Abstract—In this paper, we describe our solution for ICDM 2010 Contest Task 2 (Jams), where the task is to predict future where the next traffic jams will occur in morning rush hour, given data gathered during the initial phase of this peak period. Our solution, which is based on an ensemble approach, finished Second in the final evaluation.

Keywords—ensemble; nearest-neighbor; cross validation;

I. INTRODUCTION

ICDM 2010 Contest asks researchers to devise algorithms that tackle problems of traffic flow prediction, for the purpose of intelligent driver navigation and improved city planning [1]. The data sets come from a highly realistic simulator of vehicular traffic, Traffic Simulation Framework (TSF), developed at the university of Warsaw. Simulations employ the map of the city of Warsaw, Poland, taken from OpenStreetMap project. The contest includes 3 tasks: Traffic, Jams and GPS. In this paper, we focus on the second task, which is to predict where the next jams will occur during main phase of the morning peak, given the data gathered during the initial phase of the peak. Such prediction could be used to warn drivers in advance, before the jams actually occur. In the given data sets, there are 35170 road segments defined by 18716 different nodes, of which only 8631 road segments are major roads whose maximum velocity is more than 60km/h, and the remaining 26539 road segments are minor roads whose maximum velocity is 60km/h.

The main challenges of this task can be summarized as follows. First, the data provided are imprecise in terms that they only contain ordered sequences of jammed streets. In other words, detailed measurements of speed and number of cars, as well as the time when the jam first occurs, are missing. Second, the data cover only major roads that constitute 25% of the whole road network. However, the missing information related to minor roads may influence jams on major roads.

In this task, data samples were generated from independent 1-hour long simulations, starting each time from an almost empty road network, fed subsequently with cars according to the selected distributions of start and destination points. As input, we are given 5 road segments with road work as well as a sequence of major roads where the first jams occurred during the initial 20 minutes of the simulation. The goal is to predict a sequence of major roads where the next jams will occur in the next 40 minutes. The sequences are ordered according to time of jam appearance and their length may vary between samples. Training and test sets consist of 5000 samples (hours) each. Let $A_i$ ($i = 1, \ldots, 5000$) denote the $i$th training hour, and $B_j$ ($j = 1, \ldots, 5000$) denote the $j$th test hour. The evaluation metric is based on the concept of Mean Average Precision (MAP) from the domain of information retrieval adapted to the specifics of this task. It takes the largest value when the predicted sequence is exactly the same as the target sequence and punishes any deviations from the target, such as different length of the sequence, different order of road segments, etc. Mistakes on early positions of the sequence are punished more than later ones. Overall result for the test set is calculated as an arithmetic average of values computed for each sample. Each competing team may submit solutions many times for the whole duration of the challenge. They are evaluated instantly after submission and the resulting MAP is visible to all teams. However, this result is only based on the preliminary set, which comprises 35% of the test data. And the final evaluation is based on the final set, which comprises the remaining 65% of the test data.

Our solution has MAP 0.5580 on the preliminary set, and 0.55793 on the final set, which finished in second place in the contest. Based on our experience, jammed roads in the remaining 40 minutes are not strongly dependent on the history, and a dominating effect is the overall frequency of a road being jammed. This observation is consistent with the good performance of the baseline, which always predicts the road being jammed. This observation is consistent with the good performance of the baseline, which always predicts the main phase of the morning peak, given the data gathered during the initial phase of this peak period. Such prediction could be used to warn drivers in advance, before the jams actually occur.

Our solution has MAP 0.5580 on the preliminary set, and 0.55793 on the final set, which finished in second place in the contest. Based on our experience, jammed roads in the remaining 40 minutes are not strongly dependent on the history, and a dominating effect is the overall frequency of a road being jammed. This observation is consistent with the good performance of the baseline, which always predicts the main phase of the morning peak, given the data gathered during the initial phase of this peak period. Such prediction could be used to warn drivers in advance, before the jams actually occur. In the given data sets, there are 35170 road segments defined by 18716 different nodes, of which only 8631 road segments are major roads whose maximum velocity is more than 60km/h, and the remaining 26539 road segments are minor roads whose maximum velocity is 60km/h.

The main challenges of this task can be summarized as follows. First, the data provided are imprecise in terms that they only contain ordered sequences of jammed streets. In other words, detailed measurements of speed and number of cars, as well as the time when the jam first occurs, are missing. Second, the data cover only major roads that constitute 25% of the whole road network. However, the missing information related to minor roads may influence jams on major roads.

In this section, we introduce our ensemble-based method. The ensemble fuses information from several base predictors in order to come up with a better combined predictor. It consists of the following components.

1) Base predictors;
2) Fusing scheme;
3) Sequence length predictor;
4) Evaluation framework.

By constructing base predictors from various aspects, we are able to collect useful information from different sources. By fusing the predictors in different ways, we are able to make the best out of each base predictor. By adaptively determining the length of jammed road sequence in the remaining 40 minutes, we are able to identify most of the true jammed roads with a few false alarms. And by using cross validation [2] as our evaluation framework, we are able
to pick a set of parameters which at least correspond to a local optimum and avoid over-fitting. Next, we will elaborate on each component in the following subsections.

A. Base Predictors

In our ensemble-base method, two types of base predictors have been employed, namely geographic propagation predictors and nearest neighbor predictors [2]. They calculate the likelihood of a road segment having jams in the remaining 40 minutes of a test hour, based on geographic information and jam information from similar training hours.

1) Geographic Propagation Predictors: The intuition of the geographic propagation predictors is to track where the jammed traffic will flow based on the connectivity of the road segments. To construct this type of predictors, we first build connectivity matrix $C$, which is $35170 \times 35170$. Its element $C(i,j)$ is equal 1 if and only if the end node of road segment $i$ is the same as the start node of road segment $j$; otherwise $C(i,j) = 0$. For each simulation hour in the test set, since the drivers know ahead of the time which road segments have road work, if road segment $i$ has road work, we would set the $i^{th}$ row and $i^{th}$ column of $C$ to 0. In this way, no vehicles would enter or exit such road segments. Furthermore, we construct vector $v_0$ ($1 \times 35170$) based on the jammed road segments in the initial 20 minutes. Its element $v_0(i)$ is set to 1 if and only if the $i^{th}$ road segment is jammed in the initial 20 minutes and 0 otherwise. By multiplying $v_0$ with $C$, we get $v_1$, i.e., $v_1 = v_0 \times C$. The nonzero elements of $v_1$ correspond to the road segments where the jammed vehicles might flow once they exit the jammed road segments in the first 20 minutes. Similarly, we define $v_2 = v_0 \times C^2$ and $v_3 = v_0 \times C^3$, whose nonzero elements indicate where the jammed vehicles might flow after 2 and 3 hops, respectively. $v_1$, $v_2$ and $v_3$ are incorporated into the ensemble as base predictors. Note that,

[1] $v_0$ is NOT used as a base predictor since road segments having jams in the initial 20 minutes cannot be included in the predicted sequence for the remaining 40 minutes;

[2] Based on our experiments, higher-order propagation vectors do not improve the performance. Therefore, they are not used as base predictors.

We also experimented with variants of $C$, such as the normalized version of $C$ whose row sums are 1 and the variant of $C$ whose elements reflect the minimum travel time from one road segment to another. However, we found the above binary version works the best based on cross validation, which will be introduced in Subsection II-D.

2) Nearest Neighbor Predictors: The intuition of the nearest neighbor predictors is to make use of the information from similar training hours to help predict the jammed roads in the remaining 40 minutes of a test hour. To construct this type of predictors, for each simulation hour in the test set, we first calculate its similarity to each simulation hour in the training set. In our experiments, the following similarity measurements have been used,

[1] $\text{MAP}_1$: the MAP similarity between the sequence of jammed roads in the initial 20 minutes of the training hour and the sequence of jammed roads in the initial 20 minutes of the test hour;

[2] $\text{MAP}_2$: the MAP similarity between the reverse sequence of jammed roads in the initial 20 minutes of the training hour and the reverse sequence of jammed roads in the initial 20 minutes of the test hour;

[3] $\text{Simple}_1$: the number of common jammed roads in the initial 20 minutes of the training hour and the initial 20 minutes of the test hour divided by the longer length of the two sequences;

[4] $\text{Simple}_2$: the number of common jammed roads in the remaining 40 minutes of the training hour and the initial 20 minutes of the test hour divided by the longer length of the two sequences;

[5] $\text{Simple}_3$: the number of common roads with road work in the training hour and the test hour divided by 5.

The first similarity measurement (MAP$_1$) is straight-forward. The second one (MAP$_2$) is based on the assumption that the road segments having jams later in the initial 20 minutes may have a bigger influence over the jammed roads in the remaining 40 minutes. The third similarity measurement (Simple$_1$) is a simplified version of MAP$_1$ in that it ignores the order of the sequences, and considers two simulation hours to be similar to each other if they share many common jammed roads in the initial 20 minutes. The fourth similarity measurement (Simple$_2$) is similar to Simple$_1$ except that for the training hour, we use the sequence of jammed roads in the remaining 40 minutes instead of the initial 20 minutes. This is because if the remaining 40 minutes of the training hour and the initial 20 minutes of the test hour share many jammed roads, then the order information hidden in the initial 20 minutes of the training hour may help rank the jammed roads in the remaining 40 minutes of the test hour. The last similarity measurement (Simple$_3$) considers two simulation hours to be similar to each other if they have common roads with road work. Surprisingly, when using each similarity measurement alone to predict jammed roads in the remaining 40 minutes of a test hour, Simple$_1$ performs the best, and combining all five of them gives even better performance. We also experimented with other similarity measurements, such as the MAP similarity between the sequence of jammed roads in the remaining 40 minutes of the training hour and the sequence of jammed roads in the initial 20 minutes of the test hour, but the above combination gives the best performance based on cross validation.

Once we have calculated the similarity between the $j^{th}$ test hour $B_j$ and each training hour, we find its $k$ nearest neighbors which correspond to the $k$ training hours with the maximum similarity to $B_j$. Let $A_{j1}, \ldots, A_{jk}$ denote these
neighbors. For each of these neighbors, we assign a score to the jammed roads in both the initial 20 minutes and the remaining 40 minutes. Take MAP$^1_1$ and $A_{j_1}$ as an example. Let MAP$^1_1(j, 1)$ denote the similarity between $B_j$ and $A_{j_1}$, and $l_{j_1}$ denote the total number of jammed roads in $A_{j_1}$. Then the $i^{th}$ jammed road is assigned the following score

$$
(x + \frac{y(l_{j_1} + 1 - i)}{2l_{j_1}}) \times \text{MAP}^1_1(j, 1)^d, \ i = 1, \ldots, l_{j_1}
$$

where $x$, $y$, and $d$ are three parameters to be tuned via cross validation. Then we add the scores from the $k$ nearest neighbors together to obtain the base predictor. The base predictors from the other similarity measurements can be obtained similarly. Note that $k$ is also tuned via cross validation, and different similarity measurements have different optimal values for $k$.

B. Fusing Scheme

To combine the nearest neighbor predictors, we have two options.

1) Combining the scores obtained from different similarity measurements in a linear fashion;

2) Combining the similarity measurements as a single one with which to obtain the base predictor.

Based on our experiments, the second option gives better performance, with MAP almost 1% more than the first option. Therefore, the second option is used in our method. The coefficients of different similarity measurements are as follows: 0.9 for MAP$^1_1$, 0.6 for MAP$^2_2$, 0.8 for Simple$^1_1$, 0.2 for Simple$^2_2$, and 1.5 for Simple$^3_3$, which are tuned via cross validation. In this way, we obtain only one base predictor based on nearest neighbors. For this predictor, we used 36 nearest neighbors, $x = 10$, $y = 1$ and $d = 4$.

Next, we combine the nearest neighbor predictor with geographic propagation predictors. Notice that for a test hour, each base predictor essentially assigns a score for each road segment, with a higher score indicating more chance of having jams in the remaining 40 minutes. In our method, we combine them in a linear fashion. The coefficients of different base predictors are as follows: 1 for nearest neighbor predictor, 0.044 for both $v_1$ and $v_2$, and 0.0011 for $v_3$. These coefficients are tuned via cross validation.

C. Sequence Length Predictor

After combining the base predictors, we get a score for each road segment. Then we rank the road segments with their scores from large to small. The road segments on the top of the list are likely to have jams in the remaining 40 minutes of the test hour. Next, we need to pick a cutoff point before which the road segments will be included in the predicted sequence. Here, our key observation is that, if we knew the length of the true sequence and used that as the cutoff point, the average MAP would be low. The reason may be explained as follows. In the predicted sequence, we tend to have some false alarms, and some road segments which actually have jams in the remaining 40 minutes fall beyond the cutoff point. Therefore, the cutoff point should be a little larger than the length of true sequence in order to include as many jammed roads as possible without having too many false alarms. In our experiments, for each test hour, we first calculate the average number of jammed roads in the remaining 40 minutes among its neighbors, and then multiply it by 1.13 to get the cutoff point. However, if the length of the true sequence is particularly large, this scheme tends to underestimate the cutoff point. Therefore, to address this problem, we make the following modification to the cutoff point. If the score of the road segment at the cutoff point is larger than 0.19, which often indicates a long true sequence, we would extend the predicted sequence until the score of the road segment is less than 0.19.

D. Evaluation Framework

Our method involves 15 parameters. A simple way to tune these parameters is to generate output files with different parameter settings, upload the files, and use the MAP value returned by the server to determine the optimal setting. There are two potential problems with this evaluation scheme. First, each team is only allowed 100 submissions, which is clearly not enough for such a huge parameter space. Second and more seriously, the result returned by the server is only based on the preliminary set, which comprises 35% of the test data. Therefore, by greedily adjusting the parameters to fit the preliminary set, we are prone to over-fitting. In other words, good performance on the preliminary set does not guarantee good generalization performance on the final set, which comprises 65% of the test data. To address this problem, we use 10-fold cross validation on the training set as the evaluation framework. It is often the case that the result returned by the server on the preliminary set is very close to our result from cross validation. For example, our final solution has MAP 0.5573 based on cross validation in the training set, 0.5580 based on the preliminary set, and 0.55793 based on the final set. In this way, we effectively avoid over-fitting.

III. CONCLUSION

In this paper, we described our ensemble-based method for Task 2 Jams. It incorporates information from various base predictors, combines them in two different ways, adaptively determines the optimal sequence length, and is evaluated via cross validation on the training set. The good performance on the final test set proves its effectiveness.

ACKNOWLEDGMENT

We thank Yan Liu for helpful discussions, and we thank the contest organizers for such a fun competition.

REFERENCES
