

Business Scaling and Rebalancing in Shared Bicycle Systems

Sourabh Ghorpade¹, Uzair Baig², Pratik Botalje³, Mahendra Nivangune⁴

^{1,2,3}Student, Department of Computer, Sinhgad Academy of Engineering, Pune, India

⁴Professor, Department of Computer, Sinhgad Academy of Engineering, Pune, India

Abstract - Bike sharing systems, aiming at providing the missing links in public transportation systems, are becoming popular in urban cities. A key to success for a bike sharing systems is the effectiveness of re balancing operations, that is, the efforts of restoring the number of bikes in each station to its target value by routing vehicles through pick-up and drop-off operations. There are two major issues for this bike re balancing problem: the determination of station inventory target level and the large scale multiple capacitated vehicle routing optimization with outlier stations. There is also one problem that occurs in sharing bicycle is finding the new place to implement services.

Key Words: Bike sharing system, Clustering, Optimization, Business Scaling.

1. INTRODUCTION

The dynamics of human mobility often lead to inevitable bike supply/demand imbalance. Thus, it is crucial for service providers to redistribute bikes among stations in a proactive and economical way in order to ensure system functioning effectively. To this end, in this paper, we study the bike rebalancing problem, where the assumption is that the number of bikes at each station is known in advance and will not be changed during the rebalancing operation (when system is closed or during midnight).the emergence of multi-source big data enables a new paradigm for enhancing bike sharing services number of recent researches have studied the bike rebalancing problem.[1] Junming Liu and group have created research paper on same problem ,it consist of prediction and finding optimise route. We have used their research and further optimised it by introducing run time prediction and finding optimised route according to it .we have also introduced the solution for infinite station balancing procedure by using their probability weights.

This paper also provides the efficient way for “business scaling” problem. But the problem is how to find areas if we try to do it manually and run some surveys and find area it will need heterogeneous work and it will reduce the efficiency. One solution is to use new technologies like machine learning and data mining.

2. Problem Formulation

2.1 Station network

$G=(S, E)$ directed graph represent the bike station network. Connection between bike stations represented by directed edges from edge set E . Station represented as $s \in S$.

Bike trip records are used to construct station network. A trip record $tr = (s_0, s_d, \tau_0, \tau_d)$ is a bike usage. Record from an origin station s_0 to a destination station s_d . τ_0 is the pick-up time and τ_d is the drop-off time. Only trips with duration $\tau_d - \tau_0$ larger than 1 min are recorded.

2.2 Station pickup/drop-off demand

Station bike demand- pick-up/ drop-off frequency per unit time when the station is available. Station Availability- station is in service and there are bikes available for pick-up/ drop-off Station Unavailability- due to maintenance, empty dock (for pick-up) or full dock Station pick-up/ drop-off demand.

$$S_i.pickupdemand = \frac{S_i.pick_{freq}(t)}{S_i.pick_{avail}(t)}$$

$$S_i.dropoffdemand = \frac{S_i.drop_{freq}(t)}{S_i.drop_{avail}(t)}$$

2.3 Station rebalancing target

This gives the demand (pickup drop off) of each station the station capacity $s_i.c$, the current number of bikes $s_i.cn$ and the station bike net flow $s_i.nf(t)$.number of bikes that need to be picked up or delivered is defined by $s_i.rt$. In order to maximize the time duration in which stations remains balanced in achieved by optimal inventory level.

2.4 Un-reachability Problem

This states the scenario when user search for bicycle and nearby bicycle is at large distance than threshold. Threshold represented by th . Distance from user to bicycle represented by $D_{u,b}$. Following equation defines the satisfaction of unreachability problem.

$$D_{u,b} > th$$

3. Problem Definition

We solve first problem by predicting the station level pick up and drop off and then re balancing them efficiently. This phases are executed at runtime of transport vehicle.

3.1 Station level bike demand prediction

In this, historical set of bike trip records are taken $TRH = \{tr1, tr2, \dots, trH\}$ and weather report R_t , the problem of station level bike pick-up (drop-off) prediction is to predict the hourly pick-up and drop-off demand of each station during a day.

3.2 Static bike rebalancing optimization

Given a set of bike station locations s_i of size m , rebalancing target $s_{i,rt}$, their inter vehicle transportation distance TD_{ij} and the number of operating vehicles $|V|$ with specified capacity limit V_C , the problem of bike rebalancing optimization is to decide the optimal vehicle rebalancing routes

$$F(x) = \sum_{i=1}^m \sum_{j=1}^m \sum_{v \in V} x_{ijv} \text{Tran}_{Dist_{ij}}$$

4. An overview of framework

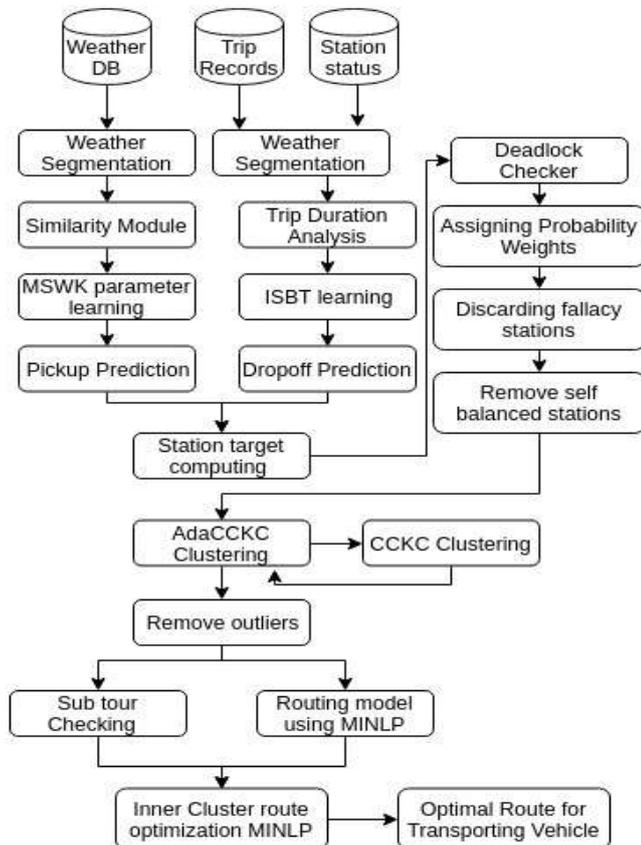


Figure 1: Rebalancing Framework

As shown in Figure 1 Rebalancing consists of 3 phases: first part is prediction part where demand (pick-up and drop-off) of each stations take place. In second phase stations causing the unbalanced chain were removed according to their weight. Third part is about finding optimal re balancing solution.

4.1 Station level bike demand prediction

Bike sharing systems are used to collect bike's historical data and station directory status data. This data is used to calculate benchmark data by using first 2 equations.

The Weather data is extracted according to time slot. Then preprocessing is applied on unstable weather state records and misplaced records. In this step, Each record in categorized according to suitability for outdoor bicycling using clustering: $W1 = \{\text{heavy snow, heavy rain}\}$, $W2 = \{\text{snow, rain, light snow, light rain}\}$, $W3 = \{\text{fog, mist, haze}\}$, $W4 = \{\text{Cloudy, Sunny}\}$.

Meteorology similarity weighted KNN (MSQK) regression is build using meteorology similarity function learning for station bike pick-up demand prediction.

Inter-station bike transition (ISBT) predictor is used to predict the station bike drop-off demand which approximate the drop-off station and drop-off time after a pick-up event based on connected network station.

4.2 Handling unbalanced chain

Underflow situation that total predicted drop-off demand can be greater than total no of bicycle available in all station. This problem causes the Bike station rebalancing algorithm to run in infinite loop. When algorithm creates cluster for building the station rebalancing network each cluster will try to add or remove the station, this will create the chain of cluster where each cluster after balancing, unbalanced the other clusters.

To avoid this chain we need to remove such stations prior to bike station rebalancing. Assigning the weight to each station and removing according to it is a optimized solution. Weights are given by the probability of prediction. In case of overloading station having low probability prediction should be removed.

4.3 Bike station re-balancing

Several self-balanced stations (stations with net-flow close to zero) are excluded, According to the station level target. Adaptive Capacity constrained K centers clustering (CCKC or AdaCCKC) is user to cluster remaining stations and outlier stations are also discovered.

In order to create optimize vehicle rebalancing routes for the rest stations within each cluster method uses MINLP model with lazy constrains.

4.4 Rebalancing optimization

When vehicle transporter login rebalancing algorithm predicts the demand of stations and calculate optimized route. We optimized it further by executing the rebalancing algorithm on each station dynamically and whenever the vehicle transporter is ready for next trip. Dynamic approach uses real time data.

4.5 Prediction of ideal location for service implementation

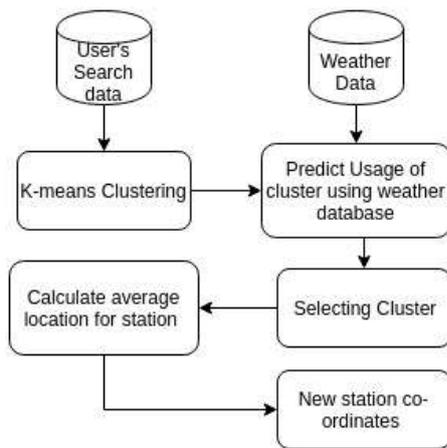


Figure 2: Business Scaling Framework

As shown in Figure 2 Business Scaling consists of single phase. This phase uses the user's location data for finding ideal location for service. When user faces unreachability problem his location get stored in db. Satisfaction of unreachability is given by (2). This phase extract location data and runs k-means clustering on that. At completion it finds the optimal location.

5. The rebalancing approach

In this section we will give the information about working of algorithm.

5.1 Station Bike Pickup Prediction

In order to predict the station level bike pick-up demand $station.pickupdemand(D^t)$ during time slot t of any given day D MSWK regressor is build which works on meteorology multisimilarity function

1. Similarity measurement:

Weather condition (sunny, raining etc.), humidity, temperature, wind speed and visibility of time slot t on day D_p , linear combination of three units: weather similarity, temperature similarity and humidity-wind speed-visibility similarity is used to calculate the similarity between 2 different days.

2. MSWK learning

According to similarity function, K days $\{D^{t_1}, D^{t_2}, \dots, D^{t_K}\}$ with the highest similarity to our target day D^{t_n} selected. Then using similarity weighed KNN station.pickupdemand (D^{t_n}) is predicted.

5.2 Station Bike drop of prediction

1. Station inter transportation analysis

Given predicted station bike pick-up demand during time slot t ($s_i.pd(t)$), the number of bikes that will be dropped off at station s_j from s_i is estimated from trip history records.

2. Station level drop-off prediction

$$s_i.dropof_prediction = \sum_{p \neq q} e_{pq^t} p_{pq^t} + e_{pq^{t-1}} p_{pq^{t-1}}$$

The first term represents the estimated drop-offs during the same time slot as their pick-ups and the second term represents the estimated drop-offs one time slot later than their origin pick-ups from other stations.

5.3 Station rebalancing optimization

Pseudocode 1

CCKC (TD, b, VN, VC, δ , E)

1. for all station s_i do

$$Cluster(s_i) = \min_{s_2 \in S} (\text{Distance}(s_i, s_2))$$

2. for all cluster c_i do

If c_i is not balanced then

While (c_i is not balanced) do

$$q = \min (\sum_{st \in c_i} \sum_{cx \in Cls} \text{Distance}(st, cx))$$

Remove (q)

Repeat step 2 if all clusters are not balanced

3. for all station s_i do

If $s_i.label == null$ then

$$Cluster = \min (\sum_{c_i \in C} \text{distance}(s_i, c_i))$$

$s_i.label = cluster.name$

4. for all Cluster C do

$$Center(C) = \text{average} (\sum_{s_i \in C} TD_{s_i})$$

5. Return clustering result C

Pseudocode 1 presents the proposed algorithm. It begins with an initial center set E, and assigns each station to its nearest stations. Then for a cluster, if the balance condition is not satisfied, we pick some stations out of the cluster. The stations, which are able to reduce the total balance of the cluster and close to other centers, are firstly picked out. In step 3, the unlabeled stations are assigned with new cluster label. For each unlabeled station, the new cluster label is determined by the total balance of a cluster and the distance between the station and its nearest station in the cluster. The unlabeled outlier stations that are far from cluster centers are preferentially processed. This step ensures these outliers scattered at the central region of the studied area, and can be easily covered by other clusters. After adjusting clustering result according to balance conditions, new centers are selected in Step 4. Step 1~4 are iterated until convergence (centers are unchanged). Step 5 outputs the clustering result.

Pseudocode 2 presents the proposed AdaCCKC algorithm. In each round, it begins with a randomly generated initial center set. Pseudocode 1 CCKC is implemented to get a temporary clustering result. If there exists unlabeled bicycle stations, a new cluster center is added to the current center set. The new added center is determined by all unlabeled stations. The break condition can also be activated if the number of outliers is below a specified threshold instead of 0, which makes the proposed algorithm more flexible for bicycle stations rebalancing problem. Considering the effect of initial center set on the final clustering result, the number of initial centers is set to vary from 1 to VN_{max} in Step 2, where VN_{max} is the maximum number of available vehicles. Step 3 picks out the best clustering result. Steps 1~3 are repeated many times to reduce the influence of initial center set. As a result, Algorithm 2 can automatically determine the optimal number of vehicles in a smarter way, and users do not need to provide an initial center set.

Pseudocode 2:

AdaCCKC (**TD**, b, VN_{max} , VC, δ , E)

1. Set $VN_{best}, Z_{best} = \sum_p \sum_q \text{Distance}(p,q)$

2. For I from 1 to VN_{max} do

Generate initial center set E

for j from I to VN_{max} do

c=call CCKC (**TD**, b, VN, VC, δ , E)

If number of outliers < threshold & 0 then
break

Else if unlabeled station \exists then

new_cluster=average($\sum_{s_i \in S} s_i \forall VC$)

3. If $VN < VN_{best}$ & $Z < Z_{best}$

$VN_{best} = VN$

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

6. The Business scaling approach

At specific time algorithm will take the location data from database use K-means clustering to divide the data into cluster. Each cluster will represent the different area. Find the cluster with large data set. Algorithm will use weather data to predict the usage at cluster in upcoming months and choose the cluster with high possible usage and large data set Find the average location using data from selected cluster. Calculated Average location will be the best place to implement the service or station.

data_collec (loc_data, weather_data, crr_time, inter_time)

Output: opti_loc

1. If $D_{US} > th$

$L = L + U_{location}$

2. If $crr_time = inter_time$

3:loc_groupid [[]] =clustering (L);

4: find l= (group_id with max member and max usage)

5:opti_loc=avg ($\sum \text{loc_groupif}[l]$)

7. CONCLUSIONS

In this paper, we developed a multi-source data smart optimization approach for addressing the station rebalancing and business scaling problem in bike sharing systems. Rebalancing Algorithm works in 3 phases, in first phase algorithm finds the pick-up and drop off demand of each station by using historical data, weather data and other resources. In second phase stations causing the unbalanced chain were removed according to their weight. In third phase Algorithm finds the optimized route for bike re-allocation vehicle by excluding self-balanced stations and clustering remaining stations.

Business scaling algorithm uses the k-means clustering algorithm to find preferred location for service implementation.

Both algorithm works in order to achieve system availability to its maximum.

ACKNOWLEDGEMENT

We take this opportunity to thank all the people involved in the making of this paper. We want to especially thank our respected guide, Prof. M.K.Nivangune for his guidance and encouragement. Our Head of Department Prof. B.B.Gite and Project Coordinator Mr. Santosh Shelke has also been very helpful and we are grateful for the support.

REFERENCES

- [1] Junming Liu, Leilei Sun, Weiwei Chen, Hui Xiong: Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization ACM, 2016.
- [2] R. Alvarez-Valdes, J. M. Belenguer, E. Benavent, J. D. Bermudez, F. Munoz, E. Vercher, and F. Verdejo: Optimizing the level of service quality of a bike-sharing system. Omega, 2015.
- [3] S. Basu, A probabilistic framework for semi-supervised clustering