AoCMM Solutions 2017

Team #726

10 October, 2017

Designing a Model to Compare Two Sets of Typing Test Results

Aditya Garg, Arjun Agarwal, Shreyas Minocha

10 October, 2017

Problem Statement

It has been shown typing patterns can be used to identify a person. To confirm this idea, we collected typing patterns from eleven of our officers using two different typing methods: fourteen short English quotes and six paragraphs composed of random characters. The quotes and paragraphs are grouped into three categories: eight quotes and three paragraphs where the officer's identity is known, six quotes where the officer's identity is unknown, and another three paragraphs where the officer's identity is also unknown. Based on the first category, how would one match the second and third categories to the officers?

Contents

1 Summary

1.1 Interpretation of Problem

As the problem states, each person has a unique style of typing. This means that the typing style of a person can be analysed and linked to the person. We aim to develop a model which would be able to compare the given typing sample to an unknown typing sample and determine how similar they are. The problem requires us to utilize our model to match the identities of the typists of the given typing samples to a set of unknown typing samples. In the following sections, we describe such a model.

1.2 Techniques and Methods

Our model takes into consideration four factors, the similarity of the typing speeds of the given and test samples, the uniqueness of the typing speeds of the samples, the errors made by the subject in each sample and the acceleration of different characters in both samples. For each of these four factors, we will compare the given and test samples and we will quantify these comparisons as similarity scores, each out of 100. Once the similarity scores have been calculated, we will merge these subscores into the final similarity score. Also, we calculate a threshold of the final similarity score beyond which we consider the two samples to be from the same subject.

1.3 Conclusion

We have developed a model to compare two typing test results by calculating an overall similarity score for the two samples. Our model can now be used to authenticate a user if typing data has previously been collected from the user. Our model does not make any assumptions about the skill level of the subject. Also, data collection for our model is convenient and can be carried out over the internet. Our model does not take into account that a user could deliberately provide incaccurate data. Similarly, the user could unintentionally provide data that does not represent his natural typing due to temporary variation. Such variation can be caused by nervousness, medication and others. Also, the typing style of subject can change over a long period of time and our model does not take this into account.

2 Introduction

It is regarded that the typing pattern of a subject is unique and its analysis can be a good way of verifying the identity of the subject. Such an analysis provides for a cheap method of authentication of a user that requires no equipment apart from a computer and a keyboard. This means that it can be conducted on a large scale and in a cost-effective way. The problem requires us to compare a given set of typing results of known typists with results from unknown typists. This can be achieved by describing a model which computes the similarity index of two typing test results. Using this, we will be able to match the typists with their respective typing test results. In this paper, we will describe a model that fits the above description.

3 Model

3.1 Introduction to the Model

For using the model, it is necessary to gather some information about the typing habits of the subject(s). We choose to do this by making all subjects give typing tests. These tests will be of two types: quotes and random letters. The subject will be expected to give multiple tests of both categories in order to ensure that the data collected is exhaustive and the sample collected from the subject is representative of his natural typing habits. We will then use our model to analyze the data collected from the typing test.

3.2 Assumptions and Justifications

We make some assumptions in the construction of our model. These are as follows:

Assumption: The text on which the subject will be tested consists of only alphanumeric characters and common symbols(those present on a standard QWERTY keyboard).

Justification: Presence of unfamiliar of unusual characters in the test may confuse the subject and looking for the character on their keyboard may reduce their speed.

Assumption: The keyboard used by the subjects to give the typing test is standardized and is the same for all trials.

Justification: The design and features of the keyboard should not cause variation in the results of the typing test. Instead, only the typing skills of the subject should influence the results of the test.

Assumption: The subject attempts the typing test just as they would normally type, that is, they do not intentionally or unintentionally exhibit abnormal typing behaviour.

Justification: Often nervousness and self-consciousness can lead to a reduced accuracy and other behavioural anomalies. Such temporal variation can make the data unreliable.

Assumption: There is no sharp change in the subject's typing behaviour between the collection of both the samples.

Justification: This is because if there is an observable change in the subject's typing behavior(due to the time period between tests, age, change in skill, etc), their typing pattern will be completely different during the test, thus it would not be possible to identify them.

3.3 Definitions

Characters per minute(CPM): The number of characters typed correctly by the subject in 60 seconds.

Word: We define a word as 5 characters. Gross WPM/Raw WPM(WPMG): The speed of the subject when he types all word correctly. This is determined by dividing the total number of words typed by the total time taken.

Error penalty: Sometimes, the subject may type a word incorrectly and not correct the mistake which will result in some amount being reduced from his WPMG. This is calculated by dividing the total uncorrected words by the total time taken to type the text.

Net WPM: The real speed of the subject which takes into account the WPMG as well as the wrongly typed words. This is calculated by subtracting the number of uncorrected words from the total correctly typed words and hence dividing the difference by total time taken.

Speed/Typing_speed: The typing speed of the subject expressed in WPM.

Accuracy: The percentage of correctly typed characters out of the total characters typed.

Sample: Each subject has given multiple typing tests. 'Sample' refers to the results of all these tests collectively.

Given: A type of sample, the identity of the typist of which is known.

Test: A 'Test' is the type of sample, the identity of the typist of which is unknown whose typing data we have.

3.4 Variables and Parameters

- Similarity of speed between the samples to be compared

If the given sample and the test sample have similar speeds, then we can guess that both samples were collected from the same subject. It just increases the similarity between all samples. Similarly, if there is a huge variation in the speeds of the known and unknown samples, it is unlikely that they were collected from the same subject.

- .Uniqueness of typing speed

If both the given and the test samples have an average speed which has a low frequency in the difficulty distribution, the similarity in speed gives us a large assurance of the samples belonging to the same subject(since there is a high probability of the speeds being similar). On the contrary, if the samples have similar speed and the speed has a low frequency, the samples should be considered more similar.

- Types of mistakes and weak characters

Types of mistakes reveal a lot about a person. If a subject is unfamiliar with certain keys or patterns, they might have a tendency to make errors in those patterns. Such tendencies are formed and magnified over the years with practice. If mistakes in certain characters are observed in the given sample and similar mistakes are observed in the test sample as well, there is a high chance that the samples were collected from the same subject.

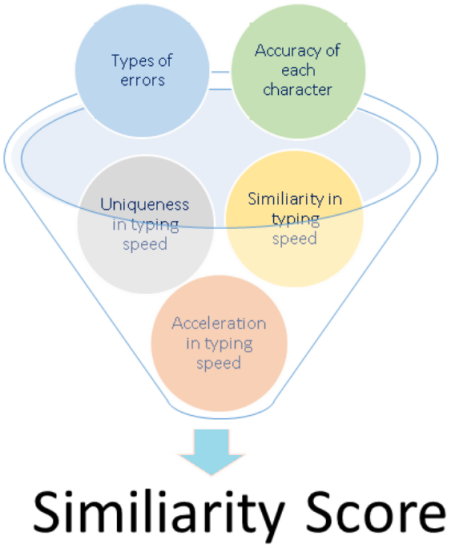
- Acceleration/Deceleration between two characters

Over a large time span, people develop muscle memories for certain keys. Certain key presses and transitions develop a faster speed and better accuracy while those that aren't practised as much have weaker muscle memories and tend to be slower and/or less accurate. Since the muscle memories one develops are unique to each person, the acceleration between certain characters can be used to reliably point out the similarity or dissimilarity between a set of samples.

3.5 Methods

Our model compares the given and the test typing speed results by calculating a score of similarity between the two samples. The similarity score is graded from 0 to 100 where a similarity score of 0 means that the samples are completely dissimilar and a score of 100 means that the samples are a sure match. We start by calculating a score on the same scale for each of our chosen factors. Once these subscores are calculated, we combine these scores into the final similarity score depending on the weightage of each factor in the final score.

Figure 1: Overview of model



3.5.1 Similarity in Typing Speeds

Similarity in typing speeds is the first thing that we decided to compare when we were planning our model. We decided to quantify the similarity in typing speed as follows:

$Similarity = \left(100 - \frac{|test - given|}{given}\right) \times 100$ where $test$ is the average speed of the test sample and $given$ is the average speed of the given sample.

3.5.2 Uniqueness of Typing Speed of the Samples

In the following difficulty distribution, the frequency of number of people that completed the test with an average speed in the 60—70 to range is very high. Hence, the probability of a random test subject's average WPM on the score falling in that range is also higher. This means that the probability of the event of the average speed of two samples landing in this range is high. On the contrary, the probability of the average speed of two samples lying in the 110—120 range is much lower. Thus, if the given and the test samples have similar average speeds and those in a low frequency range, we have a greater assurance of the similarity than if they were in a high frequency range. In order to quantitatively describe this relationship, we came up with the following formula:

$Uniqueness = 1 - P(test)$where $test$ has its usual meaning.

3.5.3 Analysis of the Errors Made by the Subject

Errors are an important factor to consider in this model. We calculate the accuracies of each letter typed by the subject(s) for the given sample. This is done as:

$$Accuracy = \frac{n(x)}{f(x)}$$

In the above equation, $n(x)$ is the number of errors made by the subject for any character x and $t(x)$is the total number of times x is typed in the sample. Once we have the accuracy of all characters present in the sample, we need to identify which of these characters are frequently mistyped by the subject, that is, mistyping which characters is a characteristic error of the subject. To do this, we need to find a threshold of accuracy beyond which we consider the character in question to be a commonly mistyped character. We decide to do this by calculating the standard deviation of the accuracies of all characters in the sample as follows:

Standard Deviation = 1Ni=iN(Ai- A)2where Aiis the accuracy for the ithcharacter and Ais the mean of the accuracies of all characters.

We decided to consider the accuracies beyond two standard deviations of the mean on the positive side to be characteristic errors of the subject.

$$DistinctiveErrorCharacters = \{x | x \in ErrorCharacters, Accuracy(x) > Mean + (2 \times StandardDeviation)\}$$

We compare the accuracy of each distinctive character thus identified in the given sample with the accuracy of that character in the test sample as

$SimilarityOfErrors(x) = (1 - \frac{|test-given|}{given}) \times 100$ where x is the character in question, $test$ and $given$ have their usual meanings.

We then calculate the mean of the similarity values calculated for each character determined to be distinctive.

error score= i=1nsimilarity of errors(xi)nwhere nis the number of distinctive errors.

The number thus obtained is later factored in the final similarity score between the given and test sample.

The types of errors a subject makes is also an important factor to be considered. We decided to consider 5 types of errors: “Case errors”, “Typed-early errors”, “Bad ordering”, “Doublet” and “Uncategorized errors”. We find out what percentage of the errors in a sample fall into each category as:

$PercentageOfErrorType(x) = \frac{n(x)}{totalnumberoferrors} \times 100$ where x is any error type and $n(x)$ is the number of errors of that type made in the sample

If for any error type the percentage of that error type is above a certain threshold, we consider that error type as a characteristic error type of that given sample and compare the percentage value of such error with the value obtained for the test sample. However, we choose to never consider “Uncategorized errors” as distinctive error types. Since we have 5 types of errors, if the percentage of any error type in the given sample is found to be more than 20%, we consider this to be a distinctive characteristic of the subject. We then compare the percentage of the each error type determined to be distinctive with that of the test sample. We quantify this as:

$SimilarityOfErrorType(x) = (1 - \frac{|test-given|}{given}) \times 100$ where $test$ and $given$ have their usual meanings.

Finally, we calculate the mean of all error type similarity indices to find the error type score. Eventually, we factor this into the final similarity score as:

$ErrorTypeScore = \frac{\sum SimilarityOfErrorType(x_i)}{n}$ where n is the number of distinctive errors

3.5.4 Calculating the Final Similarity Score

We decided to give each factor we chose to consider in our model an equal weightage in the final similarity score. First, we combine the similarness and the uniqueness as

$$SpeedScore = Similarity \times Uniqueness$$

We then combine the $SpeedScore$, the $ErrorScore$ and the $ErrorTypeScore$ to obtain the similarity score as follows:

$$SimilarityScore = \frac{SpeedScore+ErrorScore+ErrorTypeScore}{3}$$

3.5.5 Threshold of Final Similarity Score

We choose to keep 75% as a threshold for the final similarity score beyond which to consider the samples to have come from the same subject based on analysis of the given data.

3.6 Explanation of Solution

Our model helped us in matching the test samples to the respective typists by calculating a similarity score between them. The similarity score was calculated by considering the various characteristic features of the typing sample as specified in the model, with each of these features weighing equally in the final similarity score. Through our model it was found that, the typing samples of Person 4 and that of Person E were highly similar as the similarity score between them was quite high. This score was calculated by quantifying the similarity in the three parameters are model uses.

Figure 2: Character accuracies for person 4

3.6.1 Typing speed of the samples

The two typing speeds (Given: 105 Wpm, Test: 110 Wpm) were first compared for similarness with the help of our formula.

This came out to be 95%. Then the uniqueness of the speed was calculated by the formula below. For the given speed (105 wpm), this was found to be 0.95. The final similarity of the two speeds was thus calculated by multiplying the uniqueness with the similarity. For the given samples, this similarity came out to be 92%. This is the speed score of the data.

3.6.2 Error frequencies of specific characters

Through analysis, a frequency table of all the wrongly typed characters of the given people was determined. The graph of character acccuracies for Person 4 and Person E is as follows.

The standard deviation of both the graphs was determined and a threshold value was calculated by the following formula. For the given tests it came to be 3.1. Thus only the wrongly typed characters which had a frequency greater than

Figure 3: Character accuracies for person E

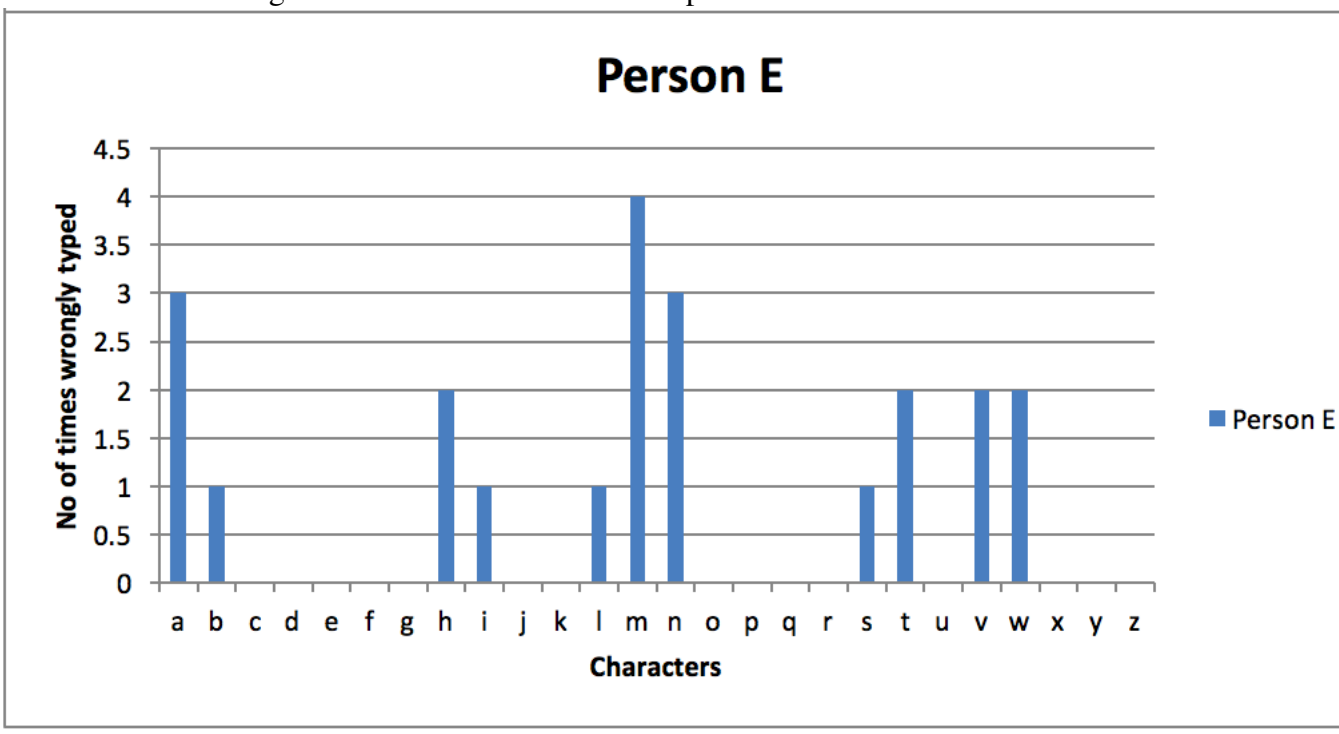


Table 1: Error types and their frequencies; Accuracies of error types

Error type	Person 4	Person E		
Bad Case	2	5		
Bad Ordering	5	5		
Doublet	1	2		
Other	12	6		

Table 2: Accuracies of error types

Error type	Person E	Person 4
Bad Case	0.27	0.10
Bad Ordering	0.27	0.25
Doublet	0.05	0.11
Other	0.60	0.30

3.1 were considered. In our case, the alphabet above the threshold was only letter 'M'. The accuracy of 'M' in both test and given was calculated by the formula as described in Section 3.5.3. These came out be approximately 11 and 13 respectively. Thus the similarness in these accuracies were calculated by the formula for similarity of errors as described in Section 3.5.3. This came out to be approximately 83.

3.6.3 Error types

For our given people, the various error types and their frequency as mentioned in section 3.5.4 was found out. The table including all the error types is given below:

The accuracy of each of the error types for each of the two person was calculated using the formula described in section 3.5.4. The total errors done by person 4 are 20 and by person E is 18. Hence the similarity score of each error type was calculated using the formula:

The similarity of each of the error types are given below.

Hence the total similarity scores for the section 'error type' was calculated by taking the average of all of these similarity scores. This came out to be 56.27.

Table 3: Similarness of error types

Error type	Similarity
Bad Case	37.03%
Bad Ordering	92.59%
Doublet	45.45%
Uncategorized	50%

Table 4: Correspondence of given samples to test samples

Test	Given
A	5
B	11
C	10
D	2
E	4
F	8
G	6
H	3
I	7
J	9
K	1

3.6.4 Computing the final similarity score

The final similarity score is calculated by taking the average of the speed score, similarity in accuracy and the total similarity score from the error types. The final similarity score between the two data is 77%. Since the threshold for the similarity between two data is 75%, we can conclude that both of these data are from the same person. So Person E corresponds with Person 4.

3.6.5 Solution to the Problem

3.7 Analysis and Assessment

3.7.1 Sensitivity Analysis

A sensitivity analysis of our model can be carried by observing how the change in a person's characteristic typing habits can lead to a change in its similarity score.

Taken typing results of two samples, a similarity score can be calculated. This score can change with the change in any of the factors (Similarness in Wpm, Types of errors, Frequency of error and acceleration in characters) in any one of the typing samples. However, our model is such that the change in different factors will lead to a change in the similarity score differently. For example, a small change in the frequency of errors of specific characters will not lead to a change in the similarity score, until the frequency goes above the threshold value. On the other hand, even a small change in WPM of a any of the samples being compared will lead to a change in the similarity score between them.

3.7.2 Strengths

- Our model does not make any assumptions about the skill level of the subject. Hence, our model can be used to compare samples of anyone, regardless of how skilled they are in typing.
- Data collection is convenient as no additional equipment is needed. Hence, our model can be used on a large scale.

3.7.3 Weaknesses

- Analysis of typing is a behavioural, rather than a physical characteristic of the subject. This means that it is subject to change over time with practice or the lack thereof. The model does not accommodate such changes in typing behaviour very well.
- Temporal influence, or the temporary change in typing behaviour due to nervousness, medication etc may lead to unreliable data.
- A subject could intentionally provide the model inaccurate data. The model does not take this into consideration.

4 Citations

References

[1] “Typing Equations - SpeedTypingOnline.” Speed Typing Online, www.speedtypingonline.com/typing-equations.

[2] “Keystroke Dynamics.” Wikipedia, Wikimedia Foundation, 6 Oct. 2017, www.en.wikipedia.org/wiki/Keystroke_dynamics.

[3] “How Do YOU Type ‘Wolfram’? Analyzing Your Typing Style Using Mathematica.” Wolfram Blog, www.blog.wolfram.com/2012/06/14/how-do-you-type-wolfram-analyzing-your-typing-style-using-mathematica.

[4] Dawn Song, et al. “User Recognition by Keystroke Latency Pattern Analysis.” User Recognition by Keystroke Latency Pattern Analysis, 8 Apr. 1997, pp. 1–4., www.users.ece.cmu.edu/~adrian/projects/keystroke/mid.pdf.

[5] “Identifying Emotion by Keystroke Dynamics and Text Pattern Analysis.” Taylor & Francis, www.tandfonline.com/doi/abs/10.1080/0144929X.2014.907343?journalCode=tbit20. Prima Chairunnanda.

[6] “Privacy: Gone with the Typing! Identifying Web Users by Their Typing Patterns.” Privacy: Gone with the Typing! Identifying Web Users by Their Typing Patterns, pp. 1–15., www.petsymposium.org/2011/papers/hotpets11-final8Chairunnanda.pdf.

[7] Sandhya Avasthi, and Tanushree Sanwal. “Biometric Authentication Techniques: A Study on Keystroke Dynamics.” Biometric Authentication Techniques: A Study on Keystroke Dynamics, Jan. 2016, pp. 1–7., www.ijseas.com/volume2/v2i1/ijseas20160125.pdf.

[8] Stross, Randall. “Bypassing the Password.” The New York Times, The New York Times, 17 Mar. 2012, www.nytimes.com/2012/03/18/business/seeking-ways-to-make-computer-passwords-unnecessary.html.

Determining What a Taxi Driver Should Do When His Taxi is Vacant and Determining What the Head of a Taxi Company Should Advise The Drivers Do in That Situation

Aditya Garg, Arjun Agarwal, Shreyas Minocha

10 October, 2017

Problem Statement

There will always be times when taxis are vacant. Some drivers say that you should head to the city center to find more customers, but is that always true? Suppose you are a taxi driver in NYC, what should you do when your car is vacant? If you are the head of a taxi company, what would you advise your drivers do?

Contents

1 Summary

1.1 Interpretation of Problem

Often, taxi drivers find there vehicles vacant. The first part of the problem requires us to determine what the driver's priority should be when looking for customers as well as develop a model to allow this priority to be fulfilled. the second part of the problem requires us to determine what the head of a taxi company should recommend his drivers to do, keeping in mind the overall profits of the company as well as ensuring that customers from all regions in the area being studied are served. This paper aims to develop a model to solve the problems as explained.

1.2 Techniques and Methods

We divided the region to be studied into several hypothetical square shaped regions. We then calculated the percentage of pickups made in each square out of the total number of pickups. We then modelled the area to be studied as a vertex-weighted undirected graph where each vertex corresponds to a region and an edge connecting two vertices corresponds to road connecting two regions. We calculate the weight of each vertex as the reciprocal of the pickup percentage we calculated. Since the driver would wish to maximize his odds of finding a customer, he should go to the region represented by the vertex with the smallest weight.

1.3 Conclusion

We have developed a model to determine where a taxi driver should go and what route should be followed when his taxi is vacant. Our paper also explains how the head of a taxi company should distribute taxis in the area being studied. Our model minimizes the time a taxi is vacant and waiting for customers. It also ensures that customers from all parts of the area being studied are served. However, our model can be used only when previous pickup data for the area being studied is available.

2 Introduction

The problem requires us to design a model to solve two problems. The first requires us to find where a taxi driver should go to look for more passengers when his vehicle is vacant. The second requires us to find out how the head of a taxi company would want taxis to be distributed throughout the city. We interpret that the head of the company would want his taxis distributed such that there are maximum taxis in high pickup density regions and less taxis in low pickup density regions.

3 Model

3.1 Assumptions and Justifications

Assumption: The taxi driver does not need to make any diversion for petrol, due to blockage etc.

Justification: In the ideal case, the driver should be engaged in doing his job throughout and should not have to divert from the optimal route for any reason.

3.2 Definitions

Pickup Percentage/Pickup Density: This is the number of pickups from a particular area.

Taxi Density: This is the number of taxis in particular area at a given point of time.

3.3 Variables and Parameters

- Percentage of pickups occurring in a certain region

The taxi driver would want to go to regions of high pickup percentage to increase chances of finding a customer. The head of the taxi company would want to distribute his taxis such that the number of taxis in a region is proportional to the pickup percentage, that is, taxi density in a region is calculated and matched based on the pickup percentage.

3.4 Usage Of Methods

Every taxi driver knows that any moment his vehicle is vacant, that time qualifies as a wasted opportunity to make money. Hence, the taxi driver's highest priority should be keeping his taxi occupied. In order to maximize the likelihood of him finding customers, the driver should drive to the area which has the maximum pickup density based on previous data.

Moreover, he should take the route that maximizes his likelihood of finding a customer. In other words, he should take the path that takes him through the areas of maximal pickup density.

We decided to divide the area to be studied into hypothetical square shaped regions of approximately 100 square metres. We then analysed the previously collected taxi pickup data to find which regions had the maximum number of pickups in the past. For each square region, we calculate the percentage of pickups that occur in that square as

$pickupdensity(x) = \frac{count}{total} \times 100$ for any square region x where $count$ is the number of pickups that were made from that region and $total$ is the total number of pickups in the available data.

We then model the area being studied as a vertex-weighted, undirected graph where each vertex represents one square region and each edge represents road connecting those two square regions. The weight of each vertex will be calculated as the reciprocal of the pickup density in that region.

$$vertexweight(x) = \frac{1}{pickupdensity(x)}$$

This decision will be explained in the subsequent paragraphs.

Since the driver wants to maximize their chance of finding a customer, he would want to drive to the region of highest pickup density and that too while taking the route that takes him through the most pickup-dense regions. To explain this in terms of our vertex-weighted graph model, starting at any vertex Vi(the taxi driver's initial region), we would like to find a route to the vertex with the highest weight(the most pickup-dense region). This route should also minimize the sum of the weights of the vertices we cross(since minimizing the sum of the reciprocal). An algorithm in graph theory called the "Shortest Path Problem". This problem involves finding a path between two vertices of an edge-weighted undirected graph such that the sum of the weights in the path is minimized. This problem pretty much corresponds to our scenario except that we have vertex weights instead of edge weights. In fact, this correspondence with the Shortest Path Problem

is precisely the reason we chose to keep the vertex weights as the reciprocal of the pickup density rather than the pickup density itself. Since the Shortest Path Problem already has several solutions such as Djikstra's Algorithm, the Bellman-Ford Algorithm, the Viterbi Algorithm and many others, we could find the ideal route for the driver to take by using a modified version of one of these algorithms.

A head of a taxi company will also wish to make sure that the taxis are occupied at all times. The head needs to distribute his taxis throughout the city on the basis of the pickup percentage. This will have the advantages:

1. More customers will be served
2. The time a taxi spends waiitng for a customer will be optimal
3. A balance between trips from high paying, pickup-sparse regions and those from low paying, pickup-dense regions(since pickup-dense regions are also dropoff-dense in general).

The head would calculate the ideal taxi density in any region as follows:

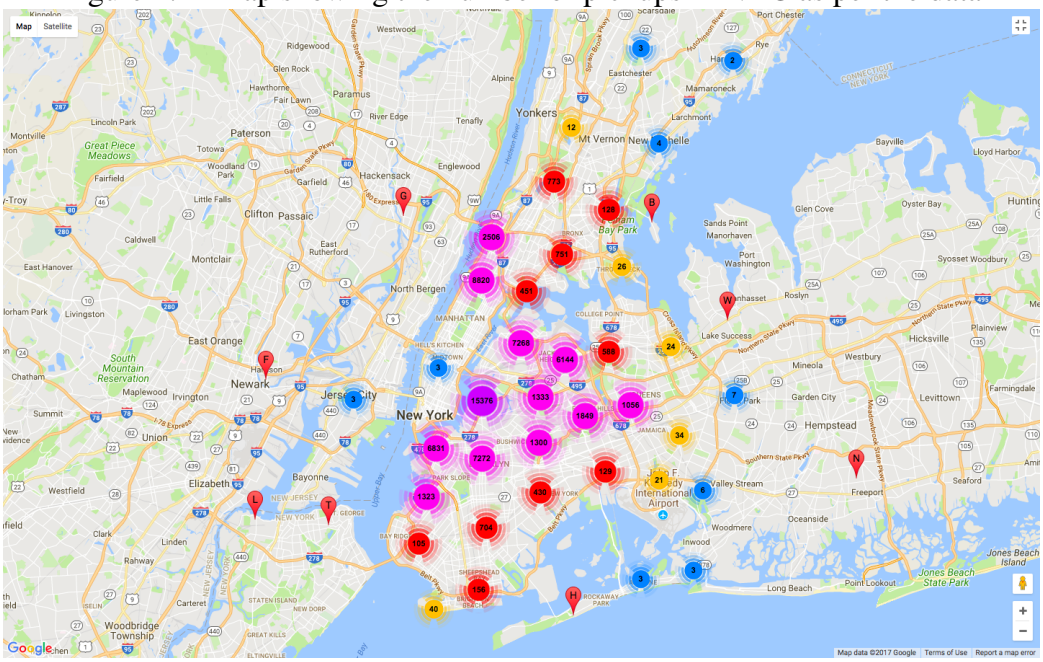
$$idealtaxidensity(x) = x \times \frac{pickupdensity(x)}{100} = x \times \frac{100}{pickupdensity(x)}$$ where x is the total number of taxis in the region to be studied.

The head will need to ensure that the taxis are distributed as per the ideal taxi density throughout. Also, the ideal taxi density should be updated in real time as the pickup density is updated. In general, taxis in regions with a real time taxi density greater than the ideal taxi density should drive to regions where the real time taxi density is less than the ideal density.

3.5 Explanation of Solution

We used the latitude and longitude data and logically divided the given area into square shaped regions. We truncate the latitude and longitude values up to the third decimal place. Then, we find the topleft most point in the data. Our hypothetical square shaped regions have a width and height of 0.001 longitude and 0.001 latitude respectively. This is approximately a 100 metre by 100 metre region in the real world. We then find the other three points of the square by adding 0.001 to the latitudinal value for one point, the same amount to the longitudinal value for the second point and finally we add 0.001 to both the latitude and the longitude value of the original point for the final point. Similarly, we use a computer program to divide the entire area to be studied into such square shaped regions.

Figure 1: A map showing the number of pickups in NYC as per the data



Next, our program calculates the pickup percentage for each square using the given data and Equation 1. We can now easily find the region with maximum pickup density. In our analysis, we found this region to be XXXXXXXXXXXX. We then map the square shaped regions onto our vertex-weighted undirected graph. As explained in Section 3.4, we calculate the weight of each vertex as the reciprocal of the pickup density of that region.the route a driver at any location should take to the region of maximum pickup density can now be calculated using a modified version of the Viterbi Algorithm.

3.6 Analysis and Assessment

3.6.1 Strengths

- Our model maximizes the number of customers served in the area to be studied.
- The amount of time a taxi spends waiting for customers is minimized.

3.6.2 Weaknesses

- Our model requires old pickup data for the region being studied. Hence, our model cannot be used for regions that do not have previous data.

4 Citations

References

[1] Goodrich, Michael T. “Weighted Graphs.” Algorithm Design , edited by Roberto Tamassia, John Wiley & Sons, Inc.

[2] McQuain. “Weighted Graphs.” Data Structures & Algorithms, 2000, courses.cs.vt.edu/~cs3114/Fall10/Notes/T22.WeightedGraphs.pdf.

[3] McGill. “Weighted Graphs.” Emory College of Mathematics and Compute Science , Emory College , www.mathcs.emory.edu/~cheung/Courses/171/Syllabus/11-Graph/weighted.html.

[4] Sedgewick, Robert, and and Kevin Wayne. “Minimum Spanning Trees.” Algorithms , Princeton University , algs4.cs.princeton.edu/43mst/.

[5] Viterbi AJ (April 1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". IEEE Transactions on Information Theory. 13 (2): 260–269. doi:10.1109/TIT.1967.1054010. (note: the Viterbi decoding algorithm is described in section IV.) Subscription required.

[6] Feldman J, Abou-Faycal I, Frigo M (2002). "A Fast Maximum-Likelihood Decoder for Convolutional Codes". Vehicular Technology Conference. 1: 371–375. doi:10.1109/VETECF.2002.1040367.

[7] “CS 312 Lecture 26 Finding Shortest Paths.” Department of Computer Science, Cornell University, University of Cornell , www.cs.cornell.edu/courses/cs312/2007sp/lectures/lec26.html.

[8] Frigioni, D.; Marchetti-Spaccamela, A.; Nanni, U. (1998). "Fully dynamic output bounded single source shortest path problem". Proc. 7th Annu. ACM-SIAM Symp. Discrete Algorithms. Atlanta, GA. pp. 212–221.

[9] Dreyfus, S. E. (October 1967). An Appraisal of Some Shortest Path Algorithms (PDF) (Report). Project Rand. United States Air Force. RM-5433-PR. DTIC AD-661265.