data* revealed that a *final feature* produces accurate results when predicting where a node would likely travel next. Create a feature vector specific to each individual is enticing because not only can their location be predicted, but nodes with similar patterns can be clustered together. Even if to users do not cross paths, they may have a similar movement pattern, and in return enjoy spending relatively similar time at similar locations. Finding a way to group users together is a foundation of collaborative filtering, which is how companies such as Amazon and Netflix create their recommendation service. Being able to extract unique features from individuals in the next location prediction phase would allow for less extraction to be done during the recommendation phase; because of this, the optimal next place prediction, to be utilized in tangent with an event recommendation service, would be the SVM model utilized in *A Potential Approach for Mobility Prediction using GPS Data*.

III. Recommendation Models

A. Preliminary

Common successful recommendation services today are built around online recommendations for products or services on their site. Spotify creates daily playlist for users based on music that is perceived to be enjoyed by the user, even if the user has not heard of the artist before. Netflix utilizes a variety of recommendation features on their site to propose film suggestions to users. The Netflix Recommendation System is so powerful that it is currently accredited for the high customer retention, which, "translates to savings on customer acquisition (estimated $1B per year as of 2016) [5]" Amazons' recommendation system has been so powerful over the last two decades that in 2017 it won the "test of time" award published by the *IEEE Internet Computing* for its paper "Amazon.com Recommendations: Item-to-Item Collaborative Filtering." Collaborative Filtering (CF) is a technique that groups together users based on their preferences or items based on their relevant transaction history. In the following two subsections, the Netflix recommendation system and Amazon recommendation system will both be examined. The final subsection here will discuss a theoretical recommendation system model that suggests events most likely to be enjoyed by the individuals.

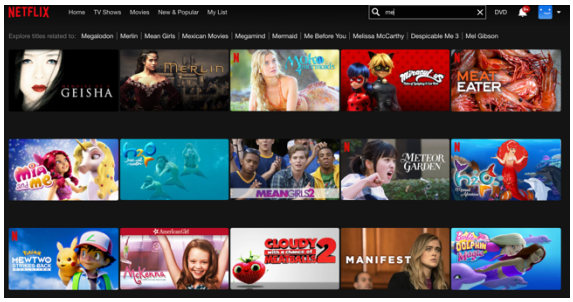


Figure 8: In the search engine, "me" is typed and the recommendations are as seen (many films that contain the letters "me", as well as topics such as "Mexican Movies", and actresses such as "Melissa McCarthy).

B. Netflix Recommendation System [10]

Research suggest that Netflix has roughly 60 to 90 seconds to help a customer chose a title before the user will lose interest. In this time, it is calculated that viewers will see between 10 and 20 titles, while viewing 3 in detail. Utilizing a variety of collaborative filtering machine learning algorithms, Netflix puts films in-front of users that they would spend time watching. The Netflix homepage is comprised of a matrix of recommendations, containing roughly 40 rows, up to three visible at a time on most devices. Each row contains up to 75 columns worth of titles per row (5 visible), equating to 3,000 potential titles for the user to choose from. Studies have shown, when someone has to many options, they often are left overwhelmed and decision-less [6]. Netflix has managed to avoid this issue by understanding what each viewer is interested in and quickly providing an optimal recommendation. Out of the 40 rows generated per homepage, 5 will briefly be discussed, along with the page generation, search algorithm and evidence selection [10].

The algorithm that is attributed for generating the rows broken down by genres is the Personalized Video Ranker (PVR). Utilizing a mixture of user-based and item-based collaborative filtering, the PVR creates a personalized catalog of movies. The user-based feature organizes films by their predicted rating, and the item-based feature organizes films by their popularity. These two rating are plotted together to create a 2-dimension dataset then dimensional reduced to generate a linear row of suggested videos [7]. After the clips are placed in their row, the row itself is generated a score, which plays into the placement on the homepage.

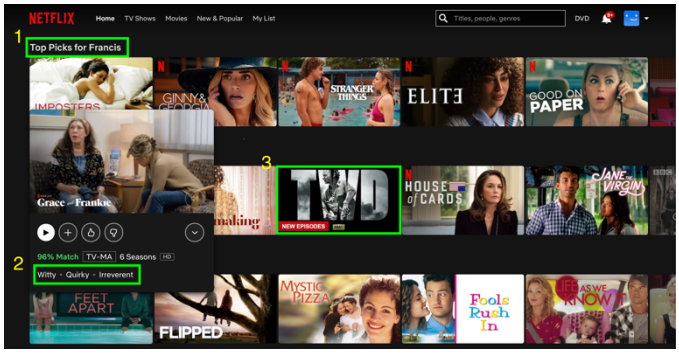


Figure 9: 1 illustrates the Top-N row for the specific user – 2 and 3 illustrate the evidence decides to show to the user.

The Top-N Video Ranker utilizes a heavier user-based CF, in that the goal of this algorithm is to cluster the best personalized recommendations out of the entire catalog, regardless of genre, which provides a freedom that the PVR does not have. The top recommendations are sorted into a row for the user to go through, as seen in figure 9. This algorithm does take into consideration popular videos, as well as seasonal videos as they are believed to be highly recommended for many viewers. Once the videos that meet the Top-N Video Ranker criteria are selected they are organized in their row according to their weight. Trending Now is another row that is generated by the Netflix recommendation system, and it focuses on short term temporal trends. Trending Now looks at two types of trends that can occur: perineal and annual. The perineal videos are those

that have a higher watch rate during certain times of the year, like romance films around Valentine's Day. These videos are commonly watched around the same time each year. Annual videos also refer to those that views spike unpredictably because of an irregular outside force. For example, during COVID many vaccinee documentary clips became more popular. Both perineal and annual clips combined generate the Trending Now row. Many viewers on Netflix enjoy the watching different episodic content and Netflix organizes these clips into the Continue Watching row. This row is generated by many of the shows that people watch, but not all of them. The temporal aperture is taken into consideration and only videos that have been watched within a certain time frame are shown in this category. Clips that user has represented that they are no longer interested in watching have been dropped off the row. The time of day (TOD) that people watch certain shows also depicts how the row is categorized. For example, if a family enjoys watching Netflix cartoons before bed, and history documentaries during the afternoon, then at the respected time the respected show would carry a high weight value. Taking into consideration the TOD requires that different mathematical and statistical models are used to train the system for this specific service. The last row to discuss is Because You Watched (BYW) row. This row is generated by an item-based collaborative filtering approach where a video-to-video similarity algorithm derives these recommendations. This row is completely unpersonalized and calculates the similarity of videos that the viewer watched recently. While the BYW row has no direct attachment to the user, many of these rows are generated, and the rows themselves are computed to see how personalized they relate to the user [10].

The page generation considers that multiple people sign-in into the same profile and aims to provide a diverse selection of rows. Providing a range of selections, Netflix incentivizes users to skip over content that they are not interested in by surrounding the least desired content (or content for another user) with films that the user using Netflix might be interested in watching. The rows chosen for the page layout take into consideration the relevance to the user and well as the diversity to the other content on the page. With, "thousands of rows that could go on one's homepage," a rule-based approach defines which type of row (genre, popular, BYW). No two homepages are required to have the same types of rows [10].

The search algorithm used by Netflix is unique as well because it is common that a user will search for a title that is not found, and again it turns into a recommendation system opportunity. Netflix's search feature use utilized by Netflix not only extracts data from the database, but it also helps generate personalization. Members that frequently search for an actor or genre will be leveraged this content in the future. The search technique is built on several algorithms combining play data, search data, and metadata to create recommendations. One search algorithm attempts to retrieve the exact title, another predicts the topic and creates a selection of films based on that, and a third algorithm uses the letters entered in the search engine to generate a concept the user may be looking for, as seen in figure 8 [10].

Evidence selection is an unique feature that Netflix has incorporated that may go unknown to most users. Evidence refers to all the information displayed for the user before a selection is made, as seen in figure 9. The film cover photo, star rating, preview description, awards received, and percent match are all example of evidence that is visually provided to the user before they select a film. Instead of providing all the same evidence to each member, Netflix uses collaborative filtering to provide unique film marketing to each user [10].

To recap and translate, studying the Netflix recommendation system provides valuable insight to an event recommendation system. While part of the Netflix recommendation system tunes its model for not knowing exactly which user is logging on, many other features of this model can wisely be integrating to a physical event recommendation system. The PVR that generates a row of suggestions based on user preference, genre, and popularity selection, could be translated to a topic of events. For example, if understanding that a user has a personal preference of rock music, then a genre of recommendations could propose a list of live rock music events that are weighted by popularity, while taking into consideration the users' distinct taste. A step further could utilize user-based collaborative filtering and find similar users and suggest bands playing that they enjoy listening to. The Top-N Video Ranker can be viewed as a way to generate a range of suggestions that are all ranked based on the users' preference. Using item-based collaborative filtering, a model could propose a variety of suggestions that are believed to be the highest rated for that user; this could consist of a top band playing, a football game, and a restaurant that has a good discount. The Top-N Ranker would be able to evaluate all events happening in an area and suggest those with the highest rating, similar to the Netflix recommendation system. The Trending Now feature would be able to provide seasonal recommendations such as Mexican food or Hispanic bands around Cinco de Mayo. The Continue Watching feature could be translated to users that have consistent patterns of attending live events; this recommendation system to keep up to date with the live events happening around them. For example, if someone historically enjoys going to watch a local soccer team, then a feature could be incorporated to suggest the games to the user, which would prevent them from having to research and find out every match that is being hosted. The Because You Watched section could be translated to an item-based collaborative filtering recommendation method that finds similarity between local events the user attended. If a user attended a live band one week on Thursday, then the next Thursday similar bands could be suggested. It is seen that the page generation takes into consideration the value of diversity on a recommendation page. This could be utilized to naturally provide a combination of different types of events, such as sports, food, music, and art (based on the users' preference). Netflix's search algorithm would be a useful feature to incorporate in order to provide a range of selection while the user is searching, as well as when the item their searching does not exists. If a user started typing "Me" a range of events could appear, such as Me-n-Ed's (a local restaurant), Mexican food near me (an option to click on to view items that would fall in this category), or Metal bands playing soon (an option to click on to view items that would fall in this category). When an item that does not exist in the database was searched, but information about that is prevalent, similar items could be recommended. For example, if a user searched for Metallica (the Rock band), but no Metallica showings were

listed, then other Rock bands showings could be listed. Lastly, implementing the evidence section would allow for specific features of the events to be highlighted and show cased based on the user. If a user is typically found going out to eat at restaurants during their specials (example: half off wings on Wednesday), then when promoting a live music show, a discounted price could be presented and highlighted.

C. *Amazon Recommendation System [11]*

Putting less weight on trying to understand the uniqueness of each individual, Amazon derives its recommendation attention away from a user-based collaborative filtering approach, and focuses on an item-based approach. For each item that a user purchases, a list of related items is generated. Items that are repeatedly generated from multiple related items lists become candidates for potential recommendation products. The products compiled are then weighted based on how related they are to the visitor's purchase. Relatedness is calculated by the probability a purchase to occur: "item B is related to item A if customers who buy A are unusually likely to buy B as well." Amazon has found that more accurate results when analyzed at the product level rather than the customer level. A relatedness metric based on the products differential probabilities is created and the greater the difference in probability, the more related the two products scaled. Outside of understanding how likely a person would purchase product A, given that they had purchased product B, rose the valuable question of how likely a person would purchase product A, given they made any purchase at all. To increase the probability of product A being purchased, Amazon found that highly discounting the items was a solution, as seen in figure 10 [11].



Figure 10: The deals that Amazon promotes to the user.

A matrix is created to verify if a person has purchased a product. Given they have purchased the product a "1" is placed in the corresponding grid cell to signify this. An autoencoder is used with a small percentage of the entire matrix and predicts an output. If the autoencoder placed a 1 in a cell where a 1 did not exist, meaning the person had not purchased this product, then this is used to signify a potential recommendation. When this was originally created the accuracy on autoencoder was dismal, being beat out by nearly all other recommendation models. What Amazon found was a great way to increase the recommendation accuracy was to limit the temporal factor. Constraining the input timeframe that a user purchased an item to a two-week period greatly increased the overall accuracy of the output. With the temporal constraint adjustment, the autoencoder ended up outperforming the traditional item-based collaborative filtering as well as the bestseller list, which were the two algorithms it was tested against [11].

Autoencoders "utilize Feed-Forward Neural Networks to learn a representation of input with a reduced dimension" [9]. What this means is from a given dataset, a portion of the set is extracted and used for training. The output goal is to reconstruct the network as close as possible to the complete input. Using backpropagation to train the model, the loss (outputs incorrect decision) is refactored during reconstruction phase. The size of the dataset determines how many iterations the algorithm is set to during the training phase. A Collaborative Deep Denoising Autoencoder (CDAE), which Amazon uses, starts by corrupting the input with a mask-out/drop-out approach. The mask-out/drop-out approach refers to discarding randomly selected nodes during each iteration to reduce the weight on the algorithm by only utilizing the neurons for a short period of time [8]. The corrupted input is then mapped to a latent representation. The created mapping is then mapped back to the original input space to reconstruct the input vector. The loss function is then recalculated.

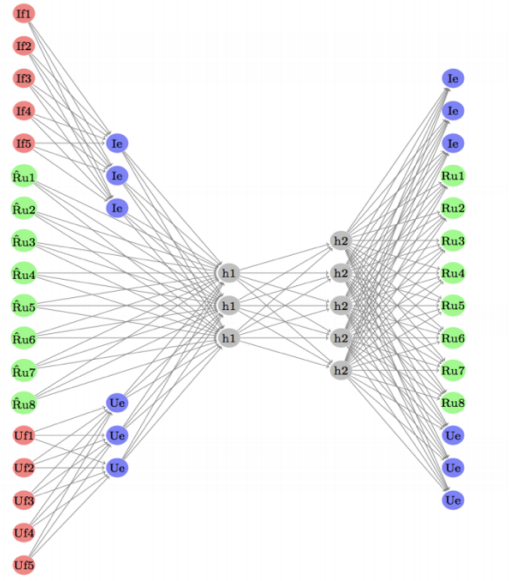


Figure 11: CDAE being used to create recommendations [8].

The Amazon autoencoder recommendation system could prove to be a very useful and accurate model to when implemented to propose event recommendations. While an event recommendation system would not begin with a consumer-to-purchase matrix, a matrix could be centered around if a person had attended an event. A CDAE could be incorporated to decide how likely a person has visited an event. In the case that the system produced a result, in which the model predicted that a person had visited an event in which they did not, then this location could provide to be a recommendation.

D. *Event Recommendation System*

In Section II, *Next Place Prediction*, three active theories were evaluated to gain a better understanding on different ways

to predict where a user would move to next. In an event recommendation system, understanding individualistic behavior and historical preferences enhances the systems accuracy. After evaluating each next place prediction model, the one that is believed to work best with an event recommendation system would be the SVM algorithm utilized in *A Potential Approach for Mobility Prediction Using GPS Data*. Not only did this approach generate the ability to predict where the next user was going with 92% accuracy, but it also incorporated extracting features that would be valuable to understand the type of person going to the location. Gaining personality traits of the individual is an important step in collaborative filtering, as illustrated in this section, which would save the theoretical event recommendation system time and computational resources. Once it is understood where a user would tend to make their next move, then recommendation models utilized by industry leaders could be integrated to figure out other events of potential interest. The Netflix BYW item-based collaborative filtering algorithm could be usefully integrated to decide other similar locations. Once a matrix of similar locations is created, they can be linearly sorted based on weight and presented to the user. Using an algorithm in line with Top-N Video Ranker, a very personalized recommendation row could be generated based on the niche events that a user enjoys attending. Taking into consideration unique user features, such as the average duration they work and the time they spend eating out during a given meal, a user-based collaborative filtering method could be calculated to find others with similar patterns. Once N similar users are categorized, events that those users attend could be evaluated. From the list formed of size N of all similar users, a niche list of size K could be provided that contains users in relative proximity. One recommendation sample could utilize the users in list K and pass their history through autoencoder method, similar to Amazon, to create many recommendations for each of those users. In another application, the events that the users in list K attend could be compared to the inquired user, and the places that the user does not naturally attend could be provided as recommendations. An additional calculation could be configured to decide, which of the users in group K have already visited the same events. From there, the users that are marked in that they have attended the same event can be extracted to form group J. Out of the events that group J attended, which ones score highest on a scale of relativity to the distinct user; the top events here could form a recommendation row.

	E1	E2	E3	E4
U1	1	1	1	0
U2	1	?	1	?
U3	?	?	1	?
U4	?	?	?	?

Figure 12: Matrix to represent users that have attended events and an autoencoder that is predicting whether or not the user has attending the event.

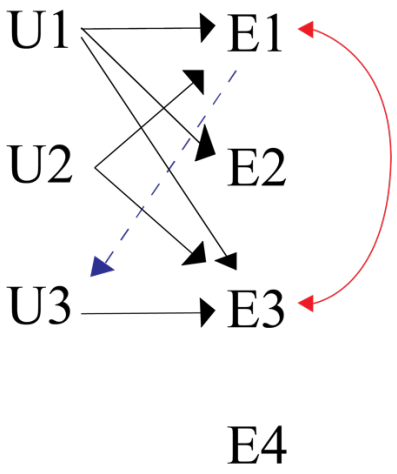


Figure 13: Item-based collaborative filtering approach to show the commonality between E3 and E1, and in turn E1 being recommended to U3.

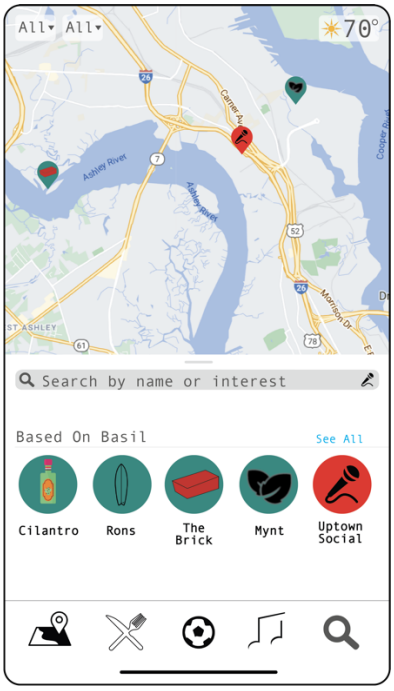


Figure 14: Using the users previous event to make recommendations about future events.

IV. Conclusion

In conclusion, a theoretical prediction-based collaborative filtering event recommendation system (TPECR) could provide valuable options for users. Figures 12 through 14 incorporate a brief overview of the models utilized in creating a TPECR. Assume an SVM model has predicted that user 3 (U3) will go to event 3 (E3) as their next location. U1, U2, U3, and U4 are all selected to be evaluated with one another based on their movement similarity, which can be calculated in the SVM phase. Features to calculate similarity would be based on their movement patterns, such as the time of their movements, the duration of their stay at certain locations, the and the events they attend. Once a niche group of individuals is filtered, an item-based collaborative filtering model will start by creating a matrix calculating places similar to E3. Diagram 13 illustrates that E1 is a similar item-based option in relation to E3. Diagram 12 shows a step in the autoencoder process to predict which places each of the user has attended. These two methods combined will be utilized in tangent to weigh the potential recommendations. Combining all three diagrams together for the representation, E3 will be, "Basil," a restaurant visited by U3, the user to provide recommendations for. U3 went to Basil recently and is highly likely to attend again according to the SVM model, and in turn, a new recommendation model has been created around this. E1 will be, "Cilantro," the top recommendation on both the item-based CF approach, and the autoencoder method. Ranking number one on both of the recommendation system puts it in the first position for the TPCR. Due to U3s' history of going to bars, the evidence provided to entice the event is present with a bottle icon, as seen in Diagram 14.

Using unique features to find similarities between people, and understanding why their decisions are made, allows for the creation of a prediction-based collaborative filtering recommendation system. A TPECR will provide useful insight for users in an area when they are looking to take a step out of their comfort zone and try something new. When a user is looking to stick with their familiar habits, the recommendation system will be able to provide those results, like the Netflix Continue Watching row.

V. Moving Forward

While no real data models were used for study of this project, the next step is to create a database of events in a local area, such as food and beverage deals, sporting events, live music, art shows, etc., as well as real human movement patterns to pass into the algorithms studied. The creation of this feature has the potential to be utilized in many mobile applications and will continue to be worked on.

References

[1] B. T. Nguyen, N. V. Nguyen, N. T. Nguyen, and M. H. Tran, "A potential approach for mobility prediction using GPS data," *2017 Seventh International Conference on Information Science and Technology (ICIST)*, 2017.

[2] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Next place prediction using MOBILITY Markov chains," *Proceedings of the First Workshop on Measurement, Privacy, and Mobility - MPM '12*, 2012.

[3] N. Yang, X. Kong, F. Wang, and P. S. Yu, "When and where: Predicting human movements based on social spatial-temporal events," *Proceedings of the 2014 SIAM International Conference on Data Mining*, 2014.

[4] K. Kruse, "Rule of 7: How social MEDIA CRUSHES old school marketing," *Kruse Control Inc*, 19-May-2021. [Online]. Available: https://www.krusecontrolinc.com/rule-of-7-how-social-media-crushes-old-school-marketing/#:~:text=The%20Marketing%20Rule%20of%207,movie%20industry%20in%20the%201930s. [Accessed: 07-Aug-2021].

[5] D. Chong, "Deep dive into Netflix's recommender system," *Medium*, 02-May-2020. [Online]. Available: https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48. [Accessed: 07-Aug-2021].

[6] "The paradox of choice," *Wikipedia*, 19-Jun-2021. [Online]. Available: https://en.wikipedia.org/wiki/The_Paradox_of_Choice. [Accessed: 07-Aug-2021].

[7] N. T. Blog, "Netflix recommendations: Beyond the 5 STARS (part 2)," *Medium*, 18-Apr-2017. [Online]. Available: https://netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-2-d9b96aa399f5. [Accessed: 07-Aug-2021].

[8] Singla, S., 2021. *Why Dropout is so effective in Deep Neural Network?*. [online] Medium. Available at: <https://towardsdatascience.com/introduction-to-dropout-to-regularize-deep-neural-network-8e9d6b1d4386#:~:text=Dropout%%2020is%20a%20method%20where,neuron%20on%20the%20backward%20pass> [Accessed 7 August 2021].

[9] "Auxiliary stacked denoising AUTOENCODER based collaborative filtering recommendation," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 6, 2020.

[10] C. A. Gomez-Uribe and N. Hunt, "The Netflix recommender system," *ACM Transactions on Management Information Systems*, vol. 6, no. 4, pp. 1–19, 2016.

[11] L. Hardesty, "The history of Amazon's recommendation algorithm," *Amazon Science*, 20-Aug-2020. [Online]. Available: https://www.amazon.science/the-history-of-amazons-recommendation-algorithm. [Accessed: 07-Aug-2021].