Integrated Framework for Real-time Urban Network Travel Time Prediction on Sparse Probe Data

Matej Cebečauer1, Erik Jenelius1, Wilco Burghout1

1 KTH Royal Institute of Technology, Department of Transport Science, Teknikringen 10, SE-100 44 Stockholm, Sweden

Abstract: The paper presents the methodology and system architecture of an integrated urban road network travel time prediction framework based on low frequency probe vehicle data. Intended applications include real-time network traffic management, vehicle routing and information provision. The framework integrates methods for receiving a stream of probe vehicle data, map-matching and path inference, link travel time estimation, calibration of prediction model parameters, and network travel time prediction in real-time. The system design satisfies three crucial aspects: computational efficiency of prediction; internal consistency between components; and robustness against noisy and missing data. Prediction is based on a multivariate hybrid method of Probabilistic Principal Component Analysis (PPCA), which captures global correlation patterns between links and time intervals, and local smoothing, which considers local correlations among neighboring links. Computational experiments for the road network of Stockholm, Sweden and probe data from taxis show that the system provides high accuracy for both peak and off-peak traffic conditions. The computational efficiency of the framework makes it capable of real-time prediction for large-scale networks. For links with large speed variations between days, prediction significantly outperforms the historical mean. Furthermore, prediction is reliable also for links with high proportions of missing data.

1 Introduction

Accurate short-term prediction of traffic conditions is important for traffic control, traveler information provision, real-time vehicle routing and trip planning, etc. For many of these applications, prediction across a wide network is desirable. On the other hand, this also requires a wide network of sensors for measuring traffic conditions. GPS devices in vehicles or smart phones allow the collection of traffic data in urban road networks at low marginal costs, which makes them highly valuable as opportunistic traffic sensors [1].

However, a number of limitations make the use of probe vehicle data to support traffic management decisions challenging [2, 3]. Systems are often designed to minimize the data transfer demand, by reporting only the location of vehicles once every one or two minutes. GPS data include errors with respect to the real geographical coordinates, and the probes need to be map-matched to network links. Finding the actual trajectory of a vehicle based on sparse probe data can be challenging especially in dense urban areas. Methods for map-matching and path inference in such settings are presented in [4–7].

With the increasing availability of probe data, the literature on arterial travel time estimation has grown recently [8–10]. Methods for link and route travel time estimation based on sparse probe vehicle data are proposed in [11–13].

Meanwhile, most research on short-term travel time prediction has focused on motorways and major arterials [14]. Naive methods such as the historical mean or instantaneous travel time without model assumptions are easily implemented and computationally effective and, therefore, widely used in practice [15]. Commonly applied models also include artificial neural networks [16, 17]. However, while these models are behaviorally rich and can capture driver response to traffic information, they are complex and their calibration and application is challenging.

Relatively few papers consider short-term prediction in the context of large urban networks and applications such as network traffic management, vehicle routing, trip planning, etc. One proposed framework considers a dynamic Bayesian network model of the spatio-temporal dependencies [19]. The traffic state of the next time interval is influenced only by the current state of the neighboring links. Another method is a spatio-temporal extension of k-nearest neighbors [20]. A methodology for network-wide travel time prediction combining advantages of Probabilistic Principal Component Analysis (PPCA) and local smoothing is proposed in [21]. Spatio-temporal correlation patterns are estimated from historical data based on an efficient EM algorithm for handling missing data.

Implementation of travel time prediction on sparse probe data requires a combination of methods for receiving streams of probe data, map-matching and path inference, link travel time estimation, and finally link travel time prediction including calibration of model parameters. This raises a number of methodological challenges that have not yet received much attention in the literature. Each step in the chain must be consistent with the output of the previous step and the input of the next step. Furthermore, each step needs to be sufficiently efficient to be run in real-time, and robust to missing values and measurement noise.

This paper proposes a methodology and system architecture for urban network travel time prediction based on probe data. To the best of our knowledge, this is the first presented integrated framework which considers all necessary computation steps. The framework is versatile in terms of the data sources and the road network to which it is applied. The hybrid PPCA methodology proposed in [21] for network-wide short-term prediction is adopted and extended to neighboring links in order to capture both network-wide and local spatio-temporal correlation patterns.

The prediction methodology is applied to the inner city network of Stockholm, Sweden, containing nearly 2,000 links, based on probe data from taxis. Prediction accuracy is evaluated against naive methods including the historical mean, and the computational efficiency
Table 1  Notation

| $k$ | link index          |
| $i$ | time-of-day interval index |
| $h$ | current time-of-day interval in prediction |
| $n$ | day index            |
| $j$ | probe observation index |
| $K$ | number of links in prediction network |
| $F$ | number of intervals in prediction horizon |
| $P$ | number of intervals to base prediction on |
| $I$ | set of days in time-of-days intervals |
| $N_E$ | set of days in model estimation set |
| $N_C$ | set of days in calibration set |
| $N_T$ | set of days in test set |
| $D$ | number of observed model variables |
| $J_{kn}$ | set of probe vehicle observations |
| $d_k$ | link length |
| $A_k$ | set of links adjacent to link $k$ |
| $(x_j, y_j)$ | GPS coordinates of probe |
| $o_j$ | offset on link from map-matching |
| $\tau_j$ | vehicle travel time from map-matching |
| $\pi_k$ | prior travel time used in estimation |
| $t_{ikn}$ | observed, predicted mean travel time |
| $v_{ikn}$ | observed, predicted space-mean speed |
| $v_{in}$ | $K \times 1$ vector of observed, predicted space-mean speeds of all links |
| $u_{ikn}$ | observed, predicted logarithm of space-mean speed |
| $u_{in}$ | $K \times 1$ vector of observed, predicted logarithm of space-mean speeds |
| $W, \mu, \sigma^2, Q, \alpha, \beta$ | parameters of the travel time prediction method |

2.2.1 Stream Processing Infrastructure: The first component receives and handles the incoming stream of probe data. The map-matcher module projects the probes to the digital road network considering the most likely vehicle paths (Section 2.3). The projected vehicle trajectories are then stored in the database.

2.2.2 Background Processes: Component (ii) consists of two modules running in the background at specified time intervals. A travel time estimation module runs every 15 minutes in the current implementation (Section 2.4). The module takes projected vehicle trajectories from the most recent 15-minute interval to estimate mean link travel times for the network for this particular interval. The estimated link travel times are then stored in the historical database as well as in the real-time database. The real-time database is only keeping data for a certain number of past intervals within the current day.

The task of the calibration module is to calibrate the parameters of the real-time prediction method (Section 2.5). The module can be run relatively infrequently, e.g., once per month, to update prediction parameters over a sliding time window based on historical data; the appropriate window length and updating frequency would depend on the dynamics of the particular network. Because of the separation between model calibration and prediction, current-day link travel times can be predicted in real-time.

2.2.3 Real-time Services: The real-time modules of component (iii) receive lists of links for which travel times are predicted. The travel time prediction module combines historical correlation patterns represented by calibrated model parameters and current-day travel time observations to predict future time intervals (Section 2.4). Finally, service modules adapt the results in an appropriate way for client applications (e.g., vehicle routing or journey planner applications).

2.2 Map-matching

Probe vehicle data consists of timestamped position reports $p_j = (q_j, s_j, (x_j, y_j))$ from a fleet of vehicles traversing the road network, where $q_j$ is a unique vehicle identifier, $s_j$ is a timestamp, and $(x_j, y_j)$ are the GPS coordinates of the vehicle location at time $s_j$. Low-frequency probe data require preprocessing to be useful for travel time estimation: reported positions $(x_j, y_j)$ must be matched to the digital road network, and the paths taken by the vehicles between probes must be inferred. The framework is based on the map-matching and path inference methodology, which considers delay penalties for traffic signals and left turns, described in [5]. The method is well suited for real-time application as it can process one probe at a time without loss of accuracy, and has been shown to be robust against low sampling frequencies [5].

In the map-matching stage, a set of candidate links in the vicinity of each probe is identified, and a projected point on each link is found. The appropriate size of the search region is network location dependent and varies across days on prediction performance.

The paper is organized as follows. The prediction framework is presented in Section 2. Section 3 introduces the implementation and case study, with results reported in Section 4. Section 5 concludes the paper.
probes, respectively. The probe vehicle observations are stored in a database.

2.3 Link Travel Time Estimation

The inferred probe vehicle observations are used to continuously estimate the mean travel times of the network links during the most recent 15-minute interval. A challenge with low-frequency probe data is that vehicle trajectories often cover multiple links, and sometimes only fractions of links. The framework is based on the non-parametric travel time estimation methodology proposed in [12], which is applicable to links as well as longer routes and can estimate many statistics beyond the mean travel time such as variance and percentiles.

Given current time-of-day interval \(i\) and current day \(n\), the set \(J_{ikn} = \{ j \mid i_j = i, k \in \pi_j, n_j = n \}\) of probe vehicle observations having some overlap with each link \(k\) is retrieved from the probe vehicle observations database. Furthermore, it is assumed that for each network link \(k\) and interval \(i\) a time-dependent prior travel time \(t_{0ik}\) is known. The estimation method consists of a sequence of computationally efficient steps: transformation, weighting, and aggregation. The first step transforms probe vehicle observations that only partially overlap with the link, or traverse additional links, to observations of the link travel time. The step consists of two main processes, allocation and scaling, based on the prior link travel times \(t_{0ik}\). Let \(\rho_{jk}\) be the fraction of link \(k\) covered by observation \(j\), computed from the link length \(d_k\), the path \(\pi_j\) and the offsets \(o_j, o_{j+1}\). The travel time allocated to link \(k\) is

\[
\tau_{jk}' = \phi_{jk} \tau_j = \frac{\rho_{jk} t_{0jk}}{\sum_l \rho_{jl} t_{0il}} \tau_j, \quad j \in J_{ikn} \tag{1}
\]

which is then scaled up to the whole link \(k\) as

\[
T_{jk} = \frac{1}{\rho_{jk}} \tau_{jk}' \tag{2}
\]

Each observation is then weighted according to its relevance as a link travel time observation (inversely proportional to the amount of transformation required),

\[
\omega_{jk} = \phi_{jk} \rho_{jk} \tag{3}
\]

An observation perfectly overlapping the link has the maximum weight 1, while observations covering less of the link and/or more of the adjacent network are weighted lower.

The final step is to aggregate all observations and calculate the weighted mean travel time for the link,

\[
t_{ikn} = \frac{\sum_j \in J_{ikn} \omega_{jk} T_{jk}}{\sum_j \in J_{ikn} \omega_{jk}} \tag{4}
\]

The output from estimation is the mean travel time \(t_{ikn}\) for every link \(k = 1, \ldots, K\), current time-of-day interval \(i\) and current day \(n\). Links without probe vehicle observations are treated as missing values. The travel times are stored in a real-time database with the most recent time intervals from the current day, as well as a historical database with a long time sequence of network travel time observations.

2.4 Travel Time Prediction

To predict link travel times in future time intervals, estimated link travel times \(t_{ikn}\) are used for two purposes. First, the historical

![Fig. 1: Architecture of the integrated real-time network travel time prediction framework.](image-url)
database is used to estimate and calibrate parameters of a network travel time model (Section 2.3). Second, recent current-day data are input to the model to predict link travel times in the near future (15–60 minutes ahead). The framework is based on the prediction methodology proposed in [21], which is designed to be computationally efficient for large networks and robust to missing data and noisy observations.

2.4.1 Network-wide Travel Time Model: Link travel times are correlated in both space and time; traffic conditions on a particular link are often related to recent traffic conditions in the nearby network. These correlations inform network travel time prediction. For large-scale urban road networks, however, the covariance matrix between links and time intervals is huge, and direct estimation with sparse probe data containing noise and missing observations is not feasible.

PPCA is a probabilistic model that assumes that observed variables are generated from a smaller number of latent variables [23]. Let $d_k$ denote the length of link $k$ and let $v_{ikn} = d_k/s_{ikn}$ be the space-mean speed of link $k$ according to Edie’s definition [24], assuming time intervals are sufficiently long in relation to the probe sampling frequency. The speeds of all links in interval $i$ and day $n$ are stored in the $K \times 1$ vector $\mathbf{v}_{in}$. Here, the logarithm $\log \nu_{nk} = \log \nu_{ikn}$ of the space-mean speed is used as the model variable.

The prediction horizon is specified by a parameter $F \geq 1$. Prediction is based on current-day observations from the $P \geq 1$ most recent intervals. Given that the current time interval is $h$, link speeds are thus predicted for time intervals $\{h + 1, \ldots, h + F\}$, and predictions are based on observations from interval $h - P + 1$ to interval $h$. Here, $h - P + 1$ and $h + F$ should be interpreted modulo $I$, i.e., the intervals may extend past midnight to the previous and the next day, respectively. The number of future intervals $F$ is an input parameter to the analysis, while $P$ is a model parameter that is calibrated to minimize prediction error.

The observations are assumed to be generated from the PPCA model,

$$
\begin{pmatrix}
\mathbf{u}_{h-P+1,n} \\
\vdots \\
\mathbf{u}_{h+F,n}
\end{pmatrix} = 
\begin{pmatrix}
\mu_{h-P+1} \\
\vdots \\
\mu_{h+F}
\end{pmatrix} + 
\begin{pmatrix}
W_{h-P+1} \\
\vdots \\
W_{h+F}
\end{pmatrix} \mathbf{x}_n + \mathbf{\epsilon}_n + \mathbf{1}_{n} \sigma^2
$$

or more compactly,

$$
\mathbf{u}_n = \mathbf{\mu} + W \mathbf{x}_n + \mathbf{\epsilon}_n
$$

Here $\mathbf{x}_n$ is a $Q \times 1$ column vector of latent random variables, assumed to be i.i.d. Normal with mean zero and variance one. $W$ is a $D \times Q$ parameter matrix representing a linear mapping between the latent variables and the observed variables, where $D = K(P + F)$; $\mathbf{\epsilon}_n$ is a $D \times 1$ column vector of measurement noise, assumed to be i.i.d. Normal with mean zero and variance $\sigma^2$. $Q$ is a model parameter; it is typically assumed that $Q < D$, i.e., that the observed variables are generated from a lower-dimensional latent space. $\mathbf{\mu}$ is a $D \times 1$ parameter vector that represents the mean logarithm of speed for each link and interval.

At time interval $h$ on day $n$ only variables from time intervals $\{h - P + 1, \ldots, h\}$ are observed, while the variables in time intervals $\{h + 1, \ldots, h + F\}$ are to be predicted. Define $\mathbf{u}_{P,n}$ as the vector of observed variables,

$$
\mathbf{u}_{P,n} = 
\begin{pmatrix}
\mathbf{u}_{h-P+1,n} \\
\vdots \\
\mathbf{u}_{h+F,n}
\end{pmatrix}
$$

and similarly for the mapping matrix $W_P$ and historical mean vector $\mathbf{\mu}_P$. Future variables $\mathbf{u}_{P,n}, i = h + 1, \ldots, h + F$, are predicted conditionally on the past observed values $\mathbf{u}_{P,n}$. The point predictor for interval $i$ on current day $n$ is

$$
\mathbf{u}_{in} = \mathbf{\mu}_i + W_i(W_P^T W_P + \sigma^2 I)^{-1} W_P^T (\mathbf{u}_{P,n} - \mathbf{\mu}_P)
$$

The predictor for space-mean link speed is obtained as $\hat{v}_{in} = \exp(\mathbf{u}_{in})$, and the predicted travel time for every link $k$ is $l_{ikn} = d_k/\hat{v}_{ikn}$. The historical mean $\mathbf{\mu}_i$ is the base value for the prediction, and the mapping to the latent factors are applied to the residuals between observations $\mathbf{u}_{in}$, $\mathbf{\mu}_i$ and $\mathbf{\mu}_P$.

If the observations contain missing values, arising from lack of probe vehicle observations, prediction is performed after removing the corresponding rows of $\mathbf{u}_{in}$, $\mathbf{\mu}_i$ and $\mathbf{\mu}_P$. This allows travel times to be predicted even for links entirely missing current-day observations.

2.4.2 Hybrid Network-wide and Local Prediction: The PPCA method captures network-wide correlation patterns across days, but is less well suited to capture more local correlations between consecutive time intervals. For this purpose, a hybrid PPCA and local smoothing methodology introduced in [21] is adopted here. The methodology is extended by taking correlations between neighboring links into consideration.

The PPCA model can estimate the mean values in past intervals conditional on the current-day observations, by applying Eq. (9) to past interval $i \in \{h - P + 1, \ldots, h\}$. The local smoothing updates the predicted speed of the pure PPCA method $\mathbf{v}_{in}$ for future interval $i$ and link $k$ with the sum of smoothed residuals between observed link speeds $\mathbf{v}_{jn}$ and PPCA estimated mean values $\hat{\mathbf{v}}_{jn}$ for the past intervals $j = h - P + 1, \ldots, h$. Let $A_k$ denote the set of neighboring links to link $k$. Define $U$ as the $K \times K$ normalized adjacency matrix where element $U_{ik}$ equals $1/|A_k|$ if $l \in A_k$ and 0 otherwise. Thus, the hybrid PPCA and local predictor $\mathbf{v}_{in}$ for future interval $i$ on current day $n$ is

$$
\mathbf{v}_{in} = \mathbf{v}_{in} + \frac{\alpha}{F} \sum_{j=h-P+1}^{h} (\mathbf{v}_{jn} - \hat{\mathbf{v}}_{jn}) \\
+ \frac{\beta}{F} \sum_{j=h-P+1}^{h} U(\mathbf{v}_{jn} - \hat{\mathbf{v}}_{jn})
$$

where $\alpha$ and $\beta$ are two model parameters that control the influence on a link of the link itself and the neighboring links, respectively. Missing values are handled in a straightforward way by modifying the mean over past time intervals for each link to only cover intervals with observations. Note that the pure PPCA prediction is recovered by setting parameters $\alpha = 0$ and $\beta = 0$ in Eq. (9).

The predicted space-mean link speeds are stored in a database to be retrieved by services such as vehicle routing and journey planner applications.

2.5 Calibration of Prediction Model Parameters

Prediction model parameters $P, Q, W, \mu, \sigma^2$, $\alpha$ and $\beta$ are updated relatively infrequently (e.g., once per month) over a sliding window based on the historical database of estimated link travel times. Separate parameters are calibrated for each time-of-day interval and are stored in a database.

The desired prediction horizon $F$ is first set. For given values of $Q$ and $P$, an efficient EM algorithm for cases with incomplete data in the observation vectors is developed in [25], and is used here to handle missing values in the historical data. For each time-of-day interval, parameters $P, Q, \alpha$ and $\beta$ are calibrated using cross-validation on the historical data (see further Section 3.3). When the best parameter values have been selected, the historical mean $\mu$, observation noise $\sigma^2$ and factor loadings $W$ are estimated on the historical data.
since more than 80% of link observations (across all days and intervals) are missing for class 5. The prediction road network consists of 1,886 links (functional class 1: 248 links, class 2: 79 links, class 3: 1,118 links, and class 4: 441 links). Figure 2 shows the speed limit and functional classes of all links in the prediction network.

The travel time prediction framework described in Section 2 has been implemented using a 40 Intel Xeon E5-2660 2.20 GHz cores server, with 65 GB of computer memory and Ubuntu 14.04 64-bit as the operating system. The typical usage of memory for the case study when calibrating parameters is under 8 GB.

PostgreSQL with the PostGIS extension is used to store all data. Map-matching and travel time estimation modules are implemented in Java. Python 3.4 64-bit with Matlab runtime library is used to handle calibration and real-time travel time prediction. All modules are run fully independently and the outputs are stored in the database. Thus, each module can run in parallel with multiple instances handling different subsets of data to achieve better performance. This is especially beneficial for large-scale urban networks.

### 3 Case Study

#### 3.1 System Implementation

The travel time prediction framework described in Section 2 has been implemented using a 40 Intel Xeon E5-2660 2.20 GHz cores server, with 65 GB of computer memory and Ubuntu 14.04 64-bit as the operating system. The typical usage of memory for the case study when calibrating parameters is under 8 GB.

PostgreSQL with the PostGIS extension is used to store all data. Map-matching and travel time estimation modules are implemented in Java. Python 3.4 64-bit with Matlab runtime library is used to handle calibration and real-time travel time prediction. All modules are run fully independently and the outputs are stored in the database. Thus, each module can run in parallel with multiple instances handling different subsets of data to achieve better performance. This is especially beneficial for large-scale urban networks.

#### 3.2 Data and Evaluation

The implemented framework is applied to the road network of the northern inner city of Stockholm, Sweden. In the digital road network each link has a number of attributes, including length, presence of a traffic signal, functional road class, and speed limit. The functional class categorizes roads from class 1 (major roads) to class 5 (minor service or side streets). For map-matching and path inference, the complete inner city of Stockholm (with 17,250 links and 1,118 links, and class 4: 441 links). Figure 2 shows the speed limit and functional classes of all links in the prediction network.

Prediction is considered for one 15-minute time interval into the future. All 96 15-min intervals from 0:00–0:14 to 23:45–23:59 are investigated. Four prediction methods are evaluated:

- Historical mean
- Local smoothing
- PPCA
- Hybrid PPCA with local smoothing

Probe vehicle data are received from ca. 1,500 taxis operating in the area. Each vehicle reports on average once every two minutes its id, GPS position, time-stamp and information whether it is hired or not. The case study uses all valid observations, independent of taxi status. Observations representing average speeds lower than 3 km/h, including taxis waiting at ranks, or higher than 140 km/h are filtered out. Previous studies have shown that both map-matching and travel time estimation methods perform well based on this source of probe vehicle data.

One year of data (year 2014) is used. Saturdays, Sundays and all holidays as well as school holidays are removed from the data set. After this filtering, data from 179 days are used. Of these, 30 days are randomly selected as a calibration set $N_C$, used to calibrate prediction model parameters $P$, $Q$, $\alpha$ and $\beta$. Further, 30 randomly selected days (set $N_T$) are used for the prediction performance evaluation and in Section 4. The remaining 119 days are used as the training set $N_E$ for the PPCA model to estimate parameters $W, \mu$ and $\sigma^2$ given calibrated $P$ and $Q$.

Figure 3(a) shows average network speed and speed variations across the day. Congestion is apparent for a number of intervals. The highest average network speed is around 35 km/h; during peak hours the speed falls below 25 km/h. The standard deviation of the network-wide average speed (dashed lines in Figure 3(a)) shows that the variability across days is usually small, around 1–2 km/h. The low network-wide variability suggests that speed at an aggregate level is similar across days. Furthermore, the dotted lines in Figure 3(a) show the average standard deviation of individual link speed across days, which is larger than the standard deviation for the network as a whole. This variability may be caused partially by measurement noise and partially by traffic signals, intersections, and variations in traffic patterns across days.

Two link properties and their impact on prediction accuracy are investigated in more detail: the fraction of missing observations across all links the $P$ past intervals, and the link speed variability across days in terms of the relative mean absolute difference (RMD). The RMD for interval $i$ and link $k$ considers the differences between all days in the model estimation data set $N_E$.

$$RMD(i, k) = \frac{\sum_{n \in N_E} \sum_{m \in N_E} |v_{ikm} - \bar{v}_{ik}|}{|N_E|^2 \bar{v}_{ik}}$$

(10)

Further, the mean link speed variability for interval $i$ across all links and particular set of days is denoted as $RMD(i)$. It is expected that the accuracy of all prediction methods decreases with the variability of link speed. Low variability suggests that actual values with high probability are close to the historical mean. On the other hand, we expect the PPCA methods to outperform the historical mean when the variability is larger, since they take into account the historical correlation patterns that can detect such variations.
3.3 Model Calibration

The combination of parameters $P$, $Q$, $\alpha$ and $\beta$ producing the best overall prediction accuracy across the calibration set $N_C$ is selected. Prediction accuracy is evaluated against measured link travel times for days in the calibration set $N_C$ in terms of the Mean Absolute Error (MAE), defined for interval $i$ as

$$\text{MAE}(i) = \frac{\sum_{k=1}^{K} \sum_{n \in N_C} |v_{ikn} - \hat{v}_{ikn}|}{K|N_C|}$$

(11)

Parameter $P$ is limited to a range of reasonable values 1, 2, …, 4, and $Q$ to the range 1, 2, …, 10. The $\alpha$ and $\beta$ parameters are restricted to only seven values 0, 0.05, …, 0.25, 0.3 with regular step 0.05, in order to decrease computational time.

Experiments show that care should be taken to avoid over-fitting of the model. Calibration of separate parameters for each link reduces the average prediction error of hybrid PPCA to only 1–2 km/h on the calibration set $N_C$. For the test set $N_T$, however, the accuracy is about 2–4 km/h worse on average compared to when common parameters for the network as one whole are calibrated.

Parameter $Q$ tends to be higher when the fraction of missing observations is small and when link speed variability across days is large. In other words, the PPCA model is able to find more useful correlation patterns for prediction in the historical data. Parameter $P$ is also decreasing with the fraction of missing observations. Thus, with a lot of observed values it is not necessary to consider many past current intervals for accurate prediction. For values of calibrated parameters across all intervals, see Table S1 in the supplementary information file.

Parameter $\beta$ is calibrated to zero for most time intervals. However, $\beta$ tends to be larger than zero when the fraction of missing observations is higher. In such situations, the calibration uses neighboring links to increase the number of current-day observations in the local smoothing. On the other hand, the parameter $\alpha$ is lower when the fraction of missing observations is large.

4 Results

4.1 Prediction Performance

The dataset is split into the training set ($N_E$), the calibration set ($N_C$), and the evaluation set ($N_T$). Sets $N_E$ and $N_C$ are used to train and calibrate the models. Prediction accuracy is evaluated against measured link travel times for days in the test set $N_T$ in terms of the Mean Absolute Error (MAE), defined for link $k$ and interval $i$ as

$$\text{MAE}(i,k) = \frac{\sum_{n \in N_T} |v_{ikn} - \hat{v}_{ikn}|}{|N_T|}$$

(12)

Results for the Mean Absolute Percentage Error (MAPE) lead to analogous conclusions and are located in the supplementary information file (Table S4). Due to the noisy nature of the probe data measurements, residuals when compared to measurements are positive even for the most accurate prediction methods.

Figure 3(a) shows the average prediction performance across all links for each 15-minute interval; numerical results are available in the supplementary information file (Table S2). The average prediction accuracy is around 3.5 km/h throughout most of the day except the afternoon peak with high day-to-day variability and the early morning with very few measurements (the maximum prediction error, which is highly sensitive to outliers in the measurements, is around 50 km/h throughout the day).

In general, averaged across all links, the hybrid PPCA method has the highest prediction accuracy compared to the historical mean and the pure PPCA method. Furthermore, the use of smoothing parameters $\alpha$ and $\beta$ improves prediction accuracy relative to pure PPCA in 80% of cases, and allows the hybrid PPCA to be better than the historical mean in 90% of all time intervals.

Pure PPCA performs better than the historical mean particularly when the fraction of missing observations is large. The largest improvement of the hybrid PPCA with respect to the PPCA and historical mean is in the afternoon peak intervals 14:00–19:00, when the
Fig. 4: Mean absolute error (MAE) of predicted vs. measured link speeds across all time intervals. (a) All case study links. (b) Links classified as functional class 2.

link speed variability across days is the highest. Both PPCA methods usually outperform the local smoothing.

Figure 4(b) considers only links classified as functional class 2. Numerical results are available in the supplementary information file (Table S3). The links speed variability is the highest for functional class 2 while the fraction of missing observations is within reasonable limits (see Figure S1(b) in supplementary information file). Here, the hybrid PPCA in some cases outperforms the historical mean by more than 40%.

Figure 5(a) visualizes predicted link speeds for an afternoon peak interval (17:15–17:30) on a particular test day (January 15, 2014) in terms of the relative difference between predicted speed and free-flow speed. This congestion index highlights bottlenecks as dark/red links, while links with moderate congestion are indicated with light colors. Figure 5(b) shows the accuracy of the hybrid PPCA method in terms of residuals to observed link travel times. Residuals have a tendency to be larger close to intersections and traffic signals.

The hybrid PPCA method significantly outperforms the historical mean when the average RMD of all case study links speed between days is larger than 0.40 (Figure 6(a)). Figure 6(b) shows that prediction accuracy in general is robust against the fraction of missing observations. However, it is important to note that the effect of travel time variability and missing observations varies a lot at the link level. The uncertainties in the input data, including link speed estimation accuracy, map-matching, traffic signals, etc., impact the input data with measurement noise that can influence accuracy independently from the speed variability and missing observations. Figures 6(c)-(d) consider only links classified as functional class 2. These links experience more day-to-day variations in congestion and travel times, which leads to the highest benefits of the PPCA methods. The results in Figure 6(d) show that missing observations may have a more negative effect on prediction performance.

4.2 Computational Efficiency

With the described system implementation, map-matching all probes received during 15 minutes takes about 20 seconds on a single core. Travel time estimation for all links in the case study area (4,480 links including functional class 5) based on the most recent 15 minutes of map-matched probe data takes about 4 seconds using only one core. The computational time is shown in Figure 7(a) to scale well as a function of problem size (number of probes).

Travel time prediction with the hybrid PPCA method takes only 0.6–1 second for all 1,886 links of the prediction network, while pure PPCA usually takes under 0.2 seconds. The computational time increases approximately linearly with the network size, as shown in Figure 7(b). Thus, the methodology scales well with the problem size.

Calibration of model parameters based on the historical database is the most time consuming part of the framework. In the Stockholm case study application of the hybrid PPCA to 1,886 links takes 10–30 hours while pure PPCA usually needs less than 10 hours per interval to calibrate P and Q and estimate μ, σ² and W. Figure 7(a) shows the increase of the computational time with the network size.
Fig. 5: Visualization of prediction results for 17:15-17:30, January 15, 2014. (a) Congestion index \((1 - (\text{predicted speed} / \text{free-flow speed}))\), (b) Prediction residuals.

Fig. 6: Impact of (a,c) link speed variability between days (RMD) and (b,d) fraction of missing observations on average prediction accuracy (MAE) of the hybrid PPCA and historical mean. (a,b) Mean values are computed across all links and (c,d) across links classified as functional class 2. Afternoon peak intervals (14:00–19:00).

Calibrating only \(\alpha\) for the local smoothing method requires less than one second. However, calibration is run relatively infrequently (e.g., once a month) and thus computational time is not critical (within reasonable limits). Calibration tends to consume more time in intervals
with larger fractions of missing observations and greater travel time variability (see Figure S2 in the supplementary information file).

5 Conclusions

This paper presented the architecture of a system for real-time urban network travel time prediction based on probe data, involving map-matching and travel time estimation as well as prediction in one integrated framework. The system is designed to be computationally efficient even for large networks, internally consistent, and robust against noisy and missing data. Correlation patterns are extracted from historical travel time data using a hybrid method of PPCA and local smoothing, and calibrated model parameters are stored in the database. Network travel times are predicted in real-time by combining current-day data and the correlation patterns stored in the estimated parameters.

The implementation for the Stockholm network showed that the hybrid PPCA method produces the most precise and robust travel time predictions of the presented methods. For most times of the day, the historical mean can be considered as sufficient. However, PPCA has significant advantages for links and time intervals where the speed variability across days is large. This is especially valuable since link speed variability tends to be larger during peak hours, when prediction accuracy is the most important. The results also show that the prediction methods are robust against missing observations, which is important for applications with sparse probe data.

The proposed framework produces predictions for all 1,886 links of the case study network from unprocessed probe data in around 25 seconds per 15-minute interval when run on a single core. Experiments demonstrate the scalability of the system for real-time prediction for large-scale networks. Although calibration takes significantly more computational time, this process runs relatively infrequently (e.g. once a month) and is therefore not time critical.

It has been shown that partitioning the network into subnetworks based on e.g. functional classes may increase prediction accuracy [21]. In future work, more sophisticated methods such as estimating parameters for clusters of links and days may help to further utilize the full potential of hybrid PPCA method.

Under anomalous conditions such as special events or severe weather conditions, data-driven approaches in general require that similar events are captured in the historical data set for accurate prediction. If such data exist, the method is able to incorporate the corresponding covariation pattern and provide more accurate predictions for the current day. Further, through the deviations between observed and predicted travel times the framework offers a signal about the validity of prediction based on historical patterns. A direction for further development is the combination of data-driven and simulation models for traffic prediction [17, 18] in problematic conditions or regions.

6 Supporting information

File S1 Supplementary information file which includes figures and tables with the numerical values of the defined performance indicators.

7 Acknowledgment

The work was funded in part by the Swedish Transport Administration through the Mobile Millennium Stockholm project, and TRENoP Strategic Research Area. The authors would like to thank Tomas Julner and Per-Olof Svensk at the Swedish Transport Administration, as well as Trafik Stockholm for their support and provision of the data to the iMobility Lab at KTH.

8 References


