

ORDER DELIVERY REALLOCATION AND TRACKING

TECHNICAL FIELD

The present disclosure relates to electronic commerce and in particular to the processing of delivery of online orders.

BACKGROUND

The delivery of an online order, such as an order for a food item from a merchant such as a restaurant generally involves a delivery driver travelling to the merchant to collect the order and then traveling to a delivery location to drop off the order. Orders for food items are time critical since the customer generally wishes to receive their food order as soon as possible and food items, particularly heated items, have a short time period in which they must be consumed. Therefore, providing customers with visibility of the progress of delivery of a food item is an important feature of electronic commerce systems which process online orders. Many online ordering systems track the progress of delivery of orders using global positioning system (GPS) signals to track the status of delivery drivers. However, it can be difficult to track drivers as GPS signals are not always reliable. For example, GPS can cause battery drainage so delivery drivers may switch the GPS functionality off. In addition to this, heuristic rule tracking based on GPS signals can lead to false indications, for example when a driver takes a detour to avoid traffic or does a U-turn.

SUMMARY

According to a first aspect of the present invention, an online order processing method for processing an order is provided. The order comprises an indication of a merchant location at which an order item is to be picked up, an indication of a delivery location at which the order item is to be delivered, and a current estimated time of delivery. The method comprises: receiving current delivery driver information relating to a delivery driver assigned to deliver the order; looking up delivery driver data relating to the delivery driver assigned to deliver the order and order variables relating to the order and the merchant; inputting the current delivery driver information, the delivery driver

data and the order variables into a statistical model trained on historical driver and order variables, the statistical model being configured to predict a probability that the delivery driver will deliver the order within an estimated time of delivery; and determining an indication of the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery.

Embodiments of the present invention provide a predictive model that can fingerprint and predict, at multiple points in time during an order, whether a delivery driver is potentially moving towards a merchant or not. The model may incorporate location, speed and history of the delivery driver to identify the status of the delivery driver. This can incorporate all types of delivery driver transport, including four wheeled transport such as cars as well as two wheeled transport such as motorbikes or mopeds.

In some embodiments, if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is below a first threshold, the method further comprises determining a new estimated time of delivery and sending an indication of the new estimated time of delivery to a customer device associated with the order. Thus, the probability that the driver is not able to complete the order is incorporated into the estimated delivery time.

In some embodiments, if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is below a second threshold, the method further comprises reallocating the order to a second delivery driver. Thus, the probability information may be used to make a decision to pass the order to another delivery driver.

In some embodiments, if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is above the first threshold and below a third threshold, the method further comprises sending a notification to the delivery driver assigned to deliver the order. Thus, if the probability is not so high as to reallocate the order or adjust the predicted delivery time, but still not low, a soft intervention to the driver is made, for example by sending a push notification. This may not result in a change to the estimated time of delivery.

Some of the embodiments described above, particularly those which involve the order being reallocated to a second delivery driver, are implemented taking into account whether the order has been collected by the delivery driver. Thus, if the order has already been collected, it may not be reallocated to a second delivery driver. However, other aspects may be implemented even after an order has been collected, for example updating the estimated time of delivery or sending a notification to the delivery driver.

In some embodiments, the method further comprises receiving a complaint indication indicating that the delivery was not delivered within an estimated time of delivery and updating the statistical model.

In some embodiments, the method further comprises receiving a complaint indication indicating that the order was incorrectly reallocated and updating the statistical model.

When complaints are received from a customer, this indicates that an issue was caused potentially by the lateness of a delivery and the statistical model is updated accordingly. Similarly, if a complaint is received from a delivery driver, this may indicate that a delivery task was incorrectly reallocated away from the delivery driver and the statistical model is updated to avoid reallocation in such scenarios in the future.

In an embodiment, the statistical model is a multivariate classification model.

According to a second aspect of the present disclosure, a computer readable carrier medium is provided. The computer readable carrier medium carries processor executable instructions which when executed on a processor cause the processor to carry out a method as set out above

According to a third aspect of the present disclosure, an online order processing system for processing an order is provided. The order comprises an indication of a merchant location at which an order item is to be picked up and an indication of a delivery location at which the order item is to be delivered, the order having a current estimated time of delivery. The online order processing system comprises: a

processor, data storage, and program storage, the data storage storing a statistical model trained on historical driver and order variables, the statistical model being configured to predict a probability that the delivery driver will deliver the order within an estimated time of delivery, the program storage storing computer program instructions operative by the processor to: receive current delivery driver information relating to a delivery driver assigned to deliver the order; look up delivery driver data relating to the delivery driver assigned to deliver the order and order variables relating to the order and the merchant; input the current delivery driver information, the delivery driver data and the order variables into the statistical model; and determine an indication of the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery.

In an embodiment, the program storage stores computer program instructions operative by the processor to: if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is below a first threshold, determine a new estimated time of delivery and sending an indication of the new estimated time of delivery to a customer device associated with the order.

In an embodiment, the program storage stores computer program instructions operative by the processor to: if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is below a second threshold, reallocating the order to a second delivery driver.

In an embodiment, the program storage stores computer program instructions operative by the processor to: if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is above a first threshold and below a third threshold, send a notification to the delivery driver assigned to deliver the order.

Thus, if the probability of driver inactivity is very high, the delivery gets reallocated to another driver with change of estimated time. If the probability is lower, the customer receives an updated delivery time and if the probability of inactivity is not very high but not low either it gets as a soft reminder to the driver through a push notification

without change of estimated time. If the probability is low, then the delivery will continue as it is.

In an embodiment, the program storage stores computer program instructions operative by the processor to: receive a complaint indication indicating that the delivery was not delivered within an estimated time of delivery and update the statistical model.

In an embodiment, the program storage stores computer program instructions operative by the processor to: receive a complaint indication indicating that the order was incorrectly reallocated and update the statistical model.

In an embodiment, the statistical model is a multivariate classification model.

BRIEF DESCRIPTION OF THE DRAWINGS

In the following, embodiments of the present invention will be described as non-limiting examples with reference to the accompanying drawings in which:

FIG.1 is a block diagram showing a network for processing online orders according to an embodiment of the present invention;

FIG.2 is a block diagram showing an online order processing system according to an embodiment of the present invention;

FIG. 3 is a flowchart showing a method of processing an online order according to an embodiment of the present invention;

FIG.4 is a flowchart showing a method of updating a prediction model according to an embodiment of the present invention;

FIG.5A to FIG.5C are tables showing features used in prediction models in methods of processing online orders according to embodiments of the present invention; and

FIG.6A and FIG.6B are tables showing example scenarios of methods of processing online orders according to embodiments of the present invention.

DETAILED DESCRIPTION

The present disclosure relates to processing of online orders, for example for food orders from merchants.

FIG.1 is a block diagram showing a network for processing online orders according to an embodiment of the present invention. As shown in FIG.1, the network 10 comprises an online order processing system 100. The online order processing system 100 allows a customer 150 to make an order with a merchant 160 which may then be delivered by a first delivery driver 170 or a second delivery driver 180. The customer 150 uses a customer device 152, which may for example be a smart phone device, to place an order using an app which is executed by the customer device 152. The order may be a food order for one or more dishes offered by the merchant 160. Once the order is placed by the customer 150 using the customer device 152, an indication of the order is sent to the online order processing system 100. The indication of the order comprises an indication where the customer wishes the order to be delivered, which is referred to as the delivery location, an indication of the merchant and an indication of the items ordered. Upon receiving the order indication, the online order processing system 100 sends a message to a merchant device 162 associated with the merchant 160 indicating the ordered items so that the merchant can begin preparation of the order. The online order processing system also sends a delivery request to a first delivery driver device 172 associated with the first delivery driver 170. The delivery request comprises an indication of a merchant location which corresponds to the location of the merchant 160 and an indication of the delivery location. The first delivery driver 170 then accepts or declines the delivery request using an app running on the first delivery driver device 172. In some scenarios, the first delivery driver 170 may configure the app running on the first delivery driver device 172 to automatically accept the order request. It will be appreciated by those of skill in the art that the delivery request may be sent to multiple delivery drivers, for example to a second delivery driver device 182 associated with a second delivery driver 180. In such cases, the

delivery driver that responds to the delivery request first may be allocated the delivery task.

Once the delivery task is allocated to the first delivery driver 170, the first delivery driver 170 travels to the merchant location and collects the order, the first delivery driver 170 then travels to the delivery location to deliver the order. During this period, the customer 150 may be provided with information on the status of the order, for example, an indication of an estimated time of arrival may be provided by the app running on the customer device 152.

The present disclosure is concerned with taking action when the first delivery driver is not moving in the desired direction of the merchant and to reallocate another driver and providing an improved estimated time of arrival using information received from the first delivery driver device 172, such as the speed of the driver and additional information such as historical information relating to the delivery driver, information on the merchant and information on the current weather conditions. This information is used to predict whether the first delivery driver 170 will arrive within a predicted estimated time of arrival. In some embodiments, if the first delivery driver 170 is predicted not to arrive within the predicted estimated time of arrival, the estimated time of arrival is updated and an indication of the updated estimated time of arrival is sent to the customer device 152 by the online order processing system 100. In some embodiment, if the first delivery driver 170 is predicted not to arrive within the predicted estimated time of arrival, the delivery task is reallocated to another delivery driver, for example the second delivery driver 180.

FIG.2 is a block diagram showing an online order processing system according to an embodiment of the present invention.

The online order processing system 100 comprises a processor 110, a working memory 112, a network interface 114, program storage 120 and data storage 130. The processor 110 may be implemented as one or more central processing unit (CPU) chips. The program storage 120 is a non-volatile storage device such as a hard disk drive which stores computer program modules. The computer program modules are loaded into the working memory 112 for execution by the processor 110. The network

interface 114 is an interface that allows the online order processing system 100 to communicate with devices such as the customer device 152, the merchant device 162, the first delivery driver device 172 and the second delivery driver device 182 shown in FIG.1.

The program storage 120 stores a look up module 122, a prediction module 124, an estimated time of arrival update module 125, an order reallocation module 126 and a model update module 128. The computer program modules cause the processor 110 to execute various online order processing methods which are described in more detail below. The program storage 120 may be referred to in some contexts as computer readable storage media and/or non-transitory computer readable media. As depicted in FIG.1, the computer program modules are distinct modules which perform respective functions implemented by the online order processing system 100. It will be appreciated that the boundaries between these modules are exemplary only, and that alternative embodiments may merge modules or impose an alternative decomposition of functionality of modules. For example, the modules discussed herein may be decomposed into sub-modules to be executed as multiple computer processes, and, optionally, on multiple computers. Moreover, alternative embodiments may combine multiple instances of a particular module or sub-module. It will also be appreciated that, while a software implementation of the computer program modules is described herein, these may alternatively be implemented as one or more hardware modules (such as field-programmable gate array(s) or application-specific integrated circuit(s)) comprising circuitry which implements equivalent functionality to that implemented in software.

Although the online order processing system 100 is described with reference to a computer, it should be appreciated that the online order processing system 100 may be formed by two or more computers in communication with each other that collaborate to perform a task. For example, but not by way of limitation, an application may be partitioned in such a way as to permit concurrent and/or parallel processing of the instructions of the application. Alternatively, the data processed by the application may be partitioned in such a way as to permit concurrent and/or parallel processing of different portions of a data set by the two or more computers. In an embodiment, virtualization software may be employed by the online order processing system 100 to

provide the functionality of a number of servers that is not directly bound to the number of computers in the online order processing system 100. In an embodiment, the functionality disclosed above may be provided by executing the application and/or applications in a cloud computing environment. Cloud computing may comprise providing computing services via a network connection using dynamically scalable computing resources. A cloud computing environment may be established by an enterprise and/or may be hired on an as-needed basis from a third party provider.

The data storage 130 stores order data 132, delivery driver data 134, merchant data 136 and a prediction model 138. The order data 132 comprises details of the order made by the customer 150. This may include an indication of the customer 150, an indication of the merchant 160, indications of items making up the order, an indication of the time and date the order was made, indication of prices of the items making up the order, indications of any delivery fees or surge pricing relevant to the order and indications of any special offers or promotions which apply to the order. The delivery driver data 134 comprises indications of driver attributes for each delivery driver registered with the online order processing system 100. Examples of driver attributes are shown in FIG.5A to FIG.5C. The merchant data 136 comprises indications of various merchant attributes. Examples of merchant attributes are shown in FIG.5A to FIG.5C. The prediction model 138 is a classification model such as multivariate classification model which correlates the delivery driver data 134 and the merchant data 136 with a dependent variable. The dependent variable is whether a driver completed an order within a predicted time or not. This may be arrived at using the 90% percentile of all past orders. As will be described in more detail below, the prediction model is used to predict whether a delivery driver executing an order is actively executing the order and whether they are likely to deliver the order within a predicted estimated time of arrival.

Once historical data is collected with the dependent and independent variables, we process the features and the data further. Sometimes not all features are present and we “impute” the data to treat the missing values. Imputation is done to fill in missing values in the input data, which arises due to no order related activity on the driver’s or merchant’s part. The imputation strategy followed is to replace the missing variable with a version of the same variable calculated over a longer time period. If the data is

not available even in the longer time period, the missing values are passed to the prediction model.

The prediction model is built using XGBoost which is an efficient implementation of the Gradient Boosted Trees algorithm. Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. A gradient descent algorithm is used to minimize the loss when adding new models. This algorithm also applies several in-built regularization techniques and handles missing values in the input variables, if any.

The online order processing system 100 is coupled to a weather data system 140 which provides real time or near real time weather data. The weather data system 140 may provide the weather data as an application programming interface (API).

FIG. 3 is a flowchart showing a method of processing an online order according to an embodiment of the present invention. The method 300 shown in FIG.3 is carried out by the online order processing system 100 shown in FIG.2.

The method 300 shown in FIG.3 is carried out after an order is allocated to a delivery driver by the online order processing system 100, in the following description it is assumed that the order is initially allocated to the first delivery driver 170. When the order is allocated to the first delivery driver 170, an estimated time of delivery is calculated and this estimated time of delivery is sent to the customer device 152 by the network interface 114 of the online order processing system 100. The method 300 may be carried out multiple times during the course of an order. The method may be carried out at fixed time intervals, for example at 3 minutes, 6 minutes, 9 minutes and 12 minutes. The method can be extended for intervals beyond this.

In step 302, the online order processing system 100 receives current delivery driver information via the network interface 114. The current delivery driver information comprises data such as the speed and location of the first delivery driver device 172.

In step 304, the look up module 122 is executed by the processor 110 of the online order processing system 100 to look up history data. The history data comprises delivery driver data 134 corresponding to the first delivery driver 134 and merchant data 136 corresponding to the merchant at which the order was placed. In addition, in step 304, the look up module 122 may also look up data associated with the order from the order data 132. Examples of the historical data are shown in FIG.5B.

In step 306, the prediction module 124 is executed by the processor 110 of the online order processing system 100 to input current delivery driver and history data into the prediction model 138. An example of the data which may be input into the prediction model 138 is shown in FIG.5C.

In step 308, the prediction module 124 is executed by the processor 110 of the online order processing system 100 to determine the probability that the first delivery driver 170 will deliver the order within the estimated time of arrival.

If in step 308 it is determined that the first delivery driver 170 is unlikely to deliver the order within the estimated time of arrival, then the method moves to step 310 or 312. If it is determined in step 308 that the first delivery driver is likely to deliver within the estimated time of arrival, but there is a small probability that the driver will deliver outside the estimated time of arrival. For example if the probability is below a first threshold to update the estimated time of arrival but above a third threshold. Then the method moves to step 316.

In step 310, the order reallocation module 126 is executed by the processor 110 of the online order processing system 100 to reallocate the order to the second delivery driver 180. Step 310 may comprise sending a delivery request to all delivery drivers within a range of the merchant location.

In step 312, the estimated time of arrival update module 125 is executed by the processor 110 of the online order processing system to calculate an updated estimated time of arrival.

In step 314, the updated estimated time of arrival calculated in step 312 is sent to the customer device 152 by the network interface 114 of the online order processing system 100.

In step 316, a reminder notification is generated and sent to the first delivery driver.

If the order is wrongly assigned to another driver, this may result in complaints from the first delivery driver. Customers may also make complaints if orders are received after the estimated time of delivery. These complaints may indicate occasions in which the prediction model 138 incorrectly predicts the estimated time of arrival and may be used to update or retrain the predictive model 138.

FIG.4 is a flowchart showing a method of updating a prediction model according to an embodiment of the present invention. The method 400 shown in FIG.4 is carried out by the online order processing system 100 shown in FIG.2.

In step 402, a customer complaint indication is received by the network interface module 114 of the online order processing system 100. The complaint indication may indicate that an order was received outside the predicted estimated time of arrival. In step 408, the module update module 128 is executed by the processor 110 of the online order processing system 100 to update the prediction model 138.

In step 404, a driver complaint indication is received by the network interface module 114 of the online order processing system 100. The complaint indication may indicate that an order was incorrectly transferred away from a delivery driver. In step 406, the model update module 128 is executed by the processor to filter received complaints. The filtering of complaints may comprise interrogating the delivery driver data 134 to determine an indication of whether the delivery driver is trustworthy. For example, if the delivery driver has abandoned deliveries frequently in the past, and any fraud signal that is marked against the driver this may indicate that the delivery driver is not trustworthy. Once complaints from untrustworthy delivery drivers has been filtered out, the method proceeds to step 408. In step 408, the module update module 128 is executed by the processor 110 of the online order processing system 100 to update the prediction model 138. The model is retrained here by updating the records with the

new signal based on customer and driver complaints. For instance, if a high probability of complaint for a driver dispatch data was not predicted but the customer still complained - we tag this data as "inactive". In another case, for example if a driver was reallocated for an order dispatch but based on driver complaint it was found genuine then we retag this data as "driver active". These tagged datasets are added to the initial dataset and used to retrain the model.

FIG.5A to FIG.5C are tables showing features used in prediction models in methods of processing online orders according to embodiments of the present invention.

FIG.5A shows real time features which are used in the prediction models. As shown in FIG.5A, the real time features comprise time features which may be determined by a clock or calendar, driver attributes which may be received from the first driver delivery device 172 or the second driver delivery device 182, order attributes which may be received from the customer device 152 as part of the order and stored as order data 132, and weather data which is received from the weather data system 140.

FIG.5B shows batch or historic features which are used in the prediction models. As shown in FIG.5B, the batch features comprise driver attributes which are stored as delivery driver data 134 and merchant attributes which are stored as merchant data 136.

FIG.5C shows the ranked order of features used in a prediction model in an embodiment of the present invention. Features which are highly ranked are given a higher weighting than those with a lower rank. The ranking of the features may be determined during the training process such that features that have a high impact on the probability that a delivery driver will deliver an order within the estimated time of arrival are highly ranked while those with a lower impact have a corresponding lower ranking.

FIG.6A and FIG.6B are tables showing example scenarios of methods of processing online orders according to embodiments of the present invention.

FIG.6A shows attributes from an example in which the model output score is a low value (0.06) which results in the decision being that the order stays with the current delivery driver. FIG.6B shows attributes from an example in which the score is a high value which results in the decision being that the order is reallocated to another delivery driver. Here, the score represents the probability that the delivery driver will not complete the delivery within the estimated time of arrival.

Whilst the foregoing description has described exemplary embodiments, it will be understood by those skilled in the art that many variations of the embodiments can be made within the scope and spirit of the present invention.

CLAIMS

1. An online order processing method for processing an order, the order comprising an indication of a merchant location at which an order item is to be picked up and an indication of a delivery location at which the order item is to be delivered, the order having a current estimated time of delivery, the method comprising:

receiving current delivery driver information relating to a delivery driver assigned to deliver the order;

looking up delivery driver data relating to the delivery driver assigned to deliver the order and order variables relating to the order and the merchant;

inputting the current delivery driver information, the delivery driver data and the order variables into a statistical model trained on historical driver and order variables, the statistical model being configured to predict a probability that the delivery driver will deliver the order within an estimated time of delivery; and

determining an indication of the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery.

2. A method according to claim 1, further comprising:

if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is below a first threshold, determining a new estimated time of delivery and sending an indication of the new estimated time of delivery to a customer device associated with the order.

3. A method according to claim 1, further comprising:

if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is below a second threshold, reallocating the order to a second delivery driver.

4. A method according to claim 1, further comprising:

if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is above a first threshold and below a third threshold, sending a notification to the delivery driver assigned to deliver the order.

5. A method according to any preceding claim, further comprising receiving a complaint indication indicating that the delivery was not delivered within an estimated time of delivery and updating the statistical model.

6. A method according to claim 3, further comprising receiving a complaint indication indicating that the order was incorrectly reallocated and updating the statistical model.

7. A method according to any preceding claim, wherein the statistical model is a multivariate classification model.

8. A computer readable carrier medium carrying processor executable instructions which when executed on a processor cause the processor to carry out a method according to any one of claims 1 to 7.

9. An online order processing system for processing an order, the order comprising an indication of a merchant location at which an order item is to be picked up and an indication of a delivery location at which the order item is to be delivered, the order having a current estimated time of delivery, the online order processing system comprising a processor, data storage, and program storage, the data storage storing a statistical model trained on historical driver and order variables, the statistical model being configured to predict a probability that the delivery driver will deliver the order within an estimated time of delivery, the program storage storing computer program instructions operative by the processor to:

Receive current delivery driver information relating to a delivery driver assigned to deliver the order;

look up delivery driver data relating to the delivery driver assigned to deliver the order and order variables relating to the order and the merchant;

input the current delivery driver information, the delivery driver data and the order variables into the statistical model; and

determine an indication of the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery.

10. An online order processing system according to claim 9, wherein the program storage stores computer program instructions operative by the processor to: if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is below a first threshold, determine a new estimated time of delivery and sending an indication of the new estimated time of delivery to a customer device associated with the order.

11. An online order processing system according to claim 9, wherein the program storage stores computer program instructions operative by the processor to: if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is below a second threshold, reallocate the order to a second delivery driver.

12. An online order processing system according to claim 9, wherein the program storage stores computer program instructions operative by the processor to: if the probability that the delivery driver assigned to deliver the order will deliver the delivery within the current estimated time of delivery is above a first threshold and below a third threshold, send a notification to the delivery driver assigned to deliver the order.

13. An online order processing system according to any one of claims 9 to 12, wherein the program storage stores computer program instructions operative by the processor to: receive a complaint indication indicating that the delivery was not delivered within an estimated time of delivery and update the statistical model.

14. An online order processing system according to claim 11, wherein the program storage stores computer program instructions operative by the processor to: receive a complaint indication indicating that the order was incorrectly reallocated and update the statistical model.

15. An online order processing system according to any one of claims 9 to 14, wherein the statistical model is a multivariate classification model.