**An Unsupervised Approach to Understand Customer Journeys**

Srinija Vobugari, Ankit Anand, Nitin Sahai, Rohit Kata, Matthew A. Lanham

Purdue University Krannert School of Management

svobugar@purdue.edu; anand57@purdue.edu; nsahai@purdue.edu; kata@purdue.edu; lanhamm@purdue.edu

**ABSTRACT**

Online services are increasingly becoming dependent on their visitor participation to understand their customers’ behavior. This study focuses on understanding the customer journeys from the lens of website state-to-state clicks. Our solution provides: (1) a way to segment the customer based on clickstream data and helps understand unique journey patterns taken by a customer group (2) identify highly probable paths that will occur in the future and their traversing patterns that could help streamline marketing efforts or improve the customer experience.  
We developed this solution in collaboration with a healthcare navigation company that has accumulated a massive volume of product searches. The various potential paths to traverse on the web platform led to almost no user having identical journeys. To address this, we developed a model to estimate if two journeys were similar and then grouped all users with similar journeys into their own segment. We devised an approach to encode the customer journey into a pattern of likely sequence strings. To quantify the similarity between two clickstreams, which are sequences of categorical dimensions, we incorporate a Natural language processing (NLP) technique called N-gram similarity as a distance function for clustering. The second step was to implement a density-based clustering algorithm to get K clusters or sub-graphs to segment the customer journeys based on N-gram similarity. This approach has two major limitations: (1) O(n2) computational complexity (2) Manual cluster Interpretation. In this paper, we discuss different approaches that were implemented to cluster the customer event sequences. Each approach has its own associated benefits and limitations. We also discuss some methods to interpret each cluster.

**Keywords**: N-gram, Density-based clustering, Levenshtein Distance, Affinity Propagation Clustering, Markov Chains, Sparse Matrix, Word2Vec

**INTRODUCTION**

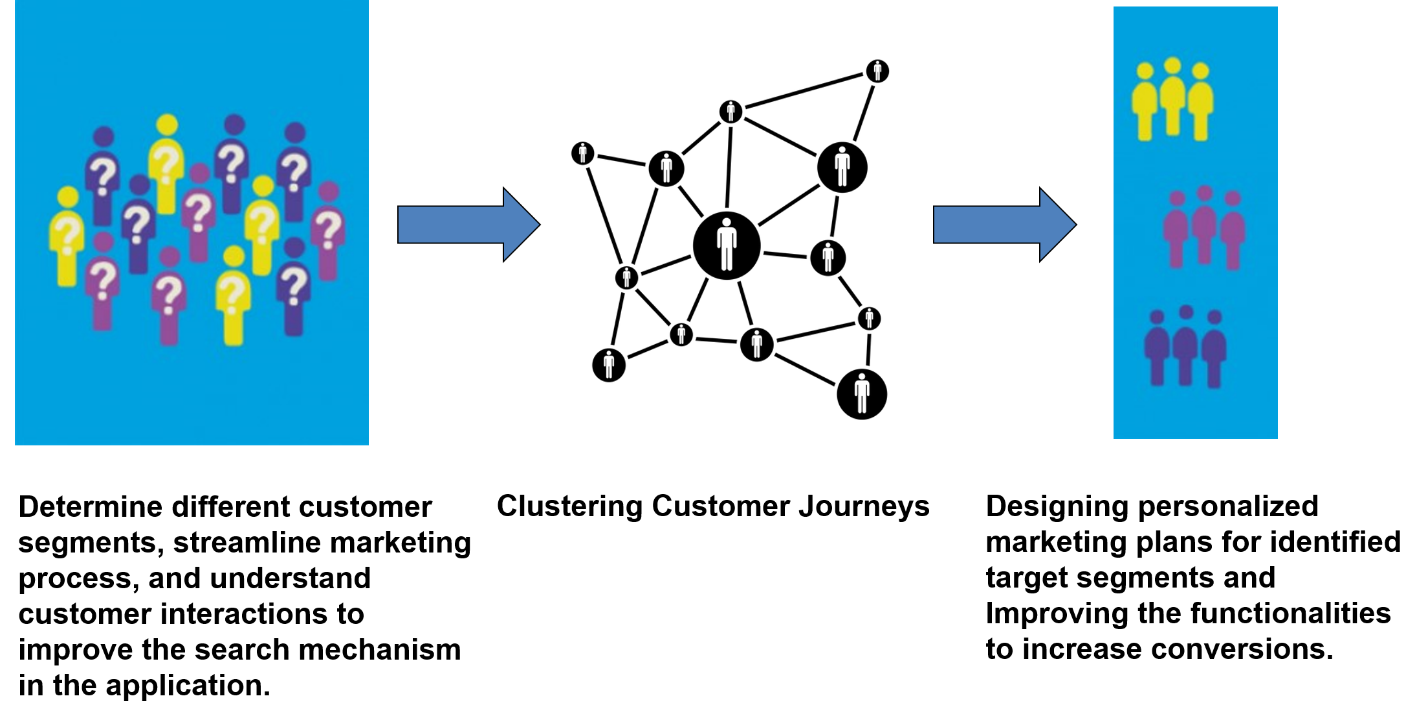
Due to the growing complexity and diversity in user behavior and search patterns on a website/web portal, it has now become very challenging to manage the users and streamline the product features. Knowledge of users’ online behavior can be used and applied in many ways. E-commerce platforms analyze customers web usage data, segment them into a specific group and use it to target their products more efficiently. Application designers are also constantly interested to know how the users interact with the user interface in order to make the right decisions. Analyzing customer journey could help us discover common user journeys which in turn help us improve user experience and maximize profit. Also, customer journey can act as a powerful tool to understand customer behavior and to engage with the customer at a personal level.   
  


Figure 1: How customer behavior segmentation helps businesses

In this paper, we propose a clickstream model to understand user behavior and how they traverse through the functionalities of a platform which provides healthcare tools and connects consumers with healthcare provider.   
The raw consumer events on the platform are processed and have the following information for the analysis:

1. Customer clickstream events: User journey log of 19 unique events (searches, claims, benefits etc.)
2. Search terms: Contained parameters such as search terms, platform, and timestamp
3. Customer demographics: Contained age, location, gender etc. of the users

For the purpose of clustering customer journeys, we chose the sequence of searches by the consumers.

**LITERATURE REVIEW**

There has been a lot of research of sequence clustering in the field of learning gene expression data. [1] One such research has built a frequency matrix each k-gram or a subsequence of the sequences and used the Euclidean distance for graph clustering. There are also a lot of research where they use Markov model in analyzing sequential data and prediction. Mixture of Markov models are used to build customer representative journeys, or in other words representative pattern of each cluster. [2] Earlier research also used clickstream data for studying customer behavior [3] using Whisper and RenRen Datasets. This approach used Iterative Feature Pruning and combined it with graph Clustering. Locality sensitive Hashing (LSH) is another popular method used in Genome-wide association studies and identifying similarities in web search patterns. [4] Implementation of Jaccard similarity using MinHash, this algorithm provides us with a fast approximation to the Jaccard similarity. Then Locality sensitive hashing (LSH) was used to reduce the complexity from the current O(n2). Recent studies have shown that popular Natural language processing (NLP) techniques could be used to cluster the customer journey encoded into sequences. We achieve this using semantic embeddings of the journeys analogous to using unsupervised data of multiple documents of words to discover similarity across words in a corpus using an unlabeled body of text. [5]

**DATA DESCRIPTION**

Data consists of member level, raw user interaction data. The raw data describes 50,000 customers’ click sequences in the healthcare application which is popular for provider search.

**METHODOLOGY**

To solve the business problem described, we came up with five approaches that share a common methodology for cleaning data and mapping customer journeys. Then these five methodologies follow their own distinct paths. Once the clusters are obtained by unsupervised means, we understand and define the customer segments manually. We came up with five different approaches because they try to address the drawbacks of an individual method and can be used based on the business context. If the business context requires high degree of accuracy and if the number of customers to be clustered is relatively low, then the method using N-gram similarity and graph theory is the clear choice. But, this method, by being a pairwise comparison method, is computationally expensive. To overcome this, we devised a second method, a Matrix based K-means clustering method. We can utilize the matrix-based K-means clustering method if the business context can tolerate some information loss, but this method is computationally inexpensive and is ideal for clustering large number of customers. We also use another approach which a based-on mixture of Markov chains is. This is a probabilistic approach which focuses on the transition between events better. The starting point for our approach is to assume that the customers belong to a small number of groups, each having a similar behavior in the sense of having the same probability of transitioning from one event to the next. The benefit of using this method of clustering is that we can using the transition likelihoods to predict the net event for the customer in that cluster. We devised this method to forecast the customer behavior based on their customer journeys.

Firstly, we will focus on the common thread followed by these five approaches. They are almost identical in their approach of data cleaning and mapping customer journeys.

**Data Preprocessing**:

Firstly, we encoded each type of events into a character so that every customer journey is represented as a sequence of characters, such as “Search” as “S”. After coding the 19 distinct events, we concatenate the events into character strings for each customer. We now obtained character strings such as “fsSsfsSsEffcfpfbvbc” for each customer which describes their clickstream. For our analysis, we eliminated customers journeys which consisted of less than 5 steps. After necessary data cleaning and pre-processing, we obtained customer journeys for 45,000 customers.

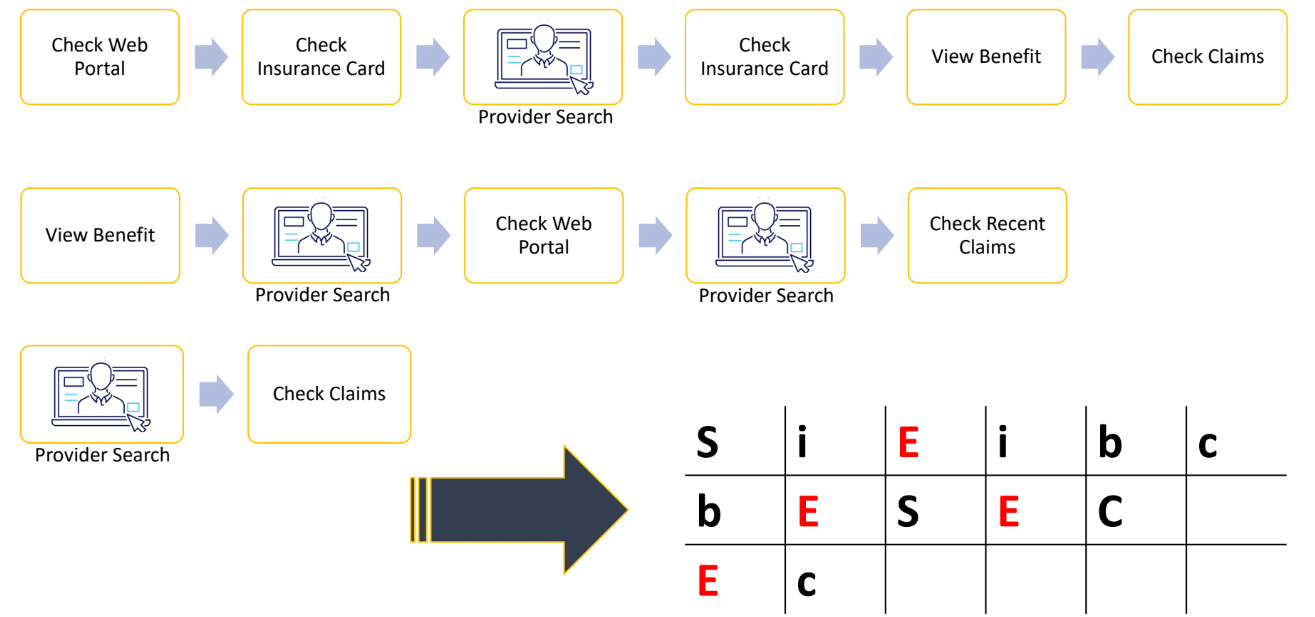
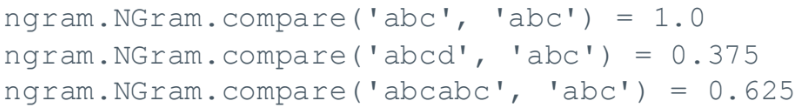


Figure 2: Mapping Customer Journeys

Now, each approach follows its distinctive path for obtaining clusters. Let us first focus on N – gram similarity and graph theory method.

**Approach-1: N – gram similarity and density-based clustering**

**Similarity Measure:** After mapping the customer journeys, we now try to come up with a score to measure similarity between two sequences. We developed a distance function called N-gram similarity function that calculates a similarity score that gauges how similar two journeys are. The N-gram score obtained lies between 0(completely distinct journeys) and 1(identical journeys).



**Identifying clusters:** We utilized the pair-wise similarity scores for all customers in order to group customers with similar journeys. For this, we incorporate graph theory to represent the customers as vertices and the similarity scores as a strength of the connections. By setting a threshold for the score, we see that every customer is connected to the neighborhood customers only if the score is above the threshold. Every customer in a cluster should ideally represent a complete graph or in other words, all the customers within a cluster are connected to each other. As this rarely happens in practice, we consider the clusters in which most of the nodes are connected.

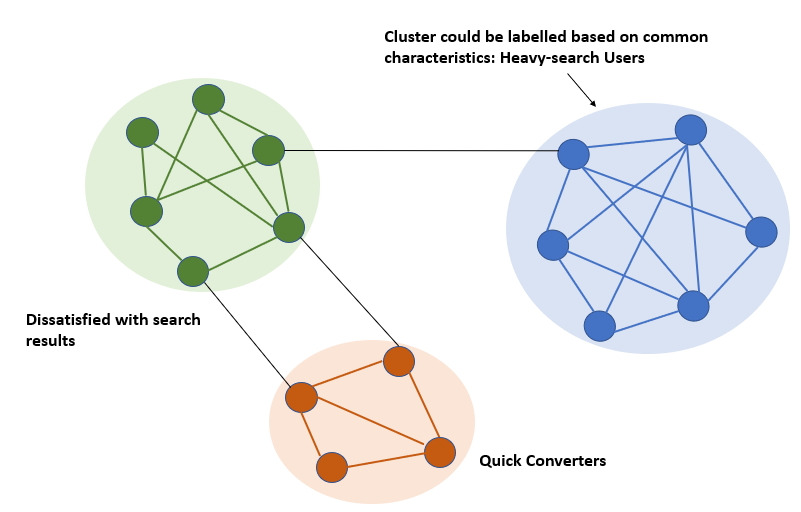


Figure 3: Graph theory-based approach for obtaining clusters

**Approach-2: Sparse Journey Matrix Approach**

In the previous approach, the n gram similarity function is computationally expensive. It is exponential and has an order O(N^2). This limits how many members we can process. The main rationale for Matrix based K means Clustering method is to overcome this limitation. It reduces the computations required by foregoing pairwise comparisons and utilizing K means for clustering. Before K means can be applied, the customer paths have to be captured in a matrix.

This is two-fold.

STEP 1: Data Pre – Processing

We only consider those customers whose event journey is greater than 5 sequence long. Also, we collapse several repeating strings such as “SSSS” to a non-repeating character S. Next, we restrict a customer journey to 30 steps to reduce computation expense further.

STEP 2: Customer Matrix

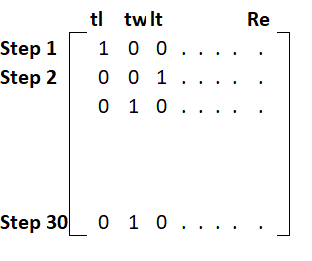


Figure 4: Customer Matrix

We obtain a matrix for each customer as shown in the figure. Since there are 19 unique events, the number of possible events a customer can traverse to in a single step is 380. This is shown in the column headings in the matrix. Customer matrix captures the event sequence of the user in the order they occur.

Let us take an example of a customer with the click path “tltw”. This indicates that the customer traverses from t. This matrix is to capture each of these transitions in steps. Rows in this matrix indicate the individual steps. The column heading such as “tw”, “tf” indicate every possible transition a customer can undertake. In step 1, if customer moves from t🡪l, 1 is placed in the matrix at the intersection of Step 1 and “tw”. Zero is placed in the remaining spots in the row. Similarly, in the second step, if a customer moves from l🡪t, 1 is placed in the matrix at the intersection of Step 2 and “tl”. We continue with this approach until we capture the entire click path of the customer in the Individual Matrix. We follow an identical approach to obtain an Individual Matrix for every customer.

STEP 3: Obtaining Final Matrix to Provide as an Input for K - Means Clustering

After we obtain Individual Matrices for every customer, we combine them to form one single matrix as shown in the figure. Final matrix captures the event sequence of the all the users in the order they occur

In the Final matrix, each row represents the click path of a single user. Thus, we now have the click path journeys of all the customers in a single matrix.

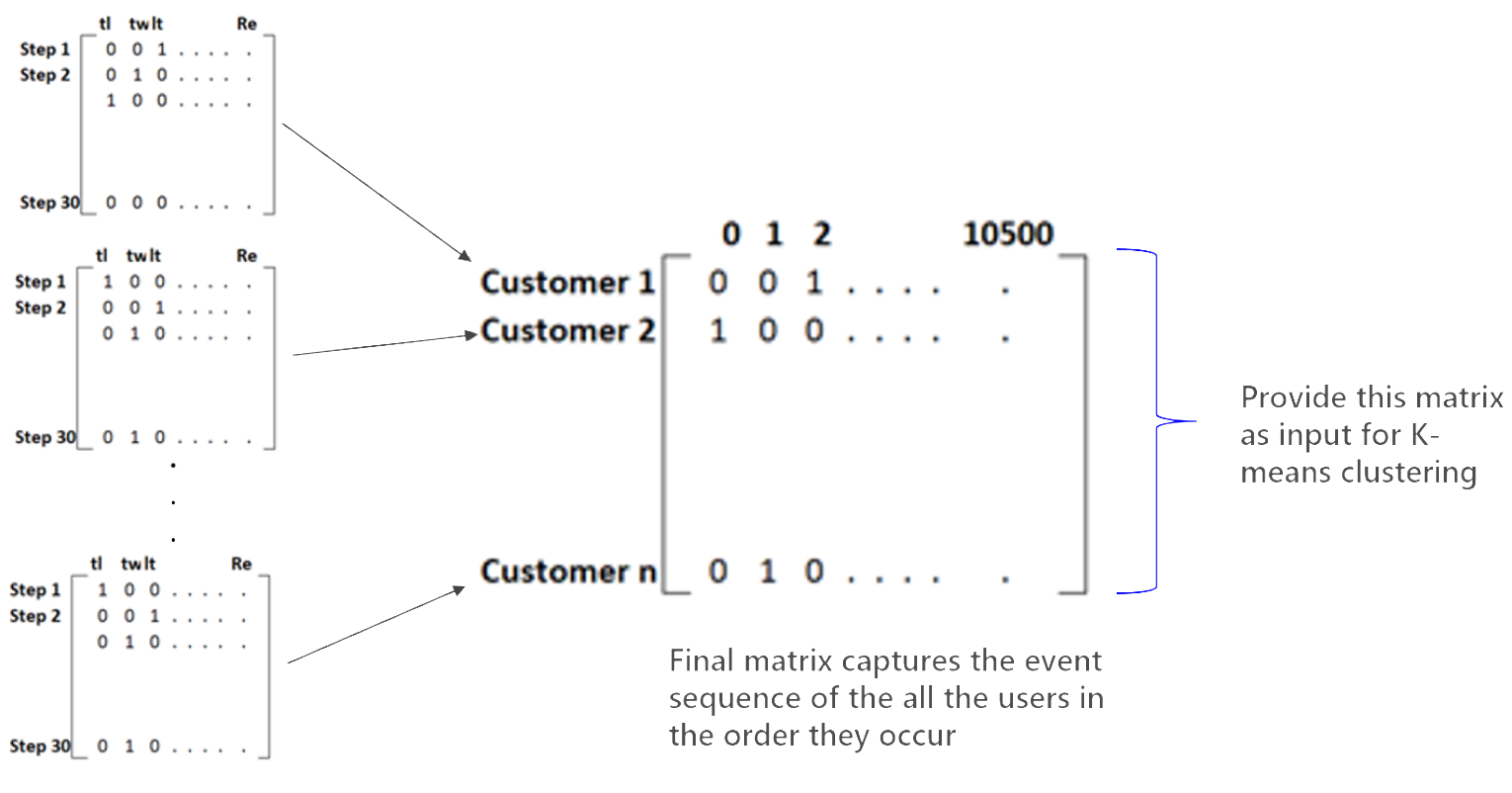


Figure 5: Process to Obtain Final Matrix

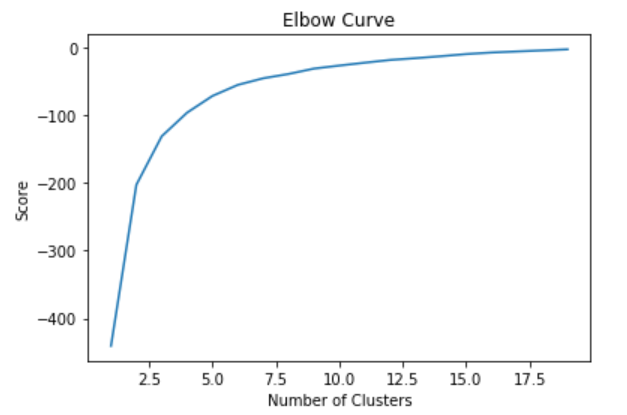
**K-Means to obtain clusters:** Once we obtained the matrix containing the click paths of all the customers, we obtain an elbow plot to determine the ideal number of clusters. After determining the ideal k using elbow plot, we use the customer matrix and ideal k obtained through our elbow plot as the input. K-means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid. We thus obtain k clusters as shown in the figure below. The distribution of customers in 4 clusters is as shown in the figure below.

Figure 6: Elbow Plot

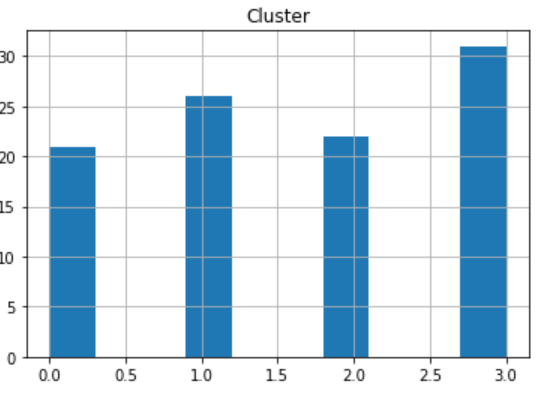
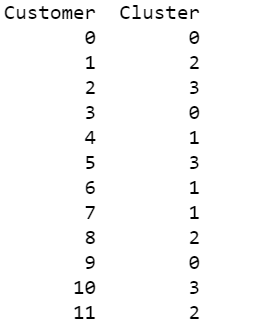
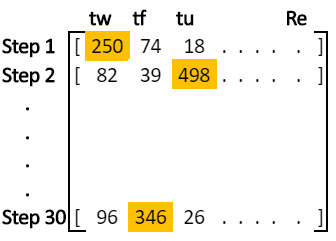


Figure 8: Distribution of Customers

Figure 7: Clusters Obtained

STEP 4: Analyzing the Clusters Obtained using List of Frequencies

****

Once we place customers in their respective clusters, it is important to understand how customers belonging to a particular cluster differ from customers in other clusters. In other words, we find a representative journey to infer the characteristics of that cluster.

The matrix in the figure is a pictorial representation of “List of Frequencies”. The column heading indicates all the possible paths a customer can traverse. The rows indicate the steps. The “List of Frequencies” captures how many customers in cluster 1 have taken a particular path in a given step. For example, from the figure we can deduce that in step 1, 250 customers have taken the path of “tw” ; 74 customers have taken the path of “tf” and so on. The most common path in step 1 is “tw”. Similarly in step 2, 82 customers have taken the path of “tw” and so on. In each step, we consider the most common path such as “tw” in step 1 and “tu” in step 2. We continue to repeat this process till step 30 for each cluster. By this approach, we arrive at a representative journey that gives an indication of the characteristics of that particular cluster.

Figure 9: Obtaining Representative Journey for Cluster 1

**Advantages Of Sparse Matrix Approach:**

* This approach reduces order of complexity from to close to linear
* This can be used to cluster large number of customers
* The clustering process for approximately 40,000 customers takes less than 5 minutes and this model can be scaled up or down

**Limitation Of Sparse Matrix Approach:**

* Clusters are forming based on length of customer journeys because of utilization of K - Means algorithm after obtaining Final Sparse Matrix. This is due to the fact that K – Means utilizes Euclidian distance for the purpose of clustering. In order to overcome this, we utilized cosine distance instead in K- Means
* Customer journeys have to be curtailed to 30 steps to reduce complexity

**Approach-3: Affinity Propagation Model**

We applied Affinity Propagation Clustering approach which provides a representation of the cluster. The similarity between customer event journeys computed using Levenshtein distance which is defined as the minimum number of single-character edits (insertion, deletions and substitutions) required to change one event sequence to the other. Pair-wise similarity matrix created using Levenshtein distance is then provided as input to the Affinity Propagation clustering model. The model randomly chooses representatives customer journey which are termed as exemplars and are chosen among the cluster elements. It operates by considering all the data points as potential exemplars. It then parses message between data points to refine the exemplar choices until a good set of exemplars are achieved. [8] While the output clusters from traditional clustering approaches are hard to interpret, this approach addresses that shortcoming by providing a cluster representation.

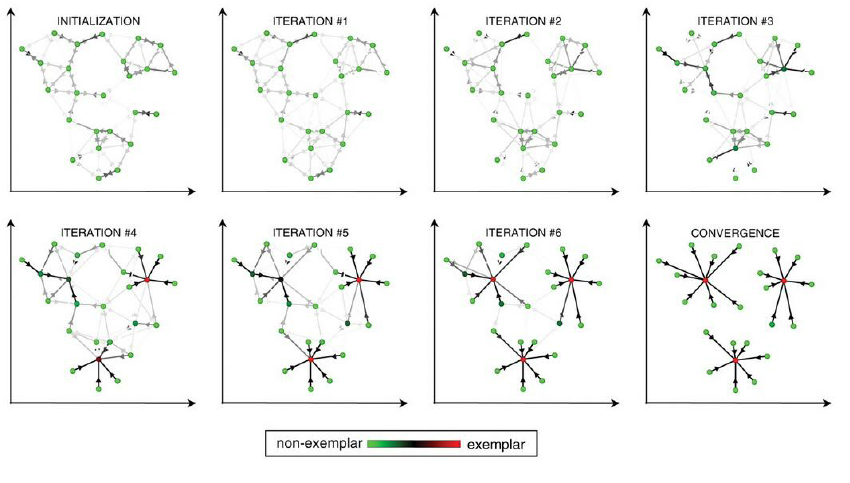


Figure 10: Affinity Propagation Clustering Model Iterations

**Source: https://www.researchgate.net/figure/Illustration-of-how-affinity-propagation-works-Taken-from-42\_fig8\_265969600**

Advantages of Affinity Clustering model:

* Provides representative customer journey making it easier to interpret clusters
* Faster than other clustering methods such as k-means, hierarchical

Disadvantages of Affinity Clustering model:

* Time and space complexity are O(n2) making it difficult to scale
* Number of clusters is determined by the algorithm which can be large

**Approach-4: Mixture of Hidden Markov Models**

We now propose a probabilistic approach that converts event logs into a customer journey map which assumes that each customer belong to a group, each having a similar behavior and the same probability of transitioning from one event to the next. Describing, visualizing, and comparing large sequence data is often complex, especially in the case of multiple channels. Hidden (latent) Markov models (HMMs) can be used to compress and visualize information in such data. [[6]](http://REFERENCES)

Hidden Markov Model for the customer journey detects the underlying latent structure of the journey. In other words, the customer journey which is a sequence of events (observed states) are regarded as probabilistic function of n hidden or latent states. These hidden states cannot be observed or described directly , but they ‘emit’ the observation state on varying probability.

The major advantage of this approach is that this model is trained and analyzed on a probabilistic framework.

Mixture of Hidden Markov Model extends this idea further and presumes that our observations x={x1,x2,…,xN} are generated by a mixture of Hidden Markov models. This approach is similar to clustering feature space using methods like k-means and Gaussian mixtures. Parameter estimation of mixture of Markov models is usually accomplished using Expectation Maximization algorithm. K-means uses Euclidean distances of the points to the cluster centroid in order to decide cluster membership. Analogous to this are the log-likelihoods that define the pairwise distances in the training process of mixture of hidden Markov models. Since this a Expectation Maximization (EM) algorithm which is based on ‘hill-climbing’ the likelihood surface, the results are purely based on the initial training conditions and could be very skewed at times. This approach needs two inputs: Number of mixture components and Number of states. Choosing these metrics involves a heuristic process. There are numerous techniques to find the optimal parameters, but these are beyond the scope of this paper.

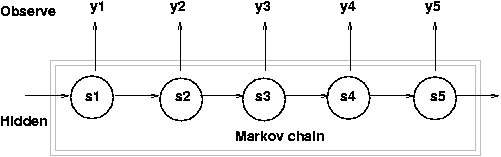
**[[1]](#endnote-2)**

Figure 11: Markov Chain

**Source:** http://personal.psu.edu/jol2/hmm.html

With a wide range of events (19 distinct events) possible in different customer journeys and , the maximum likelihood was taking longer to converge and had a risk of leaving us with local maxima. Markov Models are also great tools to predict the customer’s future path in his journey and also serve as interpretation tools for any cluster. After obtaining the sequence cluster using suitable methods, Markov model can be trained to each cluster and the obtained transition matrix could be utilized to visualize the unique transition patterns of each cluster.

**Approach-5: K-Means on Semantic embeddings of the journeys**

This approach is indirect clustering method of converting the sequences into a feature space and using the traditional K-means approach which has an advantage of less computational complexity than the density-based clustering model based on N-gram similarities. To covert the sequences to a feature space, a popular Natural language processing (NLP) technique was applied. Each sequence is converted into a meaningful vector representation.

This is analogous to the concept of using word embedding or the semantic analysis of sentences to represent them as vectors. By using the well-known bag of words representation, the information regarding the order of the words in a document is lost.

This approach is an unsupervised framework that learns continuously distributed vector representations for pieces of texts or in case the customer journey. There are many popular libraries like Word2Vec and Doc2Vec in Python available to execute this.

Now that the customer journeys are semantically represented in a feature, we now use K-means to cluster these. Using the Silhouette values, we choose K=3 as the optimal number of cluster and run the K-means using the cosine similarity instead of Euclidean distance as it captures similarities and clusters appropriately. The figure below shows the cluster frequencies obtained for 10,000 customer journeys. From looking at some random sequences in each cluster and judgement using the context, we observe that the results are well-defined.

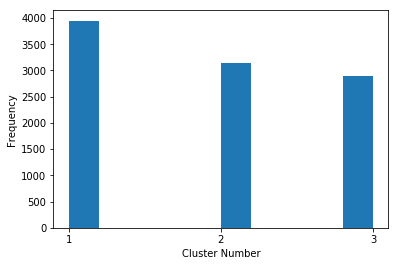


Figure 12: Obtained clusters with count of elements

The major advantages of this approach is that it can take variable lengths of sequences and also used K-means algorithm which is relatively less computationally expensive.

**RESULTS**

Each of the methodologies discussed serve a unique purpose and have a distinctive business implication. We observe that though the mixture of Hidden Markov Model had a good foundation which would follow a probabilistic approach to cluster the customer journeys, it has two limitations. This approach was hence unsuitable for clustering these numerous customer journeys giving us skewed results. The N – gram similarity and density-based clustering though computationally expensive, is very accurate and is ideal for clustering a population of a smaller size. To overcome the issue of computation, we utilized Sparse Matrix K- Means approach. This method is ideal for clustering a large population as it is computationally inexpensive. But, to understand the clusters obtained by this unsupervised method, we had to device ways such as Frequency List. To improve upon the accuracy of our understanding of clusters obtained, we implemented Affinity Propagation Clustering. This approach provides us with a representative customer journey that is an apt representation of the characteristic of that particular cluster. In other words, we obtain a persona for the clusters generated. We devised Markov approach to provide additional level of functionality of predicting the next steps of customers depending on the clusters they belong to. This method also helps in visualizing the clusters. Thus, each of these approaches have their own unique functionalities and advantages. Together, they provide a complete solution to a company to help them understand their customers better by clustering them into segments.

Now that we have obtained our clusters by using the approaches described above, we understood and defined the customer segments. Using the representative journeys obtained from the Affinity Propagation approach, we labeled customer paths from each cluster to obtain the salient characters of each cluster. Based on customer’s click path, we segmented them into 5 clusters. We identified a unique behavioral trait for each cluster obtained. This serves the dual purpose of understanding the customers better and the interaction of customers with the application’s User Interface. The characteristics of the customers are shown in the figure.

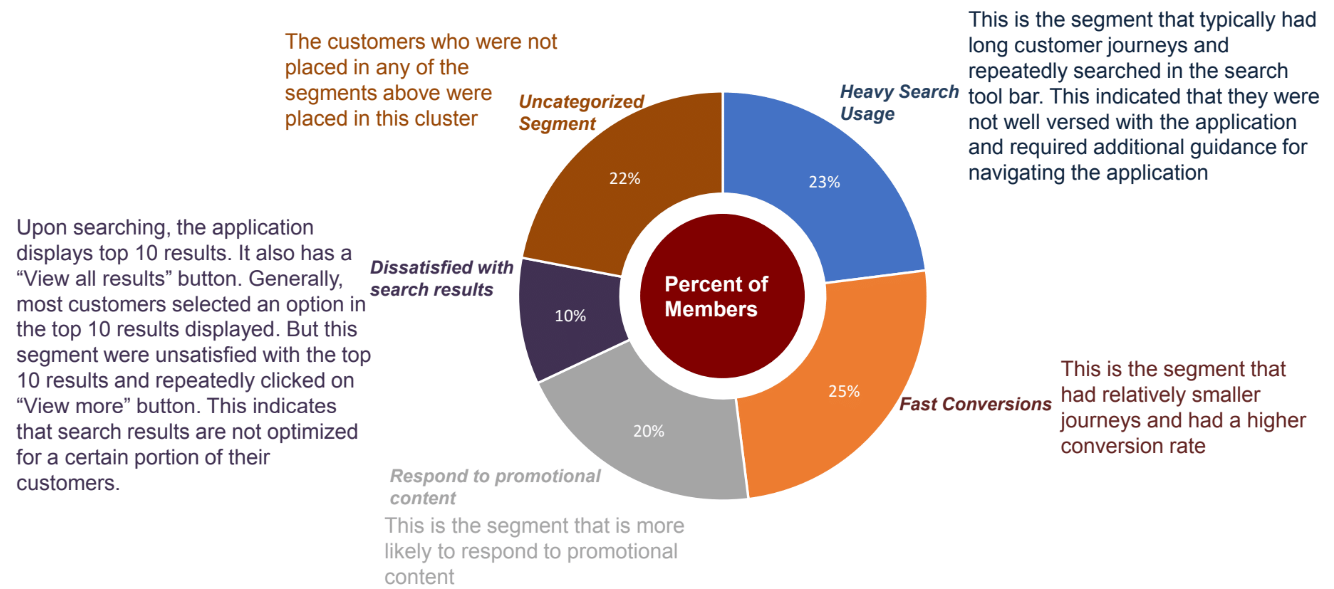


Figure 12: Identified clusters

**CONCLUSION**

This study discusses a novel approach to understand user behavior using the clickstream data of a platform. Clustering methods for clickstream data was discussed and these methods can be extended to all areas of study which works with sequential categorical data. They can be widely used in different applications of clickstream analysis and behavior modelling. This study could be taken further to incorporate the time factor of the events for a better understanding of journey similarities.

### REFERENCES

1) Daxin J., Chun T., Aidong Z. , *Cluster Analysis for Gene Expression Data: A Survey* <https://cse.buffalo.edu/DBGROUP/bioinformatics/papers/survey.pdf>

2) Matthieu H.,, Gaël B., Pietro B., Benoît G., Periklis A., *Discovering Customer Journey Maps using a Mixture of Markov Models* <http://ceur-ws.org/Vol-2016/paper1.pdf>

3) Gang W., Xinyi Z., Shiliang T., Haitao Z., Ben Y. Z., *Clickstream User Behavior Models*

4) Aristides G., Piotr I., Rajeev M., *Similarity Search in High Dimensions via Hashing* <https://www.cs.princeton.edu/courses/archive/spring13/cos598C/Gionis.pdf>  
5) Nishan S., (2018,July 12) “*Modeling User Journeys via Semantic Embeddings*”

<https://codeascraft.com/2018/07/12/modeling-user-journey-via-semantic-embeddings/>

6) Satu H.,Jouni H., *Mixture Hidden Markov Models for Sequence Data: The seqHMM Package in R*

7) Padhraic S., *Clustering Sequences with Hidden Markov Models*

8) Brendan J. Frey and Delbert Dueck, *Clustering by Passing Messages Between Data Points* <https://www.psi.toronto.edu/affinitypropagation/FreyDueckScience07.pdf>

9) Jeff Ullman, *Mining of Massive Datasets* <http://infolab.stanford.edu/~ullman/mmds/ch3n.pdf>

*10*) Aristides Gionis, Piotr Indyky, Rajeev Motwaniz, *Similarity Search in High Dimensions via Hashing* <https://www.cs.princeton.edu/courses/archive/spring13/cos598C/Gionis.pdf>

11) Ryan McConville, Xin Cao, Weiru Liu, Paul Miller, *Accelerating Large Scale Centroid-Based Clustering with Locality Sensitive Hashing* <http://www.cse.unsw.edu.au/~z3515164/publications/icde16.pdf>

1. [↑](#endnote-ref-2)