Network-Wide Bike Availability Clustering Using the College Admission Algorithm: A Case Study of San Francisco Bay Area

Mohammed H. Almannaa[1,3] Mohammed Elhenawy[2,3], Ahmed Ghanem[2,3], Huthaifa I. Ashqar[1,3] and Hesham A. Rakha[1,2,3]
[1] Charles E. Via, Jr. Dept. of Civil and Environmental Engineering
[2] Bradley Dept. of Electrical and Computer Engineering
[3] Center for Sustainable Mobility at the Virginia Tech Transportation Institute
Blacksburg, VA 24061
{almannaa, mohame1,aghanem, hiashqar, hrakha}@vt.edu

Abstract—**The significant increase in the use of bike sharing systems (BSSs) causes imbalances in the distribution of bikes, creating logistical challenges and discouraging bike riders who find it difficult to pick up or drop off a bike at their desired location. We investigated this issue by finding the network-wide availability patterns and how these patterns evolve temporally using a novel supervised clustering algorithm based on the College Admission and the K-median algorithms. The proposed approach models the clustering problem as a matching problem between two disjoint sets of agents: centroids and data points. This new view of the clustering problem makes our algorithm a multi-objective algorithm where the impurity and distance in each cluster are minimized simultaneously. The proposed algorithm showed promising performance when applied to BSS data for the San Francisco Bay area. The resultant network-wide availability patterns were used to identify imbalances in the BSS. Using a spatial analysis of these imbalances, we propose potential solutions for decision makers and agencies to improve BSS operations and make it more stable.**

Keywords— clustering; bike-sharing systems; urban computing

I. INTRODUCTION

With the rapid world-wide population growth, large, dense cities are struggling with traffic congestion. Many people have migrated from rural to urban areas, creating highly crowded cities with limited resources. Traffic jams are one of the critical issues that urbanized areas are suffering from. A number of potential solutions have been proposed to mitigate the negative impact of this phenomenon and improve private and public transportation. One solution is a bike sharing system (BSS), where residents and visitors in urban areas can ride from one bike station to another for free or for a very low rental rate, making the system accessible to every person. BSSs are cost-effective for users and cities. The BSS idea started over five decades ago in Europe, and then bloomed in 50 countries, with more than 37,000 stations in 2000 [1]. This growth demonstrates the significant transportation benefits obtained by implementing a BSS, which are further enhanced with the use of advanced technology. For example, bike riders are able to borrow a bike from any bike-sharing station using a smart card and then return it to a bike station near their destination. BSSs are considered a sustainable transportation mode, especially with last-mile trips, and help to reduce congestion, emissions, and pollution.

The significant increase in the use of BSSs has raised the issue of imbalance in the distribution of bikes. Some stations are at capacity and others nearby are empty. This issue creates logistical challenges for the BSS and may discourage bike riders who find it difficult to pick up or drop off a bike. To address the problem, recent research has been conducted on re-balancing the distribution of bikes at stations [2] [3]. Some researchers use a statistical model to predict bike availability at each station, while others use clustering algorithms, such as traditional and non-traditional clustering [4]. Traditional clustering approaches, such as the K-median, DBSCAN, and Fuzzy algorithms, are good tools for clustering bike-sharing stations based on their status, but give narrow results as clusters are based on only one factor (i.e., distance). This approach is considered to be a type of unsupervised clustering that divides the observational points into clusters with respect to distance only. The unsupervised clustering approach has only one objective function, which is minimizing the distance, so it does not consider other attributes in the data set, such as the time of events (i.e., day-of-the-week and hour-of-the-day).

Recently, supervised and semi-supervised clustering (non-traditional) approaches have been widely discussed as powerful tools that can take advantage of other attributes (labels) in the data set [5-8]. The supervised algorithms try to create clusters using all the labeled data in the set, while the semi-supervised algorithms use only some of the labeled data points. The other distinction between semi-supervised and supervised clustering is how the objective function is implemented. In supervised clustering, data points are clustered with the goal of only minimizing the impurity of clusters. In semi-supervised clustering, clusters minimize impurity and distance (i.e., dispersion in the cluster) [9].

In this paper, we take a global view of network-wide bike availability across stations using a semi-supervised clustering approach. To do so, we developed a novel supervised clustering algorithm that is built using two well-known algorithms, namely the Gale-Shapley student-optimal college admission (GSSOCA) algorithm [10] and the K-median algorithm. This approach combines the advantages of both supervised and unsupervised algorithms. Consequently. it is a multi-objective algorithm

978-1-5090-6484-7/17/$31.00 ©2017 IEEE

where the impurity and distance in the cluster are minimized simultaneously. The proposed algorithm was applied to a BSS data set for 2 years (2013–2015) in the San Francisco Bay Area. The results show that the developed algorithm performs very well in clustering the data to day-of-the-week and hour-of-the-day and providing an expected bike availability pattern at each station. Our analysis shows that some of the low-demand stations are located next to high-demand stations, making it possible to balance the system with low cost and effort.

This paper is organized as follows. Section 2 describes the related work on modeling BSSs. Section 3 gives the tools used to develop the algorithm, followed by the proposed algorithm. Section 4 provides the results and findings, and finally Section 5 presents the conclusions of this research effort.

II. Related work

Given the size and complexity of the BSS data, various researchers have attempted to derive insights by exploring trends through visualization techniques [6] [7] [8, 9]. For example, Froehlich et al. studied BSS patterns using 13 weeks of bicycle station usage in Barcelona [11]. They investigated the relationship between human behavior, geography, and time-of-day, and then tried to predict future bicycling station usage. The temporal and spatiotemporal patterns were discussed, and the results show that there were some dependencies among the stations. The available bicycling data were used to cluster the docking stations. Neighboring stations were found to be highly correlated and therefore clustered in one group. Additionally, stations located on the boundaries of the bike-sharing network were found to be less active. Kaltenbrunner et al. also attempted to improve the BSS in Barcelona using docking station data [12]. Temporal and geographic mobility patterns were obtained and analyzed with the goal of detecting imbalances in the BSS. Subsequently, they used time series analysis techniques to predict the number of bicycles at a given station and time.

Vogel et al. attempted to derive bike activity patterns by analyzing bike share data along with geographical data [13]. Cluster analysis was used to group the bike stations with respect to pickup and return activity. They used K-means, expectation maximization, & sequential information-bottleneck algorithms to conduct their analysis. Using the temporal activities of the stations, their results show that the bike stations could be clustered into five groups, and, thereby, average pickup and return for each hour was given for each group. After that, the authors tried to link these five clusters with geographical information data, and found that stations in the same cluster tend to be neighbors. However, this study only clustered the bike stations based on time-of-day. They did not consider other temporal factors such as day-of-the-week, month-of-the-year, or season. Also, they used three clustering approaches, all of which clustered the bike stations based only on one objective function, the number of bikes picked up and returned at each station.

Recently, supervised clustering has been widely introduced in data mining as a powerful algorithm that can classify data sets effectively. It is a novel approach that enhances a clustering algorithm by using classified examples (training data) and tries to identify clusters that have high probability density with respect to a single class [5]. In [14, 15] several algorithms were developed using background knowledge. However, most of the proposed algorithms follow a sophisticated approach with too many parameters. Additionally, these algorithms try to achieve only one objective function, namely, increasing purity.

In this paper, we propose a new supervised clustering algorithm that has the ability to increase the purity of the cluster and the similarity of its members considering only one parameter, namely; the number of clusters. The proposed algorithm compromises between distortion and purity in identifying clusters within the data. We applied our algorithm to a BSS in the San Francisco Bay Area and then grouped the data into different clusters with respect to time-of-day and day-of-the-week. Consequently, we predicted the number of available bikes at each station and, thereafter, addressed when and where the system would be imbalanced.

III. Methods

In this section, we briefly present the methods that were used in the proposed model, followed by the proposed algorithm.

A. K-median

K-median is a fundamental clustering algorithm in which the observation points (X_i) are clustered based on their distance [16]. Given a number of clusters (K), the K-median algorithm tries to classify the observation points to the k-nearest cluster (m_x) with the objective of minimizing the dispersion within the cluster (E). Dispersion denotes the summation of distances from all points in the cluster to the K centroid (1). At each iteration, the K-median algorithm updates the centroid of each cluster and reassigns all the observation points to the clusters. Eventually, the K-median algorithm converges when the centroids of the clusters stop moving.

$$E = \sum_{K=1}^{K} \sum_{X \in C_K} \|x - m_x\| \qquad (1)$$

where E is the distortion needed to be minimized, X is the observation point, and m_x is the centroid of cluster C_K.

The K-median algorithm is very similar to the K-mean algorithm, where the only key difference is in the way the distance is calculated. In K-median, the centroid is computed using Manhattan distance and the median of the cluster, while the K-mean uses Euclidean distance and the mean of the cluster. In this paper, the K-median was utilized in the proposed algorithm instead of K-means because it uses integer numbers as centroids that represent the bike availability at each station. The K-median algorithm needs only one parameter, the number of clusters (K). Therefore, we tested the proposed algorithm with a varying number of clusters from 2 to 30.

B. The College Admission (CA) Algorithm

In 1962, Gale and Shapley proposed a solution to the well-known Stable Marriage problem, in which an equal number of men and women are matched such that no player has an incentive to leave his/her matched partner [10]. The stable marriage problem involves one-to-one matching. The CA problem is another version of the stable marriage problem, though in this case the algorithm matches many to one. In the CA problem, there are a number of colleges and applicants that need to be matched. Each college has a ranked list of students they prefer, and each student has a ranked list of colleges they prefer. The size of the ranked list of students depends on the capacity of the college. The best-qualified candidates are offered admission first, followed by the lesser-qualified candidates. On the students' side, the ranked list of colleges is based mainly on their preference.

This problem includes the uncertainty of the colleges not knowing which other colleges the students have applied to, and thus not knowing the ranked list of each student, or whether the student has been offered admission by other colleges. Consequently, the colleges are in a blind position with very little information, which prevents them from making the appropriate decision. This can result in an unbalanced situation in which some students are offered many admissions, while others are not offered any at all. Gale and Shapley presented a stable solution that is close to the optimal condition. Eventually, each student would be accepted to the best possible college with regard to his or her list, and each college would have the best possible qualified student.

The CA algorithm finds the optimal solution through a series of iterations. At each iteration, the colleges offer admission to the best-qualified students, and the students have to reply back by either accepting the offer or not. At the end of the iteration, some students have an admission and others do not. Colleges then update their list accordingly in the next iteration and offer admission to students who did not receive an offer before, regardless of whether they have an admission or not. The students' lists do not change, but students can change their decision at each iteration if they are offered admission to a better college. The algorithm keeps iterating until reaching the best possible solution.

C. *Proposed Algorithm*

Knowing some similarities in the data set is a great advantage to clustering algorithms. It can efficiently and effectively advance the outcome of the algorithm, and create meaningful clusters. Accordingly, we developed a novel supervised clustering algorithm that is built using a combination of the CA and k-median algorithms. The proposed algorithm models the clustering problem as a cooperative game where two disjointed sets of players join the game to identify the a stable match. The first players' set consists of the centroids (clusters) and the second players' set consists of the data examples (data points). Each centroid orders the data points in its preference list based on the distance from the centroid to the data point. On the other side, each data point orders the centroids in its preference list based on the purity. For example, a data point that has label h will give preference to the centroid that has the highest ratio of members with label h. In other words, a data point gives higher preference to centroids with the majority of its members have the same label as its own label. Similarity here means that the points have the same value in the known attribute. Through a series of iterations, the proposed algorithm will try to match between the clusters (minimizing distances) and data points (maximizing average purity) until it converges. It should be noted here that a cluster purity is the number of objects of the largest class in this cluster divided by size of the cluster (Equation.2). The algorithm terminates when the objective function no longer improves (threshold is less than 0.001), which involves minimizing the distance and maximizing the purity as follows:

$$purity(c_i)^t = \frac{n_i^m}{n_i} \tag{2}$$

$$\mathrm{q}^t = \frac{1}{k}\sum_{i\in\{1,\dots,k\}} purity(c_i)^t + \sum_{i=1}^{k}\sum_{x_j\in c_i} 1/d(x_j, c_i) \tag{3}$$

$$The\ stopping\ criteria\ is\ \frac{\mathrm{q}^{t+1}-\mathrm{q}^t}{\mathrm{q}^t} < 0.001 \tag{4}$$

Where n_i is the number of objects in cluster i, $i \in \{1, \dots, k\}$, n_i^m is the number of the largest class (m) in cluster i, q^t represents the objective function at iteration t, c_i is the centroid of cluster i, $i \in \{1, \dots, k\}$, $j \in \{1, \dots, N\}$ and N is the number of data points.

The following is a brief description of the proposed algorithm assuming the model order k is known:

1. Randomly choose k points as the initial centroids.
2. Form k clusters by assigning all points to the closest centroid using $L1$ norm.
3. Re-compute the centroid of each cluster by computing the median.
4. Find the size of each cluster.
5. Each centroid creates its preference list of points based on $L1$ norm distance.
6. Each point creates its preference list based on purity
7. Find the best match using the CA algorithm.
8. Re-compute the centroid of each cluster based on the outcome of matching using the median.
9. Calculate q^t.
10. Find the size of each cluster using K-median clustering results.
11. While the stopping criteria is not satisfied, repeat steps 5-11.

To illustrate this algorithm, let us assume we have *n* data points and want to group them into three clusters as shown in Fig. 1. All the data points have only one known parameter (besides their coordinates) that can be used to add supervision to our algorithm in the second step. This parameter is assumed to be the true label.

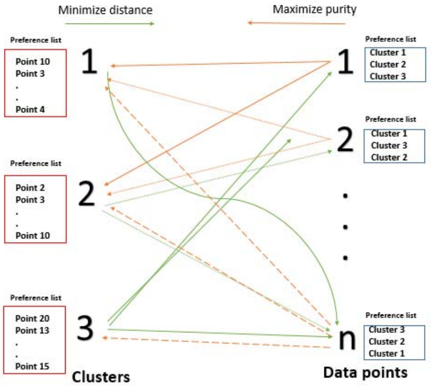

Fig. 1. Illustrative example.

To illustrate this algorithm, let us assume we have N data points and their labels are known. These labels could be any observed labels, such as the day-of-the-week. Moreover, we assume that the true number of clusters is three. The question we want to answer is how to partition the N data points such that similar data points in terms of distance and true labels are grouped together. By effectively partitioning the N data points, we can answer questions such as which days of the week have similar bike availability across the network.

In the First step, the proposed algorithm will first randomly choose three points as centroids for the three clusters. Then, it will partition the data points based on distance to get an estimate of the size of each cluster and its purity. After that, each data point builds its preference list and each centroid builds its preference list, as shown in Figure 1.

In the second step, the proposed algorithm, through a series of iterations, will try to find matches between clusters and data points and provide a stable match. At the end of this iteration, all points should be matched with one of the three clusters.

After successfully matching the point with clusters, the centroid of the three clusters will be recalculated, and a new cluster size will be created accordingly we re-partition the N data points based on L1 norm to find the cluster size for the next iteration. The algorithm will repeat the entire process of building new preference lists, matching, and calculating new centroids. The algorithm will stop when there is no significant improvement in objective function.

IV. Data Set: Case Study of San Franscisco

This study used docking station data collected from August 2013 to August 2015 in the San Francisco Bay Area. The docking station data include station ID, number of bikes available, number of docks available, and time of recording. The time data include year, month, day-of-the-month, day-of-the-week, hour and minutes at which the docking station data was recorded. As the station data was documented every minute for 70 stations in San Francisco over two years, the data set contains a large number of recorded station data. Consequently, the size of the data set was reduced by sampling station data once at every quarter-hour instead of once every 1 minute (e.g. 8:00 am, 8:15 am, 8:30 am, 8:45 am … etc) and obtaining the exact values without any smoothing process. This was done to reduce the complexity of the data and take a global view of bike availability in the entire network every 15 minutes, with the goal of finding the similarity between these views and clustering them with regard to similarity and recorded time. Similarity refers to the bike availability in all stations, while the recorded time refers to the day-of-the-week and hour-of-the-day. We discarded other time attributes such as day-of-the-month, year, and minutes in the analysis as they might not have a significant impact on bike availability.

During the data processing phase, we found that numerous stations were recently added to the network and others were terminated. Consequently, the data set was cleaned by eliminating any entries missing docking station data. This reduced the number of entries from approximately 70,000 to 48,000. Each entry included the availability of bikes at the 70 stations with the associated time (day-of-the-week and hour-of-the-day). The availability of bikes represents the coordination measure for each entry, which is used in the K-median method to determine the entry closeness measure. This resulted in each entry constituting 70 dimensions (70 stations).

V. Clustering Results and Discussion

The data set (15-min entries) was clustered using the proposed algorithm with regard to two labels (day-of-the-week and hour-of-the-day). The proposed algorithm was run with 30 seeds and a K value ranging from 2 to 30. The stopping criterion was the combination of purity and distortion or a maximum of 100 iterations. Before presenting the results obtained for each label, it is important to discuss the methodology we used when choosing the size of the cluster for each label.

A. Selecting the Number of Clusters

A varying number of clusters, ranging from 2 to 30, were considered in selecting the optimal number of clusters (K) for each label. Recently, numerous cluster ensemble methods have been proposed to assist in finding the optimal K in which the result of a specific K is stable across all seeds. One is consensus clustering (CC), which subsamples the data set and calculates the consensus rate between all pairs of samples [17]. The result of this method is a similarity matrix that identifies the number of times two data points are assigned to the same cluster centroid, $k \in K$, that can be used to show the degree of stability for each K. One of the measures for CC that can show the cluster stability is the cumulative distribution function (CDF) against consensus rate. In the CC method, every curve represents a K, and the more the curve is flat, the more stable the number of clusters K is.

In our algorithm, we used the aforementioned method to determine the best K for each label. The hour label was tested for a varying value of K, and a CDF against consensus rate was drawn, as can be seen in Fig. 2. Each curve represents a K and can be seen that the curve of K=2 is flatter compared to the other curves; indicating that K=2 is the optimal K for the hour-of-the-day label. Consequently, we analyzed the data in more detail for K=2 for the hour-of-the-day label. Similarly, we applied this technique to the day-of-the-week label, as shown in Fig. 3, and we found that the optimal K is 3. Results for each label are discussed in detail in the following sections.

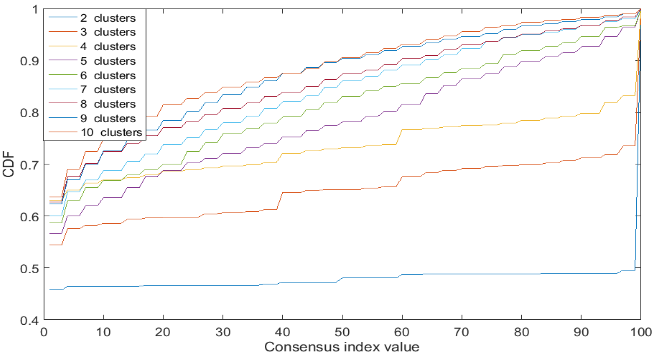

Fig. 2. Cumulative distribution function against consensus index value for each cluster (hours).

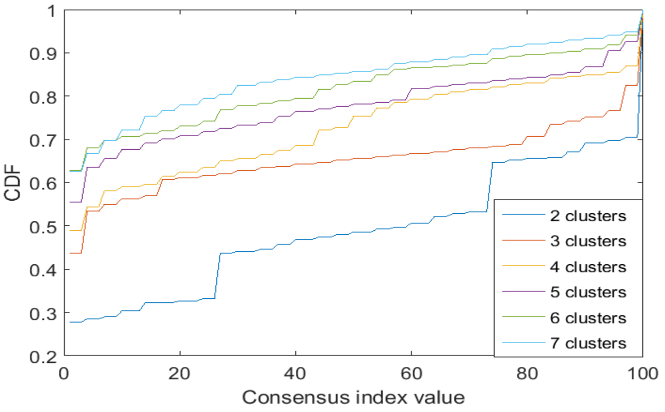

Fig. 3. Cumulative distribution function against consensus index value for each cluster (day of week).

B. *Day-of-the-Week*

After determining the optimal number of clusters (*K*=3) using CC, we explored the cluster factor. The results of the three clusters are presented in Fig. 4, in which each day takes a probability of being in one of the three clusters. The three clusters are dominated by specific days: Saturdays and Sundays are clustered in the first cluster; Mondays and Fridays are clustered in the second cluster; and finally Tuesday, Wednesday, and Thursday are clustered in the third cluster. This pattern differs from previous research [12] that showed bike patterns grouped into two clusters (weekend and weekdays). Our research shows that the weekdays can be split into groups: (a) Mondays and Fridays, and (b) Tuesdays, Wednesdays, and Thursdays. This appears to be logical as the beginning and the end of the week are different from the rest of the weekdays.

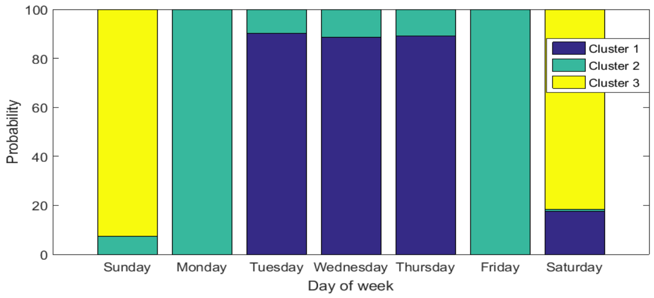

Fig. 4. The probability of the day of week to be in one of the three clusters (k=3).

Each cluster is associated with a pattern for the availability of bikes for each station. The results for the three clusters are provided in Fig. 5, in which the capacity of each station is added to the three patterns along with the patterns of the clusters. It should be noted that the pattern represents the centroid of the cluster (the median of the cluster) so that the exact values of the pattern are not shown here. The pattern can serve as an indication of when the station is more likely to be full or empty on any specific day-of-the-week. Four observations can be drawn from Fig. 5. First, the three patterns of the three clusters generally follow the pattern of the capacity of the stations, which could be the result of the rebalancing efforts performed by the system operators. Second, the pattern of the three clusters shows fluctuations in the bike activities. None of the days of the week has the highest activity for the entire network, which depends on both spatial and temporal factors. Third, several stations appear to be more likely empty or full on either weekdays or weekends. The difference in demand between the three clusters appears clearly for some stations, but not others. For example, station 31 (Cowper at University), located in Palo Alto, and station 41 (Embarcadero at Folsom), located in San Francisco, are rarely used by bikers, especially on Tuesdays, Wednesdays, and Thursdays (i.e., bikers might not be able to drop their bikes at these stations on these days as they are almost full). In contrast, station 36 (Washington at Kearney) and station 39 (Spear at Folsom) are highly used by bikers, and thus are more likely to be empty. Consequently, it might be wise to consider either relocating or changing the size of these stations. Surprisingly, when analyzing the location of station 41, we found that station 39 (Spear at Folsom), which is highly used by bikers, is very close (only one block away); thus, it might be better to move bikes between them overnight, especially on Tuesday, Wednesday, and Thursday, as the largest differences in usage occur on these days. Another suggestion is to incentivize bikers to drop off their bikes at station 39 instead of station 41 or pick up their bikes from station 41 instead of station 39. Fourth, the three clusters have the same pattern for stations 6 to 15, 67, and 70. In other words, all the days of the week have the same bike activities. When examining their locations, we found that all of these stations are located in San Jose, so we can imply that the bike activities in San Jose are approximately similar during the entire week. That could be caused by the fact that both tourists and workers use the bikes for their trips, or that the traffic patterns in San Jose are the same during the entire week.

The bike activities for cluster 1 (Tuesday, Wednesday, and Thursday) and cluster 3 (Saturday and Sunday) are similar for some stations in the network. That can be seen in stations 58 and 59 (San Francisco Caltrain 2–330 Townsend and San Francisco Caltrain–Townside at 4th). When taking a closer look at the location of these two stations, we found that these two stations are located close to the Caltrain station. Consequently, the similarity between these two clusters can be linked to the train timetable.

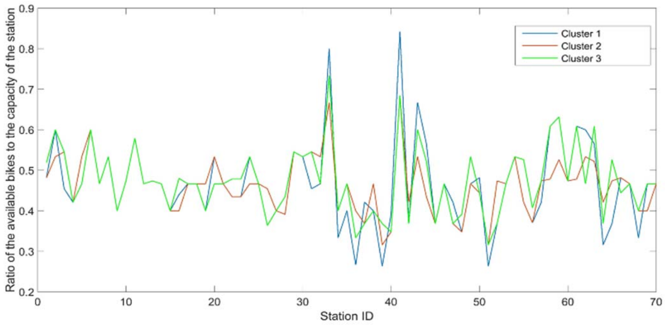

Fig. 5. The ratio of the availabile bikes to the capacity of the station for the three clusters at station in the network.

C. *Hour-of-the-Day*

Only the station data at the beginning of each hour was considered (e.g. 8:00am, 9:00 am, 10:00am …etc). The optimal number of clusters was found to be two (*K*=2). The analysis of the data reveals that the two clusters are peak (cluster 2) and non-peak (cluster 1) hours, thus confirming the previous research.

The results of the clustering are given in Fig. 6 and Fig. 7, which give the probability of an hour being in one of the two clusters and the pattern of each cluster. It can be concluded from Fig. 7 that when the patterns of the two clusters are lined up, the bike activity in the peak and non-peak hours is the same. For the sake of space, we will not go in detail with regards to the hour label.

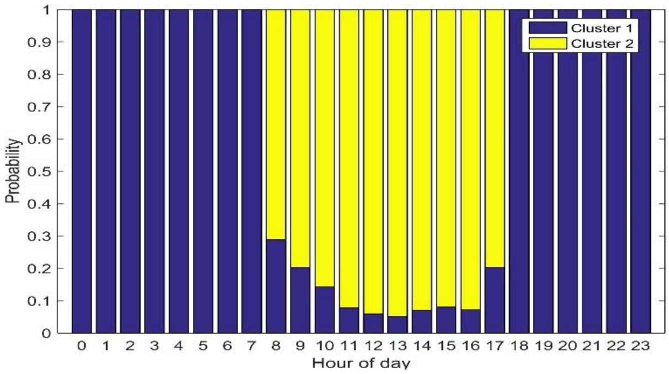

Fig. 6. Probability of hour to be in one of the two clusters (k=2).

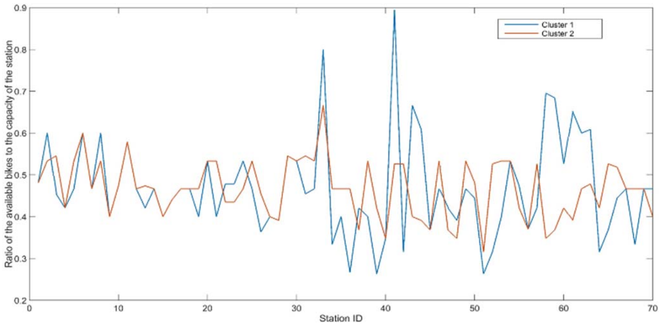

Fig. 7. Available bikes of the three clusters for each station in the network.

VI. Conclusions

This paper provides a useful tool for agencies and researchers to anticipate bike availability at stations with respect to a time event. A new supervised clustering algorithm was proposed and tested on a BSS in the San Francisco Bay Area. The proposed algorithm clusters data at 15-minute intervals of the bike availability across the network and finds the similarity between them according to day-of-the-week and hour-of-the-day. Subsequently, it provides an expected pattern of bike usage for each cluster. The algorithm provides insight into the usage patterns of the San Francisco Bay BSS that operators can use to anticipate imbalances in the system and plan accordingly.

The clustering results show that the days of the week can be grouped into three clusters, one for weekends and the other two for weekdays. The time-of-day is clustered into two groups, peak and off-peak hours. Given that each cluster has an associated pattern of bike availability, a prediction can be made to identify the imbalance in the system for each day of the week and each hour of the day. An exploratory spatiotemporal analysis was conducted, leading to different suggestions on how to rebalance the system with minimum cost and effort, thus making the network a more-effective component of a sustainable urban transportation system. One of our interesting findings is that some high-demand stations are located next to low-demand stations, which presents the possibility of linking them together with either a free transportation service or incentive discount for bikers to change their pick up or drop off destinations. Furthermore, this research can be easily extended to include the demographics of San Francisco and San Jose and make a comparison between them. More time labels can also be investigated.

References

[1] P. DeMaio, "Bike-sharing: History, impacts, models of provision, and future," *Journal of Public Transportation,* vol. 12, p. 3, 2009.

[2] J. Schuijbroek, R. Hampshire, and W.-J. van Hoeve, "Inventory rebalancing and vehicle routing in bike sharing systems," 2013.

[3] R. Alvarez-Valdes, J. M. Belenguer, E. Benavent, J. D. Bermudez, F. Muñoz, E. Vercher, *et al.*, "Optimizing the level of service quality of a bike-sharing system," *Omega,* vol. 62, pp. 163-175, 2016.

[4] C. Etienne and O. Latifa, "Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the Vélib'System of Paris," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 5, p. 39, 2014.

[5] C. F. Eick, N. Zeidat, and Z. Zhao, "Supervised clustering-algorithms and benefits," in *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, 2004, pp. 774-776.

[6] S. Basu, M. Bilenko, and R. J. Mooney, "Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering," in *Proceedings of the ICML-2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, 2003, pp. 42-49.

[7] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *ICML*, 2003, pp. 11-18.

[8] J. Sinkkonen, S. Kaski, and J. Nikkilä, "Discriminative clustering: Optimal contingency tables by learning metrics," in *European Conference on Machine Learning*, 2002, pp. 418-430.

[9] A. Demiriz, K. P. Bennett, and M. J. Embrechts, "Semi-supervised clustering using genetic algorithms," *Artificial neural networks in engineering (ANNIE-99),* pp. 809-814, 1999.

[10] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly,* vol. 69, pp. 9-15, 1962.

[11] J. Froehlich, J. Neumann, and N. Oliver, "Sensing and Predicting the Pulse of the City through Shared Bicycling," in *IJCAI*, 2009, pp. 1420-1426.

[12] A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. Banchs, "Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system," *Pervasive and Mobile Computing,* vol. 6, pp. 455-466, 2010.

[13] P. Vogel, T. Greiser, and D. C. Mattfeld, "Understanding bike-sharing systems using data mining: Exploring activity patterns," *Procedia-Social and Behavioral Sciences,* vol. 20, pp. 514-523, 2011.

[14] D. Marcu, "A Bayesian model for supervised clustering with the Dirichlet process prior," *Journal of Machine Learning Research,* vol. 6, pp. 1551-1577, 2005.

[15] G. Forestier, P. Gançarski, and C. Wemmert, "Collaborative clustering with background knowledge," *Data & Knowledge Engineering,* vol. 69, pp. 211-228, 2// 2010.

[16] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*: Prentice-Hall, Inc., 1988.

[17] Y. Şenbabaoğlu, G. Michailidis, and J. Z. Li, "Critical limitations of consensus clustering in class discovery," *Scientific reports,* vol. 4, p. 6207, 2014.