Spatio-temporal Probabilistic Short-term Forecasting on Urban Networks

Cyril Furtlehner, Jean-Marc Lasgouttes, Alessandro Attanasi, Lorenzo Meschini, Marco Pezzulla

Spatio-temporal Probabilistic Short-term Forecasting on Urban Networks

Cyril Furtlehner, Jean-Marc Lasgouttes, Alessandro Attanasi, Lorenzo Meschini, Marco Pezzulla

RESEARCH REPORT
N° 9236
December 2018
Project-Teams Tau and Rits

ISSN 0249-6399   ISRN INRIA/RR--9236--FR+ENG

Spatio-temporal Probabilistic Short-term Forecasting on Urban Networks

Cyril Furtlehner[*], Jean-Marc Lasgouttes[†], Alessandro Attanasi[‡], Lorenzo Meschini[‡], Marco Pezzulla[‡]

Project-Teams Tau and Rits

Research Report n° 9236 — version 2 — initial version December 2018 — revised version July 2019 — 30 pages

Abstract: The probabilistic forecasting method described in this study is designed to leverage spatial and temporal dependency of urban traffic networks in order to provide accurate predictions for a horizon of up to several hours. By design, it can deal with missing data both for training and running the model. It is able to forecast the state of the entire network in one pass with an execution time that scales linearly with the size of the network. The method consists in learning a sparse Gaussian copula of traffic variables, compatible with the Gaussian belief propagation algorithm. The model is trained automatically from an historical dataset through an iterative proportional scaling procedure that is well suited to compatibility constraints. It is tested on three different datasets of increasing sizes ranging from 250 to 2000 detectors corresponding to flow and/or speed and occupancy measurements. The results show a very good ability to predict flow variables and reasonably good performances on speed or occupancy variables. Some understanding of the observed performances is given by a careful analysis of the model, making it to some degree possible to disentangle modelling bias from the intrinsic noise of the traffic phenomena and its measurement process.

Key-words: Traffic forecasting, belief propagation, Gaussian copulaes, Markov random fields

[*] Inria Saclay - LRI, Tau project team, Bât 660 Université Paris Sud, Orsay Cedex 91405
[†] Inria Paris-Rocquencourt, Rits project team
[‡] PTV-SISTeMA, Rome Italy

RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

Prediction spatiale et temporelle probabiliste du traffic urbain

Résumé : La méthode probabiliste de prediction de trafic décrite dans cet article exploite la dépendance spatiale et temporelle du trafic sur un réseau urbain, permettant de fournir des estimations précises jusqu'à quelques heures en avance. Par consruction elle permet de traiter les données manquantes aussi bien pour l'apprentissage du modèle que durant son utilisation. Elle prédit l'état d'un réseau complet en une seule passe, pour un temps d'exécution qui varie linéairement avec la taille du système. La méthode consiste à apprendre une copule Gaussienne sur des variables de trafic, compatible avec l'algorithme de propagation de croyances. Le modèle est appris automatiquement à partir de données historiques, via une procédure dite "iterative proportional scaling" bien adaptée pour imposer cette contrainte de compatibilité. Des tests sont effectués sur 3 jeux de données différents, de taille allant de 250 à 2000 détecteurs, correspondants à des variables de flux, de vitesse et/ou de densité. Les résultats indiquent une très bonne aptitude du modèle à prédire les flux, ainsi qu'une performance raisonnablement bonne sur les vitesses ou les dentités. Une analyse détaillée des résultats et du modèle nous permet également de séparer dans une certaine mesure les biais de modélisation des fluctuations intrinsèques du phénomène de trafic et de sa mesure.

Mots-clés : Prediction du trafic, propagation de croyances, copules Gaussiennes, champs markoviens aléatoires

Highlights

- Delivering accurate prediction by leveraging spatial and temporal dependencies.
- Dealing with missing data by construction.
- Forecasting the whole network state with an execution time which is linear with the size of the network.

1 Introduction

The rapidly evolving sector of traffic management information systems has made the problem of traffic forecasting at the urban network level a central and key issue. Traffic management systems aim to monitor, allow real-time scenario evaluations, and find control actions that keep the traffic flow as fluid as possible. They generally operate on main axes or principal sub-networks of the transportation infrastructure, and typically rely on traffic flow models, which in principle are able to predict the onset and the propagation of congestion. To be accurate, these flow models must have a high level of detail, which requires calibrating an enormous amount of parameters. But the calibration of these parameters is so challenging and time-consuming that the effectiveness of flow models can be greatly reduced. In this respect, data driven models can leverage the calibration effort and provide real-time input to the flow models in order to improve their results.

Moreover, the massive increase in available traffic data over the last decade has triggered a surge of interest in traffic forecasting based on data-driven models. These can be divided into two main categories.

- *parametric models*: vector auto-regressive models (VAR), ARIMA, STARIMA, probabilistic models, Bayesian models, MRF-based models;
- *non-parametric models*: k-NN, random forest, Gaussian process, support vector regression, neural networks.

Parametric models, based on ordinary statistical considerations, are more traditional. They are sometimes preferred to their non-parametric counterparts owing to their interpretability. Machine learning can potentially offer a very large variety of non-parametric models with a wide range of complexity and potential efficiency. General references on these various approaches can be found in [1]. Many methods are targeted toward independent segment modeling. Although methods attempting to leverage spatial dependencies are far fewer, they have been recently growing considerably in number [2]. If we focus more specifically on forecasting models which attempt to address the problem at the network scale, the requirements we find for such models to be deployed in online applications are the following

- *accuracy*: predictions should be significantly better than a simple persistent predictor combined with an historical time-of-day-dependent average, for instance, used when data are incomplete.
- *missing data*: we cannot expect to have a complete information of the network state at any time, which means that both the training and the running of the model have to be able to take missing data into account.
- *scaling*: the model should scale up to high systems size, i.e. networks of the size of an urban area, where the number of road segments involved can be around one or two hundred thousand. Actually, if we think in terms of detectors, this requirement might be lower.

Currently, the number of effective detectors covering a given urban area is smaller by one or two orders of magnitude than the number of road segments.

Traditional methods based on autoregressive models [3] have recently been adapted to this context, e.g. for treating floating car data (FCD) on a small scale (120-point location in central Rome) [4]. While yielding a good level of interpretability, such methods do not seem to scale up well to large network sizes. In order to capture local spatial features of traffic patterns, several studies (see e.g. [5]) have proposed hybrid machine learning methods involving neural networks and L_1 regularization of the weight matrix connecting the input to the hidden layer. They have, however, remained limited to a scale in the order of a few hundred detectors. More recently, deep learning approaches have been proposed: in [6], a stacked autoencoder is trained layer-wise on highway data at a coarse grain level, by considering the aggregation of traffic flow along each freeway direction. In order to address forecasting at a more detailed level, graph convolutional neural networks – a generalization of convolutional neural networks to graph structured data – have been proposed [7, 8] with various specifications and combinations with other RNN architectures like LSTM, in order to encode the temporal dynamics of spatial features extracted by the GCNN. Most of them show convincing improvements in performance compared to traditional methods, though they are often demonstrated on small-scale problems, again involving a few hundred variables, presumably due to the computationally heavy training procedure [9].

Another limit of such methods, aside from the computational resources that are needed for training and prediction and the seemingly limited scale of application, is the assumption that the data are complete. Missing values have to be imputed beforehand in one way or another in order to train the model and to use it [10, 11].

This paper proposes a different direction that copes with missing data, based on Markov Random Field (MRF) modeling. In this approach, a graphical model is used to identify conditional independence between segments at different locations and different time steps through a graph of interactions, which is generally assumed to be pairwise. Owing to its flexibility, this approach is particularly suitable for data imputation, even when observability of each segment is not known in advance. One difficulty is to train the model offline from historical data; another one is to be able to run it in real time, i.e. to perform the probabilistic inference of all the missing variables from the observed ones. For instance, it is shown in [12] that a simple Gaussian multivariate model with very few parameters can already perform better than simple interpolation methods at large scale with a cost that is linear w.r.t. system size, thanks to mean-field techniques. More refined models, like that proposed in [13] based on a mixture of sparse Gaussian random variables, can be trained efficiently using Expectation Maximization combined with an L_1 regularization to impose sparsity. They provide a more precise interpolation of missing values but with a cubic computational cost due to matrix inversion, which limits their use to medium-scale graphs. In a previous study, we investigated [14] the possibility of building an MRF which could encode both spatial and temporal dependencies and come with a linear computational time when running the probabilistic inference task. We considered two types of models involving either latent binary variables [15] or Gaussian copula models, both coming with an associated learning algorithm to generate compliant models, respectively with Generalized Belief Propagation for binary variables [16] and Gaussian Belief Propagation for real-valued variables [17].

Our purpose here is twofold: first to pull together some of the techniques developed in these previous studies by focusing more specifically on the Gaussian copula as a forecasting approach in a setting with missing data; second, to perform comprehensive experimental tests on real traffic datasets, in order to illustrate the effectiveness of this method in various conditions.

Our approach is designed to have the following desirable features which, to our knowledge, are not currently provided by any other method:

- provide accurate predictions: by exploiting spatial and temporal dependency, our method can deliver very accurate flow predictions, routinely with more than 85% of GEH < 5 up to one hour in advance (see Section 3.1 for a definition of GEH), and to some extent speed predictions of good quality as well;
- deal with up to 80% of missing data, after which the prediction accuracy decreases rapidly;
- deal with online constraints for large network sizes, owing to the linear scaling of belief propagation.
- deal with inhomogeneous data, e.g. speed, flow, travel time etc, all data being normalized through the definition of a traffic index;
- incorporate in an economical way, through the aforementioned traffic index, both time of day and seasonal dependencies within a single model;
- provide confidence intervals along with the predicted value.

Up to a point, these features meet the requirements listed earlier for a network-wide forecasting method. As regards the scaling to large system sizes, we are currently hampered by the training method, which is run offline. While belief propagation could realistically run online on models composed of many time layers of 10^5 variables on a standard CPU server, the training method has a cubic scaling with the size of the system. This means that we can train models containing up to the order of 10^4 variables in a reasonable time. Since we need at least two or three time layers to reach a good forecasting accuracy, this limits us with systems containing up to the order of 5000 detectors. Beyond that, some simplifying heuristics could be possibly be proposed, but they are beyond the scope of the present paper.

The paper is organized as follows: Sections 2.1–2.3 give the necessary materials, respectively the Gaussian belief propagation algorithm used for inference, the $\star$-IPS algorithm used to construct the model and the copula encoding. Then the workflow of the method is described in Section 2.4. The results of experiments performed on three different traffic datasets with various features are then reported and analyzed in Section 3. Comparison will be made in particular with results from single detector times series [18] forecast models to measure the advantage gained from spatial dependencies. Section 4 presents a statistical analysis of the errors of the model in which we try to separate intrinsic uncertainty of the traffic phenomena from systematic errors based on modeling issues. Section 5 concludes the paper and outlines future work.

2 Material and methods

2.1 Gaussian Belief propagation

We consider a set of discrete random variables $\mathbf{x} = \{x_i, i \in \mathcal{V}\}$ associated to a set of nodes $\mathcal{V}$ obeying a joint probability distribution of the form

$$\mathcal{P}(\mathbf{x}) = \prod_{ij\in\mathcal{E}} \psi_{ij}(x_i, x_j) \prod_{i\in\mathcal{V}} \phi_i(x_i), \tag{1}$$

where $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of edges and ϕ_i and ψ_{ij} are functions associated respectively to a single variable x_i and to an unordered pair of variables $ij \in \mathcal{E}$. The ψ_{ij} are called the "factors" while the ϕ_i are there by convenience and could be reabsorbed in the definition of the factors. $\mathcal{E}$ together with $\mathcal{V}$ define the graph $\mathcal{G}$, which will be assumed to be connected, and ∂i denotes the set of neighbors of node i in $\mathcal{G}$. Assuming that the graph is a tree, computing the set of marginal distributions, called the belief $b_i(x) = \mathcal{P}(x_i = x)$ associated to each variable i can be done efficiently. The Belief Propagation algorithm (BP) [19] does this for all variables in one single procedure, by taking into account that the computation of each of these marginals involves intermediate quantities called the messages $m_{ij\to j}(x_j)$ [resp. $n_{i\to ij}(x_i)$] "sent" by edge ij to variable node i [resp. variable node i to edge ij], and which are also necessary to compute other marginals. The idea of BP is to compute all these messages simultaneously, using the relation between them as a fixed point equation. Iterating the following message updates

$$
\begin{aligned}
m_{ij\to j}(x_j) &\leftarrow \sum_{x_i} n_{i\to ij}(x_i)\psi_{ij}(x_i, x_j),\\
n_{i\to ij}(x_i) &\leftarrow \phi_i(x_i) \prod_{k\in\partial i\backslash j} m_{ik\to i}(x_i),
\end{aligned}
$$

yields, when a fixed point is reached, the following result for the beliefs:

$$
\begin{aligned}
b_i(x_i) &= \frac{1}{Z_i}\phi_i(x_i) \prod_{ij\ni i} m_{ij\to i}(x_i),\\
b_{ij}(x_{ij}) &= \frac{1}{Z_{ij}}\psi_{ij}(x_{ij}) \prod_{i\in ij} n_{i\to ij}(x_i).
\end{aligned}
$$

This turns out to be exact if the factor graph is a tree, but only approximate on multiply-connected factor graphs. For a multi-connected factor graph, the beliefs b_i and b_{ij} usually form a pseudo-marginal distribution.

In practice, these equations are only usable in two cases: when each x_i takes a finite number of values, and when $\mathbf{x}$ is a Gaussian vector. These are the only cases where the message update formulas can be parameterized easily. The case of binary values, which correspond to an Ising model, has been studied in [14, 15]. Our focus here is on Gaussian variables, for which the algorithm takes on a specific form, referred to as Gaussian Belief Propagation (GaBP) [20]. The factors are naturally parameterized as

$$
\begin{aligned}
\psi_{ij}(x_i, x_j) &= \exp\bigl(-A_{ij}x_ix_j\bigr),\\
\phi_i(x_i) &= \exp\bigl(-\frac{1}{2}A_{ii}x_i^2 + h_ix_i\bigr).
\end{aligned}
$$

Pairwise factor messages can be seen as being sent directly from one variable node i to another one j in Gaussian form:

$$
m_{ij\to j}(x_j) = \exp\Bigl(-\frac{(x_j - \mu_{i\to j})^2}{2\sigma_{i\to j}}\Bigr).
$$

This expression is also stable w.r.t. the message update rules. Information is then propagated via the 2-component real vector $(\mu_{i\to j}, \sigma_{i\to j})$ representing bias and variance with the following update rules:

$$
\mu_{i\to j} \longleftarrow \frac{1}{A_{ij}}\Bigl[h_i + \sum_{k\in\partial i\backslash j} \frac{\mu_{k\to i}}{\sigma_{k\to i}}\Bigr], \tag{2}
$$

$$
\sigma_{i\to j} \longleftarrow -\frac{1}{A_{ij}^2}\Bigl[A_{ii} + \sum_{k\in\partial i\backslash j} \sigma_{k\to i}^{-1}\Bigr]. \tag{3}
$$

At convergence, the beliefs take the form:

$$b_i(x) = \sqrt{\frac{\sigma_i}{2\pi}} \exp\Big(-\frac{(x-\mu_i)^2}{2\sigma_i}\Big),$$

with

$$\mu_i = \sigma_i\big(h_i + \sum_{j\in\partial i} \frac{\mu_{j\to i}}{\sigma_{j\to i}}\big), \tag{4}$$

$$\sigma_i^{-1} = A_{ii} + \sum_{j\in\partial i} \sigma_{j\to i}^{-1}, \tag{5}$$

and the estimated covariance between x_i and x_j is written

$$\sigma_{ij} = \frac{1}{A_{ij}(1 - A_{ij}^2 \sigma_{i\to j}\sigma_{j\to i})}.$$

There is at most one fixed point, even on a loopy graph. It is not necessarily stable but, if convergence occurs, the single variable beliefs μ_i provide the exact marginals [21], with the computational time $O(N\log(N))$ roughly linear with the system size.

2.2 ⋆-IPS algorithm for model selection

Since GaBP may often encounter convergence issues, especially with non-sparse structures, it can be of practical interest to construct off-line a GMRF that is compatible with GaBP. By combining various methods proposed in the context of sparse inverse covariance matrix estimation [22, 23, 24], one way to do that has been developed in [17] in the form of the ⋆-IPS algorithm[1]. The starting point is the likelihood maximization

$$\mathcal{L}(A) = \log\det(A) - \mathrm{Tr}(A\hat{C})$$

of the precision matrix A, given some covariance empirical matrix $\hat{C}$. Without any constraint on A, the maximum likelihood estimate is trivially $A = \hat{C}^{-1}$. In our context, where compatibility with GaBP has to be imposed, one feature like sparsity can be desirable, albeit without much guarantee. Specific topological properties like the presence of short loops, are likely to damage the GaBP compatibility, even on a sparse graph. Additional spectral properties, e.g. walk-summability [25], can guarantee the compatibility with GaBP-based inference. ⋆-IPS incorporates these explicitly, by combining an approach based on the iterative proportional scaling (IPS) procedure [26], with block updating techniques used in [22, 23]. The rationale of ⋆-IPS is to construct the graphical model $P(\mathbf{x})$ link by link, by ensuring at each step that the constraints are satisfied. If P is the current approximate model after some steps, it turns out that

$$P'(\mathbf{x}) = P(\mathbf{x}) \times \frac{\hat{p}_{ij}(x_i, x_j)}{p_{ij}(x_i, x_j)}, \tag{6}$$

is the optimal deformation of link (i, j), where $\hat{p}_{ij}$ is the empirical pairwise marginal, while p_{ij} is the pairwise marginal of P. The corresponding log-likelihood gain is given by $\Delta\mathcal{L} = D_{KL}(\hat{p}_{ij}\|p_{ij})$. Sorting all the potential new links w.r.t. this quantity yields the optimal 1-link correction to be made. In terms of precision matrix modification, this corresponds to a 2×2 update which

[1] Available at `https://gitlab.inria.fr/bptraffic/star-ips`.

involves the current covariance matrix $C = A^{-1}$ of the approximate model. This covariance matrix has to be maintained after each update, which can be done efficiently thanks to the Sherman–Morrison–Woodbury formula for low rank modifications of the precision matrix A. Direct inspection of the modified precision matrix, shows that positive definiteness of the matrix is preserved by such updates.

Each modification is accepted only if it satisfies the constraints. The best candidate link can thus be discarded if the constraints are violated by this addition. Two families of constraints are considered:

- Topological constraints avoid the presence of small loops, with possibly a distinction between frustrated/non-frustrated loops, i.e. loops with a negative product of partial covariances $(-A_{ij})$.

- Spectral constraints like walk-summability [resp. weak walk-summability] involve definite positiveness of matrix $\mathrm{diag}(A) - |A - \mathrm{diag}(A)|$ [resp. matrix $2\,\mathrm{diag}(A) - A$], where $\mathrm{diag}(A)$ is the matrix containing only the diagonal elements of A.

When a new link is added, existing links can become detuned by a slight amount. In order to optimize existing links, (6) can be used. Other local updates are also available like block updates, via a single row-column update of the precision matrix, as originally proposed in [22] and refined in [23]. In practice, $\star$-IPS alternates many link additions, corresponding to a significant increase in mean connectivity, with block coordinate descent procedures. Overall, sparse precision matrices of good likelihood are generated in $O(N^3)$ steps, with the advantage of having the complete optimization path available, by means of many graphical models of intermediate connectivity. Note finally that we have discarded the standard way of generating sparse precision matrices based on the Lasso penalty [23, 27]. There are two reasons for that: firstly the L_1 norm penalty suffers from a modeling bias, due to excessive penalization of truly large magnitudes entries of A; secondly it is not flexible enough for the kind of constraints we are interested in, in order to produce graphical models compatible with GaBP of high likelihood [17].

2.3 Gaussian copula model of traffic indexes

A key feature of our method is a mapping of raw data, that can correspond to flow or speed for instance, to a standard normal variable, a kind of properly normalized traffic index on which the prediction is performed. The joint probability measure of these traffic indexes is approximated through a Gaussian copula. We define this index in the following way. Let t be a discretized time, measured in time steps δ_t of fixed length. N_t represents the number of such time steps contained in a single day. For a given t, the time of day $\tau \in \{0, \dots N_t - 1\}$ is given by $\tau = t$ modulo N_t. For instance we have $N_t = 96$ if we consider $\delta_t = 15\,\mathrm{min}$ time steps. At each time step, we assume the system to be represented by N_v variables X_i^t, $i \in \{1, \dots N_v\}$ corresponding to traffic detectors. Now for each variable X_i and each time of day τ we build from the historical data a running average $\bar{X}_i^\tau$ and a variance V_i^τ. If the dataset is clustered into a certain number of weekly or seasonal patterns, then these quantities will be estimated for each cluster labeled by some extra index ℓ. Then, for each variable index i, time t (and associated time of day τ) and cluster label ℓ, we map $X_i^{t,\ell}$ to the following variable

$$U_i^{t,\ell} \stackrel{\text{def}}{=} \frac{X_i^{t,\ell} - \bar{X}_i^{\tau,\ell}}{\sqrt{V_i^{\tau,\ell}}}, \tag{7}$$

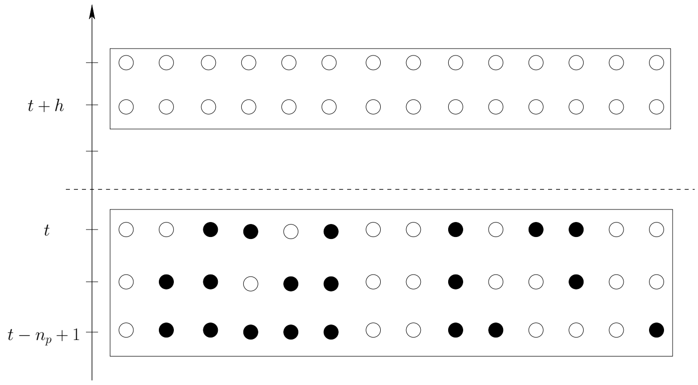

Figure 1: Time layers setting of one basic space time configuration. Each circle represents one single variable at a given time, filled circles correspond to observations.

which represents a centered and normalized variable for given τ and ℓ. From this transformed historical data, we build for each index i a single cumulative distribution function

$$F_i(x) = P(U_i < x),$$

which for a given realization $U_i^{t,\ell} = x$ represents the traffic index. The purpose of this index is to encapsulate all average time-dependent trends, week-day and seasonal dependencies, while the Gaussian copula will represent the fluctuations around these trends. In order to build a sparse Gaussian copula of all the indexes we first transform each variable $U_i^{t,\ell}$ into a normal variable via the following mapping:

$$Y_i^{t,\ell} = F_{\mathcal{N}(0,1)}^{-1} \circ F_i(U_i^{t,\ell}), \tag{8}$$

where $F_{\mathcal{N}(0,1)}$ is the cumulative distribution of a standard normal variable.

The copula model corresponding to $n+h+1$ time layers (n past layers, 1 present and h future layers) is then obtained by considering the vector

$$Z^t = (Y_i^{t+k,\ell}, i = 1 \dots N_v, k = -n, \dots, 0, \dots h)$$

and constructing its associate sparse multivariate approximate model with $\star$-IPS. This model will be used to generate predictions $\hat{Y}_i$, which in turn can be converted into predictions $\hat{X}_i$ of the original variables by inverting (8,7).

2.4 Workflow of the method

The complete workflow of our method actually comprises two separate tasks: creating the model, and using it to produce detailed forecasts. We give the details of our implementation below.

2.4.1 Model building (MB, offline)

This task is only supposed to be run daily, weekly or even monthly, in order to refresh the model with recent data. It typically takes several hours to complete.

(MB0) architecture Choose an architecture for the model which is defined by

- n_p: number of past and present layers of variables for which observations may be available. If t is the present time, this covers times $t - n_p + 1, \ldots, t$.
- n_f: number of future layers we want to predict, $n_f \geq 1$.
- h: horizon of prediction of the first future layer.

From this, we construct by concatenation of $(n_p + n_f)N_v$ variables the dynamical configurations (see Figure 1):

$$\{X_i^{t-n_p+1}, \ldots, X_i^t, X_i^{t+h}, \ldots, X_i^{t+h+n_f-1}, i = 1, \ldots, N_v\}.$$

This subtask is only supposed to be done once, when setting up the original model. There is no need to tweak the values later when updating historical data.

(MB1) statistics Compute the required statistics of the model from the training set: the mean and variance $\bar{X}_i^{\tau,\ell}$ and $V_i^{\tau,\ell}$ for each variable i, time of day τ and label ℓ for the day (when a clustering is available or has been pre-processed). Then, for each i compute the cdf F_i of $U_i^{t,\ell}$ aggregated over all t and approximate it by an invertible monotonous linear piecewise function.

(MB2) covariance matrix Generate using (8) a training set of vectors

$$\{Y_i^{t-n_p+1}, \ldots, Y_i^t, Y_i^{t+h}, \ldots, Y_i^{t+h+n_f-1}, i = 1, \ldots N_v\}$$

and compute the corresponding $(n_p + n_f)N_v \times (n_p + n_f)N_v$ covariance matrix. Because the data is incomplete, some regularization can be required if some modes are associated to large negative eigenvalues. In such a case the eigenvalue is replaced by its absolute value.

(MB3) MRF model The model is built using the $\star$-IPS algorithm [17], which takes the regularized covariance matrix as input and generates an almost continuous set of models with increasing mean connectivity $\bar{d}$. The variant that we use (FLOOP-5-IPS) avoids frustrated short loops of sizes up to 5 (see Section 2.2). The mean connectivity $\bar{d} = d^\star$ of the model is selected somewhere in the region where the derivative of the log likelihood as a function of the mean connectivity $\bar{d}$ starts to flatten (see Figure 2). Its value is 4 for Turin and 6 for the two other datasets.

2.4.2 Inputation and prediction (IP, online)

At this point, the model is ready to use for prediction tasks. It is defined by a multivariate Gaussian distribution $P(Y)$ in which the variables corresponding to observations are clamped, while the other (future and non-observed past variables) are inferred using GaBP, as described in Section 2.1

(IP1) Initialization Initialize the GMRF with observations, using either strong of soft beliefs. In the strong case, the variable, say x_i is fixed to its observed value $\hat{x}_i$ and detached from the factor graph. Neighboring variables $\{x_j, j \in \partial i\}$ see their local field modified as

$$h_j \longrightarrow h_j - A_{ij}\hat{x}_i.$$

In the soft case, the observable $\hat{x}_i$ comes with an uncertainty either given by some variance $\hat{\sigma}_i$ or else it is unknown. In the experiments presented later only the strong beliefs will be used.

(IP2) Inference Run GaBP according to equations (2)–(3) until convergence is reached. The predictions are then obtained from (4) for each variable to be predicted, along with an estimation of the uncertainty provided by the variance (5). Note that, for observed variables inserted with soft constraints, the value of the variance (5) provided by the fixed point may give some indication of whether the input observation is trustworthy or not.

2.4.3 Practical considerations

There are a couple of practical issues to be aware of when using this workflow.

Model tuning There are basically two hyper-parameters of the model, namely the number of past layers n_p and the mean connectivity d^*, to be tuned in one way or another. Indeed h is imposed by the task and n_f is an optional choice, with its default value $n_f = 1$. These hyper-parameters are set manually in our experiments by performing a few tests on the training set. The sensitivity of the performance to these parameters around the chosen values appears to be actually very small, which greatly simplifies their tuning.

Complexity and affordable systems sizes Since $\star$-IPS has a cubic complexity, the model building task (MB) has a running time that scales like $\big((n_p + n_f)N_v\big)^3$. In practice, the total number of variables $(n_p + n_f)N_v$ should not exceed the order of 10^4 so as to be able to train the model in a reasonable time, which means that if we consider $n_p = 2$ and $n_f = 1$ we can deal with systems of roughly 3000 detectors.

Concerning the inputation and prediction task (IP), in our setting GaBP can converge on networks of size 10^5 within a few tenth of a second on an ordinary CPU. Typically, the online constraint for traffic information is to generate a forecast every five minutes. Thus, it is in principle possible to run a network-wide forecast of some 10^5 of detectors.

3 Experiments

In order to check the robustness of the method, we have tested our system on a certain number of datasets with different characteristics, corresponding either to urban or highway traffic.

3.1 Error metrics

Since no single metric is able to account for the errors made by a given method, we consider a set of metrics commonly used in traffic analysis. General purpose metrics are the root mean square error (RMSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE). Letting X be the variable to predict and $\hat{X}$ its predictor, the metrics are defined as follows.

$$\mathrm{RMSE} \stackrel{\mathrm{def}}{=} \sqrt{\mathbb{E}[(\hat{X} - X)^2]}$$

$$\mathrm{MAE} \stackrel{\mathrm{def}}{=} \mathbb{E}[|\hat{X} - X|]$$

$$\mathrm{MAPE} \stackrel{\mathrm{def}}{=} 100 \times \mathbb{E}\left[\frac{|\hat{X} - X|}{X}\right]$$

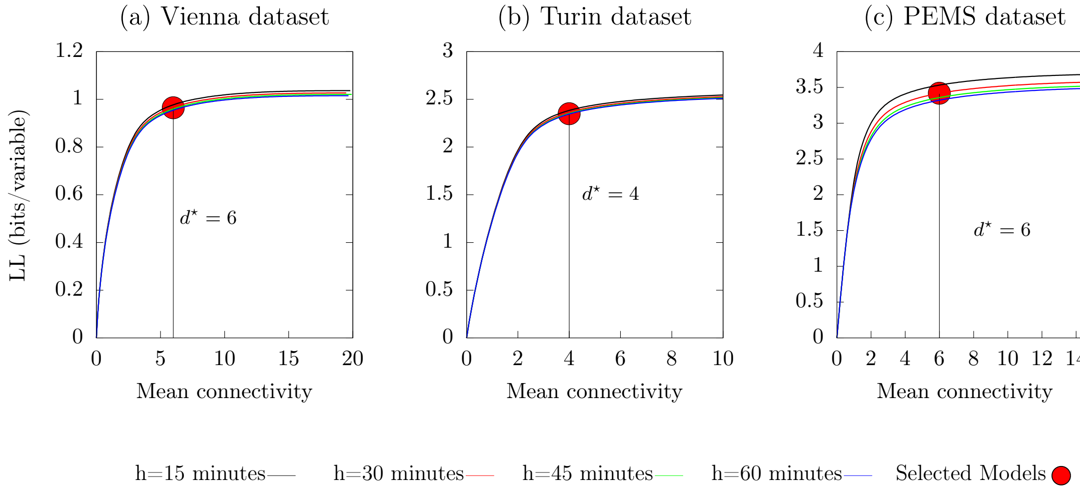

Figure 2: Log likelihood of the sparse copula models as a function of mean connectivity at various forecast horizons for the Vienna, Turin and PeMS datasets

Additionally, the GEH statistic can be specifically used for traffic flow predictions. When X is an hourly traffic flow, the GEH statistic is defined as

$$\mathrm{GEH} \stackrel{\mathrm{def}}{=} \sqrt{\frac{2(X-\hat{X})^2}{X+\hat{X}}}.$$

This number has been empirically designed to yield comparable numbers for a broad range of arterial road capacities. Typically, the prediction is considered good if GEH is below 5, and the statistic of interest is thus the percentage of GEH < 5.

The expectations above are taken uniformly over all detectors and all time steps for which there is an observation to compare against the prediction. Note that MAPE is not defined when $X = 0$, which can happen in particular for flow measurements. To regularize this, we simply threshold the denominator in MAPE to $X_{\min} = 10$ in practice.

3.2 Baseline predictors

In order to provide some reference points of comparison for our results we use some simple predictors and when possible, i.e. for the Vienna dataset, other results from the literature. Our simple baseline predictors are the following:

- Mean(t) is the historical average $\bar{X}_i^{\tau,\ell}$ for detector i at time of day τ and label ℓ, as defined in Section 2.3.
- t_0 is the persistent predictor, sometimes called the random walk predictor. It is equal to the last observed value within the past time window of the detector. If no observation is found within the past time window, Mean(t) is used instead.
- k-NN is a k nearest neighbors predictor. Given the past and present time observations, which can comprise many time layers of incomplete data, we look in the training set for the

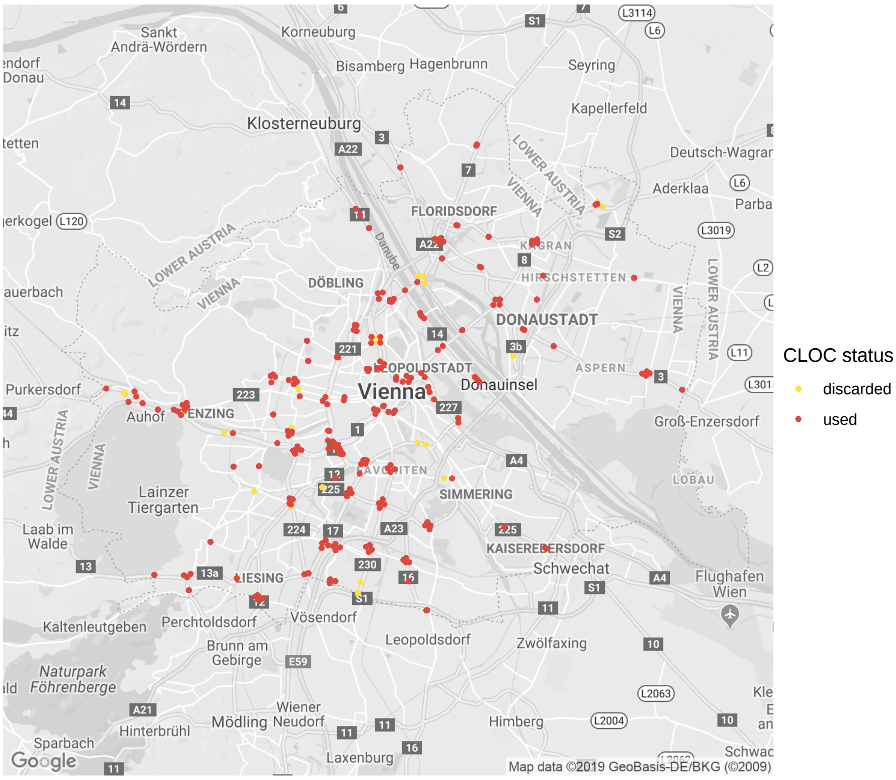

Figure 3: Location of the CLOCs in Vienna. Only the red ones have enough data to be usable.

k closest states to the current ones by means of an $\mathbb{L}_2$ discrepancy. The hyper-parameter k needs to be optimized.

For the Vienna dataset, other methods based respectively on Bayesian networks (BN), neural networks (NN) and clustering of day profiles have been previously tested [18] on a subset of the count locations that we are considering. The interesting point is that these methods treat each detector independently, so the comparison gives an indication of the added value of taking spatial dependencies into account.

3.3 Vienna dataset

Our first dataset concerns urban traffic in the Vienna (Austria) urban area, as depicted in Figure 3. It consists in flow data collected over 4 years (2011–2014) from 292 count locations (CLOC) measuring vehicles every 15 minutes. Since these detectors are quite often down, we arbitrarily select those which are up during at least 10% of the time. This leads to retaining 266 CLOCs out of 292. With this setting, at any given time, we have typically around 65% of the detectors which are up. This means that, in addition to forecasting a certain number of layers of variables ahead in time, our model will perform an imputation of the 35% of missing observations. The first 3 years (2011–2013) are used as a training set, while 2014 is used for testing. As a preprocessing step, we have performed a clustering analysis of the daily traffic conditions. This analysis yields 10 meaningful clusters, presented in Table 1. This clustering will be used in two ways: first it will serve to build a baseline predictor $\bar{X}_i^{(\tau,\ell)}$ for all CLOCs i and where τ runs from 1 to $N_t = 96$

	J	F	M	A	M	J	J	A	S	O	N	D
Sun.	9	9	9	5	5	5	5	5	5	9	9	9
Mon.	1	1	1	4	4	4	7	7	8	8	10	10
Tue.	1	1	1	4	4	4	7	7	8	8	10	10
Wed.	1	1	1	4	4	4	7	7	8	8	10	10
Thu.	1	1	1	4	4	4	7	7	8	8	10	10
Fri.	3	3	3	3	3	3	3	3	3	3	3	3
Sat.	2	2	2	2	2	2	6	6	6	2	2	2

Table 1: Description of the 10 clusters according to month and day of the week. Public holidays are classified as Sundays; the ranges for Saturday and Sunday are actually by season, so that Cluster 5, for example, starts on March 21$^{\text{st}}$.

time-steps and ℓ the cluster index running from 1 to 10; secondly our model, as explained before, will be built on the residues $U_i^{(t,\ell)}$ defined in (7) from this baseline or some related ones, obtained in the same way after merging some of the clusters, in order to have better statistics per clusters.

For this setting, several architectures lead to similar performances, as long as at least 3 layers in the past are taken into account.

Specializing the model to one single horizon instead of directly performing the forecast for many features ahead does not seem to help much. The results presented in Figure 4 were obtained with a model with 4 past layers and one single specialized future layer. The results summarized in Table 2 were obtained with a multi-step ahead model having 4 past and 4 future layers. Specialized or multi-step ahead models yield identical performance within error bars. The multi-step ahead model appears to be more advantageous in practice as it can deliver all horizon predictions in one pass. The only drawback comes from the effort necessary to build it since it contains more variables.

As our model is able to perform both completion of missing data and forecasting at the same time within a single GABP fixed point convergence, we first study the dependency of the forecasting performance on the fraction of observed variables. To this end, we randomly occult from the roughly 65% of observations available some variables of the past and present layers in order to have a given density ρ of observed variables per layer as illustrated in Figure 1. The result is shown in Figure 4 (left). This plot shows that a lot of information is already gained by observing only a small fraction of variables, say 5–10% and after that the forecasting and inputation errors continue to decrease more slowly but steadily, while for the k-NN model the prediction error saturates after $\rho = 0.2$. The forecasting performance of the model as a function of the horizon is shown in Figure 4 (right). With respect to the Mean(t) predictor, we see that the error is reduced by 33% on the short-term prediction. In addition, our model provides meaningful information up to 5 hours in advance w.r.t. this baseline, which means that it is able to identify additional traffic patterns not captured by clustering. Finally, a qualitative indication that the GaBP predictor is performing well is obtained by looking at the individual CLOCs' time series and associated predictions. As an example of the typical behavior of the model, a small sample of these time series is shown in Figure 5. We can see that GaBP follows the changes in traffic conditions very well without any delay and performs a kind of smoothing of the actual traffic flow signal.

Table 2 provides a point of comparison with other models based on single-time series forecasting on a reduced set of CLOCs. We can therefore evaluate, to some degree, the benefits of leveraging spatial dependencies. From these results, we see some gain at all horizons regarding all metrics of evaluation, rather marginal for $h = 15'$ but rapidly sizable when the horizon increases. By

	Flow (36 CLOCs)				Flow (266 CLOCs)			
	%GEH<5	RMSE	MAE	MAPE	%GEH<5	RMSE	MAE	MAPE
$h=15'$								
BN	-	33.86	23.06	16	-	-	-	-
NN	-	32.38	22.13	16	-	-	-	-
Cluster	-	35.51	23.84	16	-	-	-	-
Mean(t)	73.51	49.65	29.59	23.56	78.88	40.97	21.30	36.66
t_0	70.49	47.54	30.19	18.94	80.32	31.71	17.57	20.71
k-NN	81.33	36.99	23.07	17.87	83.46	31.89	17.03	27.27
GaBP	**87.19**	**31.89**	**19.45**	**15.31**	**89.11**	**25.97**	**13.62**	**18.85**
$h=30'$								
BN	-	38.98	25.81	17	-	-	-	-
NN	-	37.10	24.34	17	-	-	-	-
Cluster	-	39.65	25.88	18	-	-	-	-
Mean(t)	73.51	49.65	29.59	23.56	78.88	40.97	21.30	36.66
t_0	70.33	45.68	29.39	20.44	74.74	35.17	20.02	24.10
k-NN	80.50	37.90	23.60	18.31	82.77	32.60	17.43	28.39
GaBP	**85.68**	**34.07**	**20.55**	**16.19**	**87.82**	**27.74**	**14.41**	**20.22**
$h=60'$								
BN	-	47.20	30.92	20	-	-	-	-
NN	-	46.68	28.46	20	-	-	-	-
Cluster	-	46.68	29.81	20	-	-	-	-
Mean(t)	73.51	49.65	29.59	23.56	78.88	40.97	21.30	36.66
t_0	58.47	63.00	40.24	27.57	62.42	47.84	27.56	32.61
k-NN	78.90	39.32	24.57	19.16	81.44	33.32	17.98	29.31
GaBP	**83.70**	**36.58**	**22.01**	**17.27**	**85.90**	**29.70**	**15.51**	**21.66**

Table 2: Results on the Vienna dataset for different horizons of forecasting and comparison on a subset of CLOCs with single time-series forecasting based methods (BN and NN from [18]).

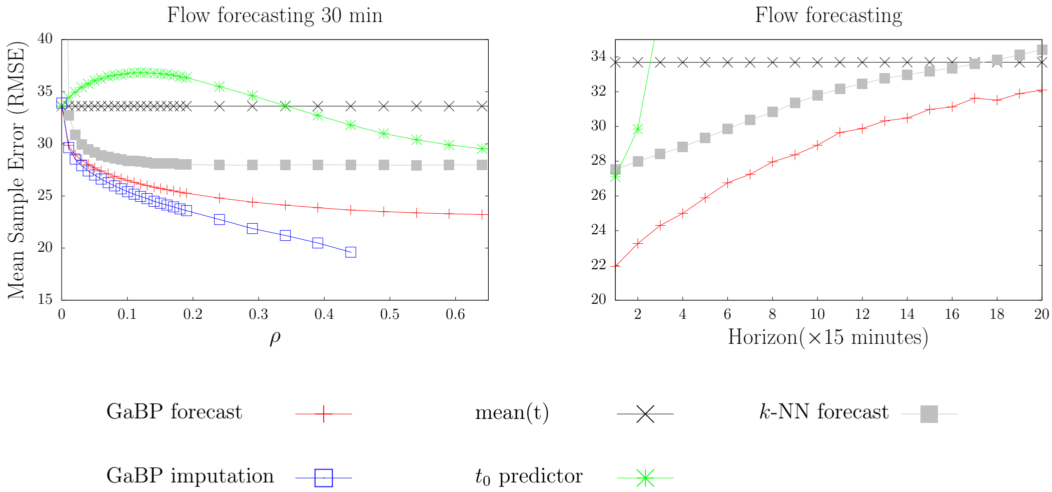

Figure 4: Vienna dataset: (left panel) average flow forecasting error for a given time lag of 30 minutes, as a function of the fraction ρ of observed variables in the past time layers, averaged over 5000 test samples. The inputation error for missing data is also shown (in blue). A point of comparison is given by the k-NN predictor, optimized for $k = 50$. (Right panel) Average flow forecasting error as a function of time lag at maximum possible observation rate $\rho \sim 0.65$ (average over the full test set).

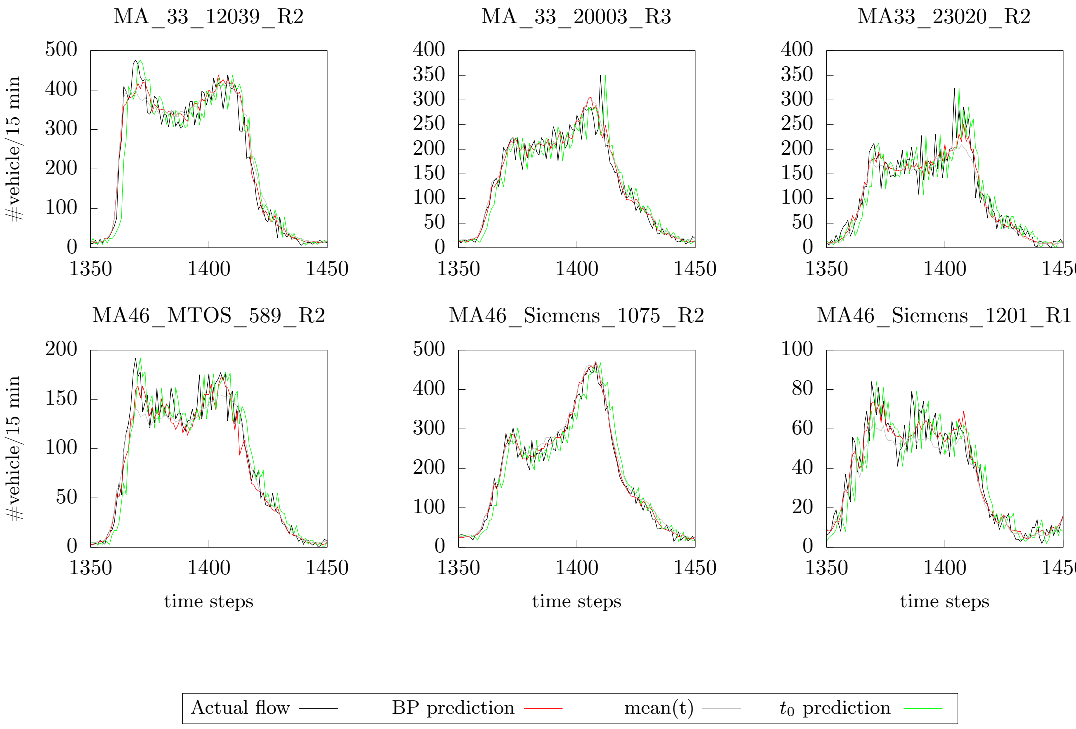

Figure 5: Vienna dataset: excerpt of flow time-series along with GaBP and t_0 prediction for a 30-minute horizon for 6 different CLOCs.

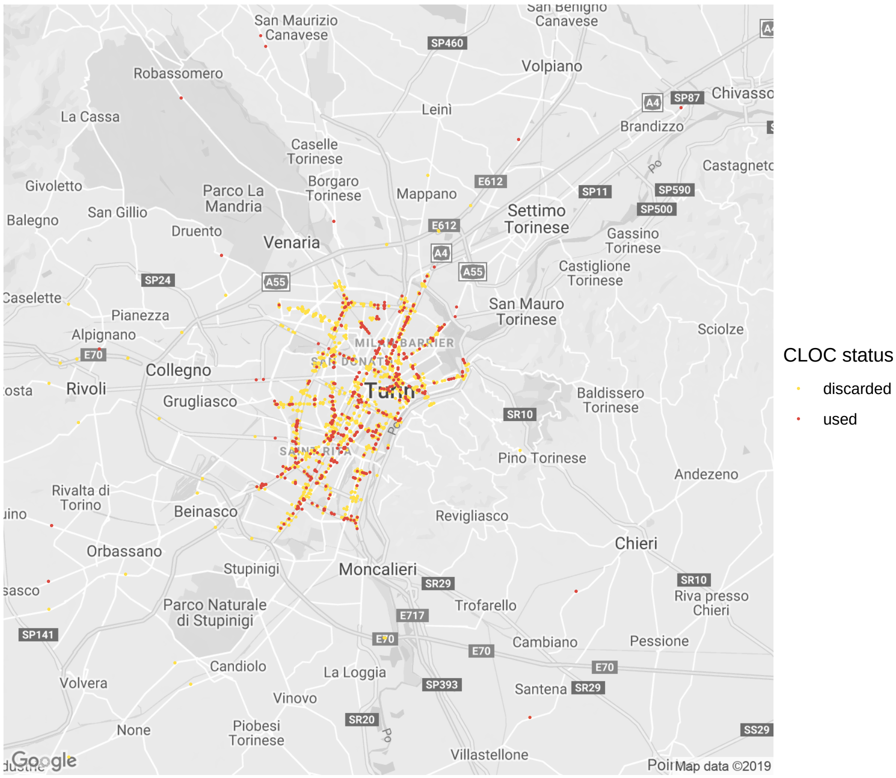

Figure 6: Location of the CLOCs in the center of Turin. Only the red ones have enough data to be usable. Note that there are 148 additional sensors outside of this area that are not displayed here for the sake of clarity.

looking more specifically at individual CLOC errors, those having less than 70% of GEH < 5 are either badly behaving sensors clearly sending corrupted data, or sensors for which a systematic bias is present in our Mean(t) baseline possibly reflecting a long lasting modification in the traffic conditions for the corresponding segment.

3.4 Turin dataset

The second dataset again concerns urban traffic, in the Turin (Italy) urban area. There are 1489 count locations, giving in principle speed measurements in addition to the flow. The map of these CLOCs in the center of Turin is shown in Figure 6. Two years of data (2015–2016) have been collected. As for the previous dataset, we select detectors based on an activity threshold of 10%. We end up with 685 detectors, of which 566 correspond to flow and 119 to speed measurements. With this setting, the ratio of available data is around 60% in the past layers. Year 2015 is used for training and year 2016 for testing. A straightforward clustering is done here with 4 clusters: one for Monday–Thursday and the other ones respectively for Fridays, Saturdays and Sundays. This leads to a slightly noisier Mean(t) baseline (around 45 in RMSE instead of 40 for Vienna). Figure 7 shows the effect of available data ratio and prediction horizon for a model with $(n_p, n_f) = (3, 1)$, i.e. three past layers and one future layer to predict, and therefore of size equal to 2740 variables. Table 3, in contrast, compares different methods to a multi-steps ahead

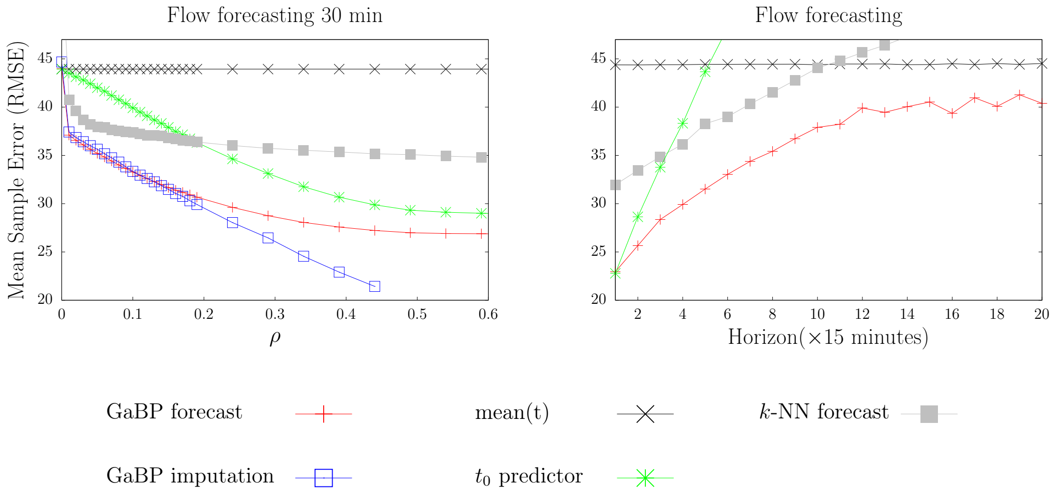

Figure 7: Turin dataset: (left panel) average flow forecasting error for a given time lag of 30 minutes, as a function of the fraction ρ of observed variables in the past time layers, averaged over 5000 test samples. The inputation error for missing data is also shown (in blue). A point of comparison is given by the k-NN predictor with $k = 50$. (Right panel) Average flow forecasting error as a function of time lag (right) at maximum possible observation rate $\rho \sim 0.6$, average over the full test set.

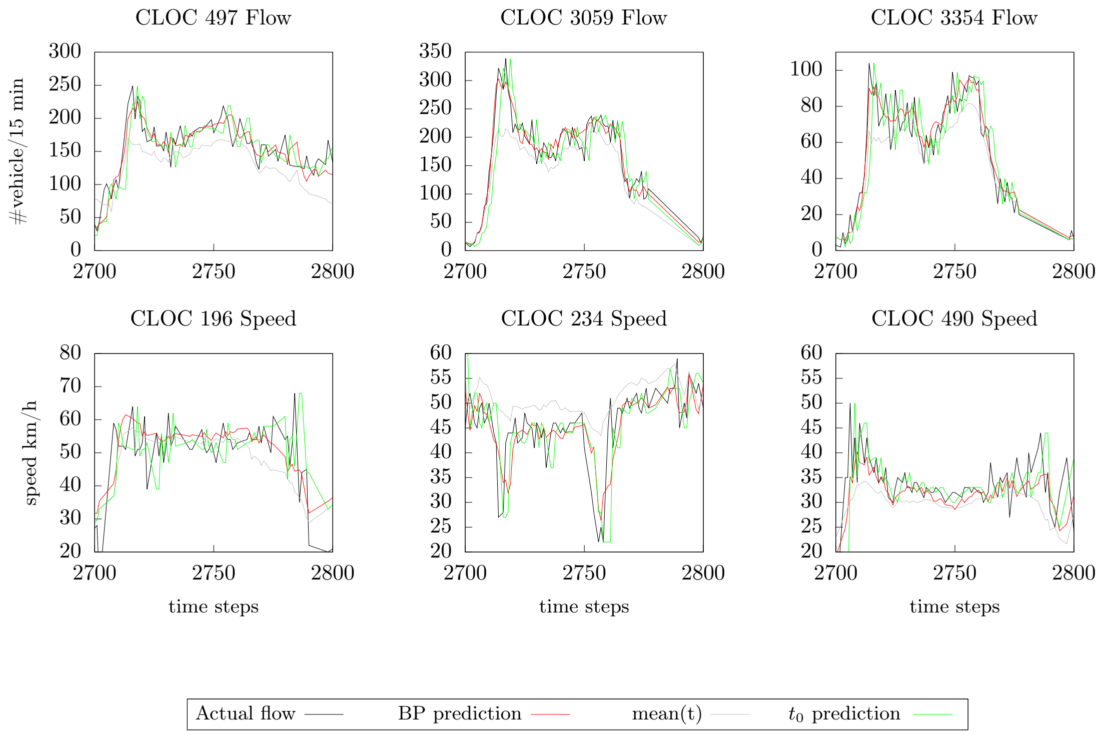

Figure 8: Turin dataset: excerpt of flow or speed time-series along with GaBP and t_0 prediction for a 30-minute horizon for 6 different CLOCs.

	Flow (566 CLOCs)				Speed (119 CLOCs)		
	%GEH < 5	RMSE	MAE	MAPE	RMSE	MAE	MAPE
$h = 15'$							
Mean(t)	75.46	45.09	21.65	31.02	8.40	5.21	13.73
t_0	81.41	27.58	15.43	20.32	9.24	5.00	11.74
k-NN	79.78	37.71	18.47	27.19	8.20	4.99	13.28
GaBP	**88.10**	**27.56**	**13.32**	**19.03**	**7.65**	**4.24**	**10.44**
$h = 30'$							
Mean(t)	75.46	45.09	21.65	31.02	8.40	5.21	13.73
t_0	73.67	35.02	19.22	24.50	9.68	5.35	12.70
k-NN	77.65	39.70	19.55	29.42	8.34	5.11	13.78
GaBP	**87.10**	**30.73**	**14.20**	**20.68**	**7.56**	**4.25**	**10.70**
$h = 60'$							
Mean(t)	75.46	45.09	21.65	31.02	8.40	5.21	13.73
t_0	62.00	47.92	26.67	32.22	10.36	5.88	14.12
k-NN	73.99	42.28	21.32	31.51	8.43	5.15	13.32
GaBP	**85.24**	**33.90**	**15.45**	**22.58**	**7.67**	**4.34**	**11.09**

Table 3: Results on the Turin dataset for different horizons of forecasting, for both flow and speed measurements.

GaBP with $(n_p, n_f) = (4, 4)$ with 5480 variables. In both cases, the selected models are sparser than for Vienna, with mean connectivity 4 (see Figure 2). Yet the behavior and performance are comparable, with e.g. more than 85% of GEH < 5 at a 1 hour horizon. All indicators for the flow are very similar to those of the Vienna dataset, with slightly higher RMSE and lower MAPE and MAE, because of an higher overall capacity bias of the segments in the case of Turin. As far as speed is concerned, we observe for instance at a 30-minute horizon an RMSE improvement of 10% w.r.t. the Mean(t) baseline and 20% w.r.t. the t_0 predictor. The results for flow are clearly more impressive than those for speed predictions. This hides important disparities between various days. In fact, the aggregated error for the speed is dominated by nighttime prediction errors, where the small number of speed measurements leads to a very noisy signal.

Figure 8 shows some excerpts of single detectors' prediction time series. As for the Vienna dataset, the model is able to anticipate correctly the changes in traffic flow even far from recurrent traffic conditions. Sudden drops in speed are not always anticipated, as shown in the last panel of this figure, for instance.

3.5 PeMS dataset

The data from the PeMS monitoring system (`http://pems.dot.ca.gov`) used here have been acquired on the freeways of district 4 of the San Francisco Bay area during the first 11 months of 2013. These data are provided without any missing values, thanks to an automatized interpolation procedure [28]. For each station as many values as freeway lines are reported and an aggregated value coming with an observation rate. This rate corresponds to the percentage of actual observations that make up the aggregate value. This aggregate value is the one we consider in our experiments. Since we don't want other interpolation procedures to interfere much with our own, we retained only data coming with a high observation rate ($> 80\%$), the other data being considered as missing (see Figure 9). On top of this filtering procedure, we select detectors that

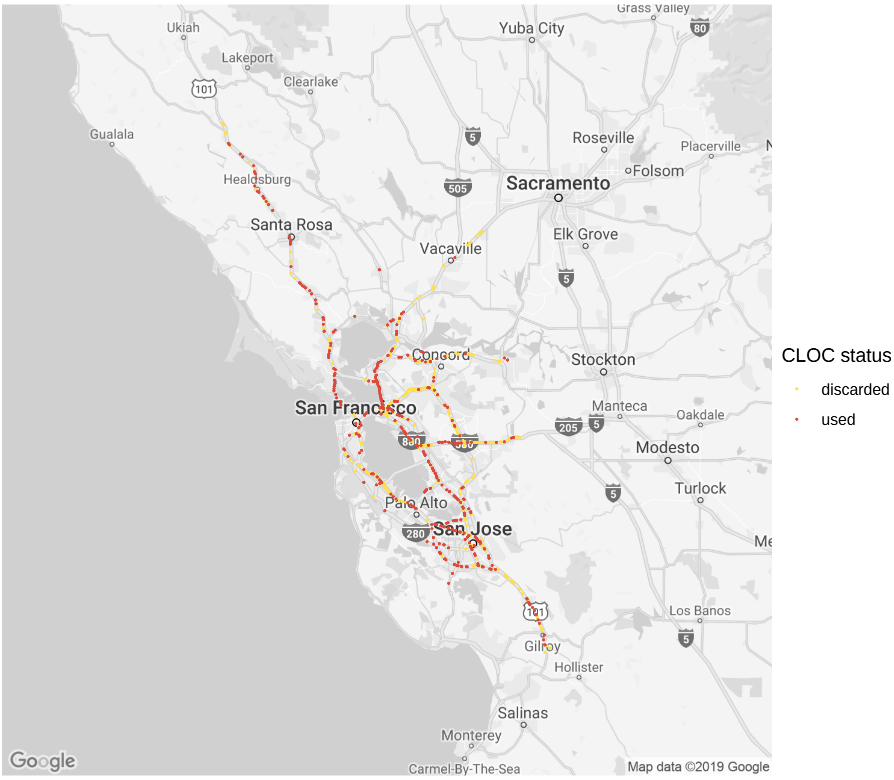

Figure 9: Location of the CLOCs in the PeMS dataset. Only the red ones have enough data to be usable.

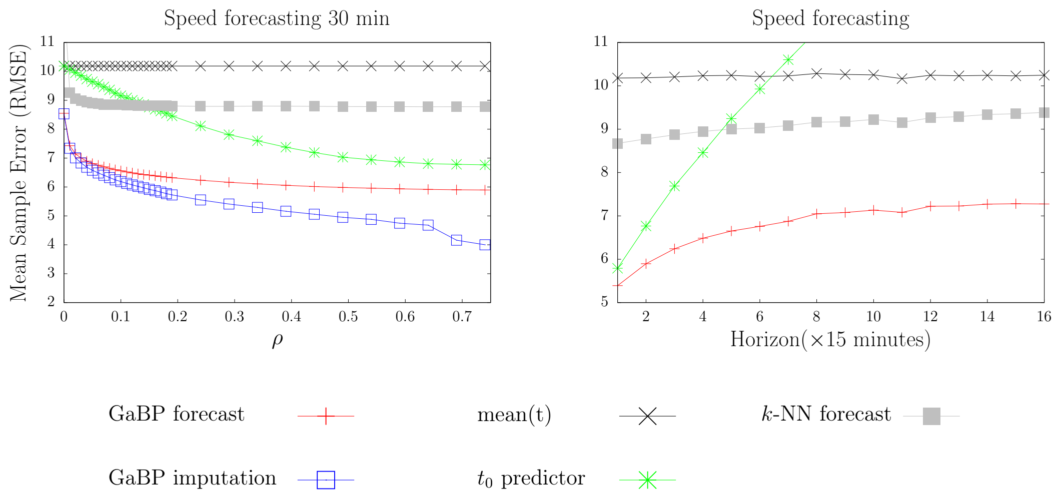

Figure 10: PeMS dataset: average speed forecasting error for a given time lag of 30 minutes, as a function of the fraction ρ of observed variables in the past time layers, averaged over 1000 test samples (left). Average speed forecasting error as a function of time lag (right) at maximum possible observation rate $\rho \sim 0.75$, (averaged over 1000 test samples). The inputation error for missing data is also shown (in blue). A point of comparison is given by the k-NN predictor ($k = 50$).

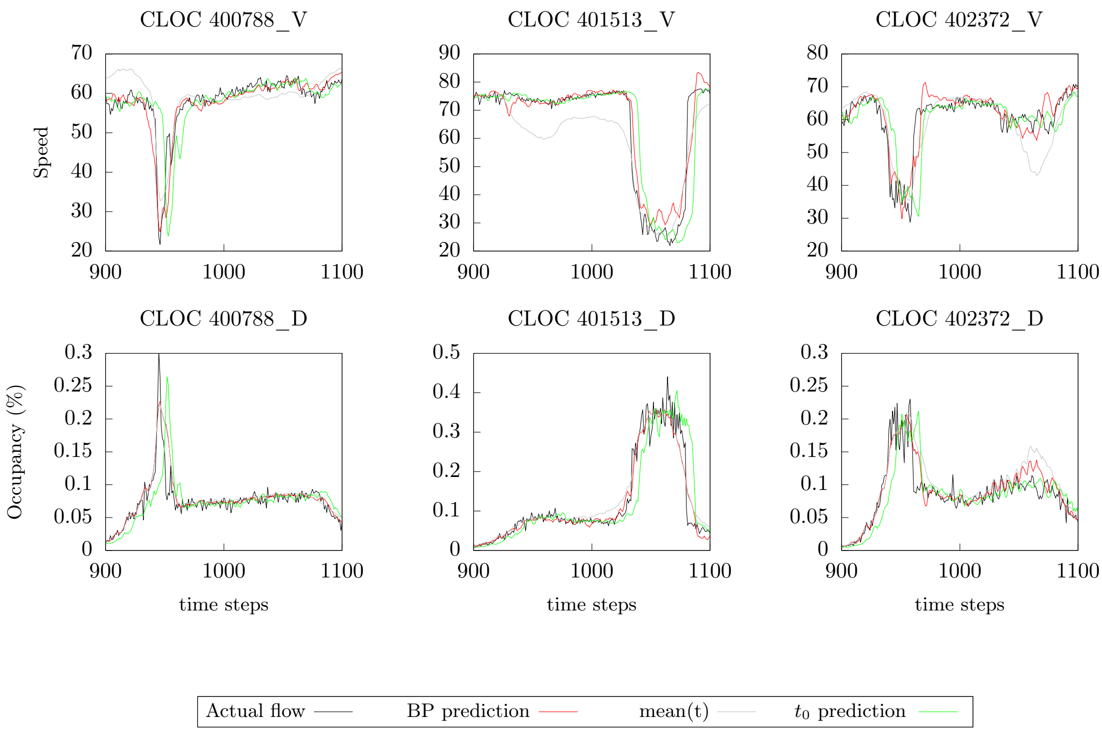

Figure 11: PeMS dataset: excerpt of speed time-series along with GaBP and t_0 prediction for a 30-minute horizon for 6 different speed detectors

	Flow (645 CLOCs)				Speed (645 CLOCs)			Occupancy (645 CLOCs)		
	%GEH < 5	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
$h = 15'$										
Mean(t)	58.01	123.5	71.56	22.08	6.17	3.51	7.72	0.030	0.012	37.10
t_0	66.53	88.34	53.41	14.34	3.88	1.89	**3.75**	0.023	0.009	**25.54**
k-NN	71.78	93.64	51.89	16.20	4.86	2.56	5.57	0.025	0.0010	30.24
GaBP	**79.40**	**75.50**	**41.64**	**12.41**	**3.54**	**1.88**	3.98	**0.023**	**0.009**	26.67
$h = 30'$										
Mean(t)	58.01	123.5	71.56	22.08	6.17	3.51	7.72	0.030	0.012	37.10
t_0	53.58	119.3	73.19	19.39	5.26	2.47	5.09	0.029	0.012	31.32
k-NN	71.39	95.17	52.70	16.07	5.05	2.65	5.79	0.026	0.010	31.53
GaBP	**76.18**	**85.80**	**46.56**	**13.72**	**4.22**	**2.20**	**4.79**	**0.026**	**0.010**	**28.39**
$h = 60'$										
Mean(t)	58.01	123.5	71.56	22.08	6.17	3.51	7.72	0.030	0.012	37.10
t_0	38.07	175.42	111.6	30.34	7.17	3.40	7.26	0.037	0.016	44.32
k-NN	68.90	101.7	56.32	17.47	5.34	2.80	6.13	0.028	0.011	34.32
GaBP	**72.09**	**95.79**	**52.32**	**15.35**	**4.90**	**2.55**	**5.69**	**0.027**	**0.010**	**30.86**

Table 4: Results on the PeMS dataset for different horizons of forecasting, for flow, speed and occupancy measurements.

are active more than 10% of the time over the training period. We end up with an incomplete vector of $3 \times 645 = 1935$ variables, corresponding to speed, flow and occupancy measured every 5 minutes. To make the flow prediction error comparable to previous ones, we aggregate 5 minutes flow of vehicles into 15 minutes flow measurements. Forecasts concerning speed or occupancy are performed on raw measurements. The first 8 months of data are used to train the model and the last 3 months for testing. Days are grouped in only two clusters, one for Monday–Friday and one for Saturday–Sunday. For the architecture of the model, we again take $(n_p, n_f) = (3, 1)$ for an overall number of variables equal to 7740. This is the largest model that we have trained for this paper. To give an idea of the computational cost, 4 hours (on an ordinary laptop) are necessary to train the model (the model building task described in Section 2.4), while each forecast for the test is performed within ~ 0.1 seconds (the prediction and inputation task).

There are two main differences with previous datasets: first, this one corresponds to freeway data, with a different structure of the underlying road graph; second, forecasting is now done on flow, speed and occupancy, while previous ones were mostly on flow. Looking first at the left of Figure 10, we see that most of the information needed to perform the forecast is obtained from the first $\rho \sim 30$–50% of observed segments. After that, all the curves remain mostly flat, which was not the case before. This behavior seems to be specific to speed and occupancy, and is not exhibited by flow variables. Regarding dependence w.r.t. horizon of prediction, the right panel of Figure 10 indicates a flattening of the mean error for GaBP, well below the Mean(t) baseline. Again we can interpret this difference as specific daily features captured by GaBP and not by the baseline.

Compared to other datasets, in Table 4 we see much lower GEH, higher RMSE or MAE but better MAPE performances regarding the flow. It is to be noted that the flow values involved in these highway data are much higher than for urban roads in Vienna and Turin, which certainly bias the RMSE and MAE but also GEH toward larger values. There is also, however, some modeling bias for PeMS due to a reduced training set, compared to the other datasets. Additionally, the data seems to have higher frequency noise in the case of PeMS. We also mention the difficulties encountered with k-NN, especially for this PeMS dataset. This is expected as the dimension increases. For the experiments reported in Figure 10, which is restricted to speed variables, the vector dimension is 2885. Presumably, this means that information has to be extracted locally: k-NN is a global predictor, since it compares global configurations, while the MRF proposed here does most of the inference via local connections. Considering fewer past layers, as is done to obtain the k-NN results of Table 4 with the flow, speed and occupancy variables being predicted separately, helps to improve the k-NN performance. Note that for GaBP, forecasting the flow, speed and occupancy variables altogether with the help of a single model does not lead to significant improvements w.r.t. separate models. Designing a more sophisticated traffic index, based on fundamental diagrams for instance, to be coupled in the Gaussian MRF might help to leverage the dependencies between variables of distinct types.

Now looking at the time series excerpt shown in Figure 11, we see that the GaBP predictor is able to anticipate, to some extent, sudden changes, as seen by looking at the t_0 predictor in contrast.

4 Error analysis and limits of the model

In this section, in order to estimate the margin and possible directions for improvement, we first discuss the main limit of accuracy of our model and then perform a data analysis of the errors made by the model. In particular, we would like to be able to separate components in the errors due to modeling bias from the intrinsic noise of the traffic phenomenon, which cannot be predicted. When looking at Figure 5, we observe for instance a high frequency noise in the time series of the actual data, which, if properly measured, would somehow give a lower bound to the minimal error that can be reached by any method.

Up to now, we have left aside the fact that our prediction μ_i based on GaBP in (4) comes with an uncertainty estimate through the variance σ_i (5). While μ_i is guaranteed to be the exact marginal of the model once GaBP has converged, σ_i on the other hand is only an approximate value of the true variance. Actually, these values can be exploited to deliver levels of confidence on our predictions. Applying the inverse map of (8) to the corresponding confidence interval, defined by say ± 1 standard deviation in copula space, we get straightforwardly a confidence interval of our prediction in the original space of the predicted variable, e.g. flow, speed or occupancy. As shown, for example, on the left of Figure 12, these confidence intervals are quite consistent and meaningful and may actually help to identify detector errors. Indeed, on the top left panel of the same figure, we see that the detector wakes up most probably in the middle of a time interval after being silent during 6 time steps. Then it delivers a clearly underestimated observation of traffic flow, 2 standard deviations away from our confidence interval. These confidence intervals reflect an estimation of some intrinsic noise of the traffic signal which might not be possibly predicted. In any case, it is comparable to the mean error that we finally end up measuring on our predictions.

That said, there are systematic errors that our model makes which we would like to identify and cure. The main source of error comes from the Gaussian copula hypothesis. Recall first that, after the transformation (8) is performed, each $Y_i^{t,\ell}$ taken individually is by construction and up to numerical precision, a standard normal variable. However, the joint distribution has no specific reason in general to be multi-variate Gaussian. In order to estimate how far from a multivariate Gaussian our model is, we consider the main directions of fluctuations of Y by extracting the dominant eigenmodes of the covariance matrix. Then for each of these modes $e_{.,\alpha}$, we compute

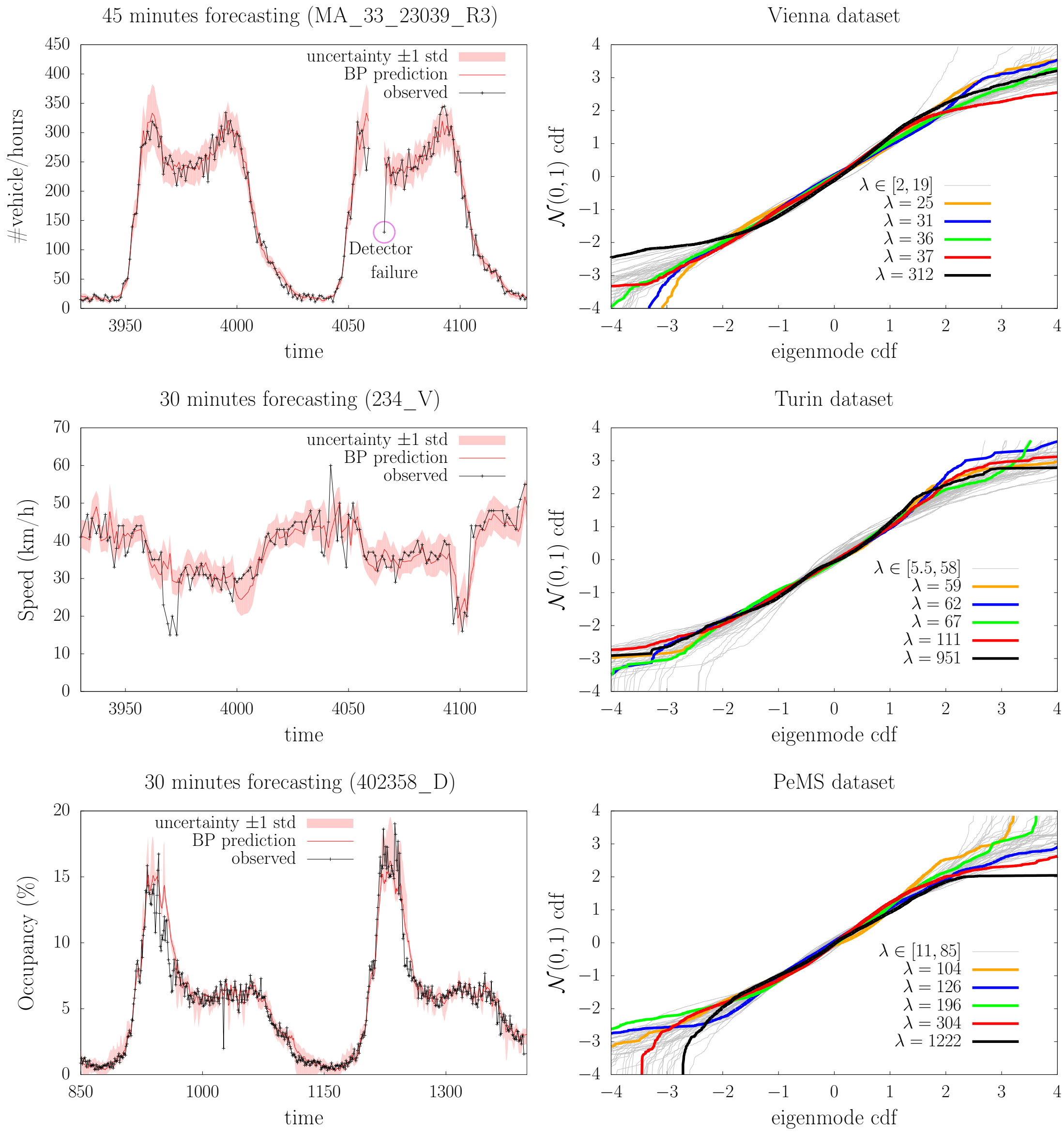

Figure 12: (left panel) Flow, speed and occupancy forecast, including uncertainty estimate delivered by GaBP, for a given CLOC respectively in the Vienna, Turin and PeMS datasets. The confidence interval corresponds to 1 std in copula space, corresponding i.e. to 84% of confidence that the true value is within this interval. (Right panel) Empirical cumulative distribution of the normalized projection of the (copula transformed) data along the 50 first principal modes against the cumulative distribution of a standard normal variable. Modes are ordered by decreasing eigenvalues λ_α (From top to bottom: Vienna, Turin and PeMS dataset)

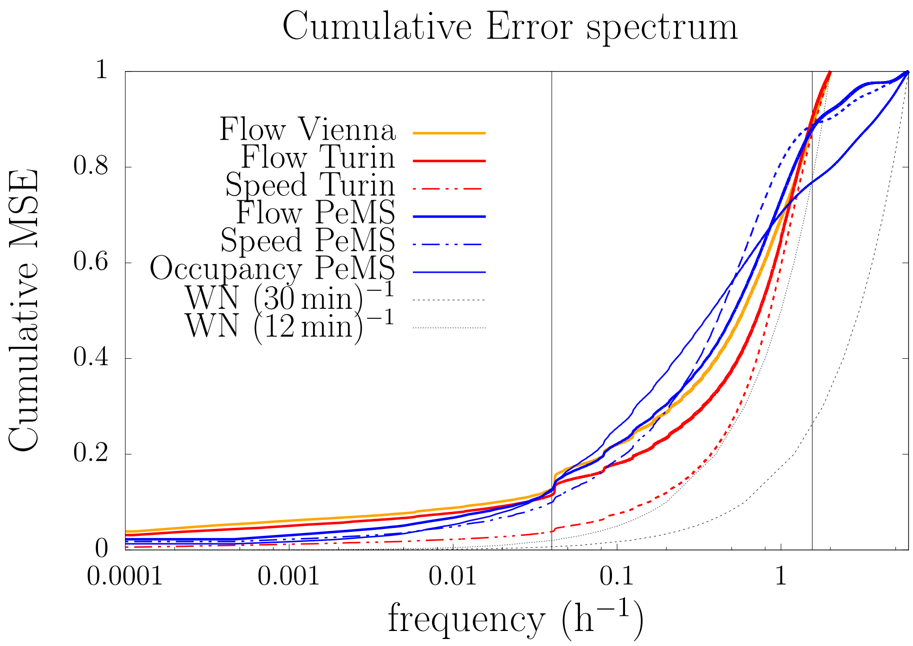

Figure 13: Cumulative power spectrum of the mean errors signal of the GaBP predictor, for the different datasets and type of variables. Vertical lines indicate one day and 40 min time scales. Dotted lines indicate a white noise (WN) reference with respectively $(30\,\text{min})^{-1}$ for Vienna and Turin, $(12\,\text{min})^{-1}$ for PeMS as maximum possible frequency.

the corresponding cumulative distribution $P(z_\alpha^{t,\ell}/\sigma_\alpha < x)$ of the components

$$z_\alpha^{t,\ell} = \sum_{i=1}^{N} e_{i,\alpha} Y_i^{t,\ell}$$

of the data normalized by the standard deviation

$$\sigma_\alpha \stackrel{\text{def}}{=} \sqrt{\mathbb{E}\big[(z_\alpha^{t,\ell})^2\big]}$$

along these modes. Figure 12 shows for the various datasets how this distributions compares to the expected cumulative distribution of a standard normal variable. As we can see, for all datasets the alignment is pretty good for most of the dominant modes over more or less 2 standard deviations. Beyond that, the model shows inadequacy in the distributions tails which are clearly non-Gaussian. Tests done without the clustering preprocessing step show as expected degraded performance, with a much more pronounced deviation from the normal distribution and a multi-modal behaviour. This underlines the importance of the clustering of the data beforehand. Another way to look at the results is to consider the Fourier spectrum of the error signal, and in particular how the error is distributed along the different frequencies. Figure 13 shows this for the different datasets and types of variables, compares this to a white noise predictor, that is a predictor with random decorrelated errors. First, we see that the departure from the white noise predictor is much more pronounced for PeMS than for Vienna and Turin. What is also noticeable is the finite contribution of very low frequencies, especially for Vienna and Turin. When looking more precisely at the data, this actually corresponds to detectors which, for a long period, send values completely at odds with the ones observed during training. These bad behaving detectors may either correspond to corrupted ones, or to drastic changes of the traffic conditions on the corresponding segment, because of road work for instance. At the other end of the spectrum, there are contributions from high frequencies which seem rather uniform beyond a point corresponding to $f = (40\,\mathrm{min})^{-1}$, where all curves intercept (except the occupancy error of PeMS). This corresponds to the time scales between 30 minutes (for Vienna and Turin) or 10 minutes (for PeMS) and 40 minutes. When looking at the data, we observe such a high frequency noise on most of the signals, and it is particularly pronounced on the PeMS occupancy variables. We don't know whether something can be done about this or whether this noise corresponds to natural and uncorrelated fluctuations of the traffic phenomenon. A more careful analysis will provide the answer, but clearly in our approach these contributions are interpreted as random noise by our model. Then there remains the domain corresponding to time scales ranging between roughly 45 minutes and one day, represented by the two vertical lines on the plot. Overall, as seen in Figure 13, this corresponds to 70 to 80% of the total error. It is in this region that we can mostly pinpoint modeling biases. For Vienna and Turin the discrepancy w.r.t. the white noise predictor remains somewhat constant, which confirms that the model is behaving correctly. For PeMS on the other hand, the discrepancy increases drastically in this region, which confirms a global modeling problem. Our interpretation is that in the case of the PeMS dataset, the test set is not sufficiently representative of the training set (different months of the year). Also, the part of the spectrum corresponding to daily time scales indicates that the day profile prototypes (week-day and week-end for PeMS) are not precise enough, and more training data would clearly help to build refined clusters of the days in that case.

Finally, we examine a subset of the results from a transportation engineering point of view. We consider a set of 22 CLOCs from the Turin dataset for which both flow and speed are available. For each of these CLOCs, the object of interest is the empirical fundamental diagram, i.e. the flow versus the density (obtained as a simple division between the flow and the speed) of all the data of the year 2016. Then, each data point is labeled as describing either a fluid (hypocritical) or a congested (hypercritical) traffic condition. The manual labeling of the traffic conditions has been performed by three different experts, all of them trained to recognize visually the traffic conditions. These three classifications are not 100% identical, since some boundary situations are difficult to classify. In those cases the majority label is used, which is always possible with an odd number of outcomes. The main use for this manual labeling is to assess the expectation that congested traffic conditions are harder to predict than fluid ones. To this end, we measure the performance of the flow forecast against actual data through the GEH indicator, evaluated either on the congested states, the fluid ones or all of them, as shown in Table 5. More insightful results are visible in Figure 14, where the cumulative distributions of GEH is reported for different forecast horizons and the three different methods, showing the difference between the overall aggregation of all traffic state conditions and the congested ones only. As expected, the conclusion is that, at all forecast horizons, and for each methodology, the congested states are more difficult to predict. However, GaBP achieves consistently a GEH score that is smaller in distribution than the other methods.

%GEH < 5	All Data	Fluid	Congested
$h = 15'$			
Mean(t)	70.93	71.11	63.12
t_0	79.18	79.37	72.03
GaBP	**88.28**	**88.36**	**85.32**
$h = 30'$			
Mean(t)	70.80	70.96	62.93
t_0	66.71	66.83	61.08
GaBP	**85.71**	**85.79**	**81.43**
$h = 45'$			
Mean(t)	71.09	71.26	62.99
t_0	57.47	57.50	55.73
GaBP	**84.81**	**84.95**	**79.90**
$h = 60'$			
Mean(t)	70.94	71.13	62.61
t_0	50.34	50.28	53.08
GaBP	**83.14**	**83.28**	**77.30**

Table 5: Results on the Turin dataset manually labeled for different horizons of forecasting, for flow measurements and considering all the data together and grouped by traffic state conditions.

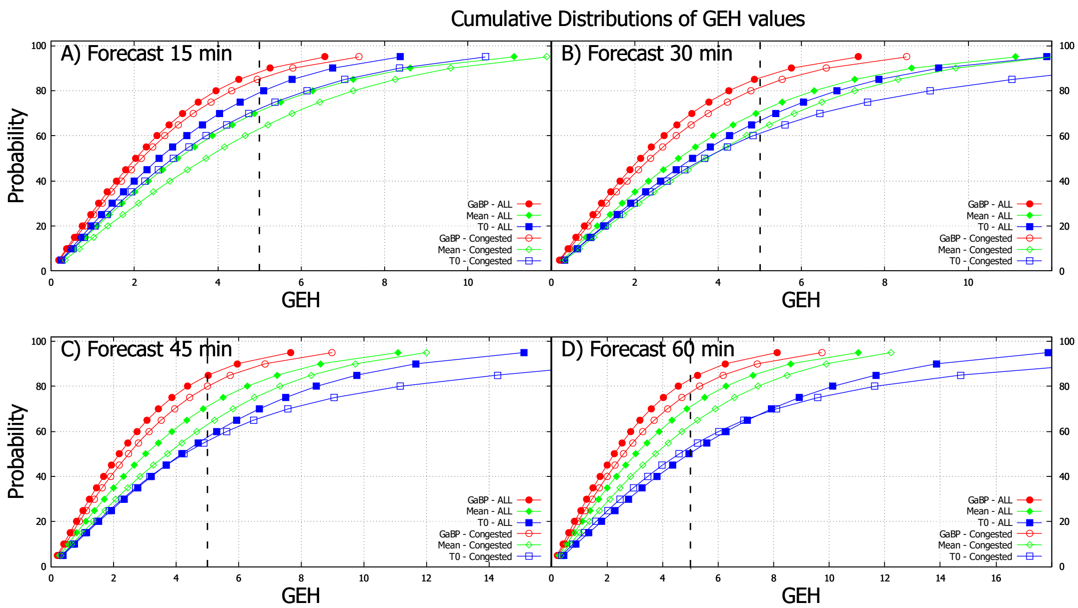

Figure 14: Cumulative distributions of the GEH for different forecast horizons and the three different methods. The vertical black dashed line indicates the value of GEH equal to 5, under which a forecast result is considered good.

5 Conclusions

The method presented in this paper is intended both to provide valuable predictors of traffic variables on large traffic networks, in order to feed traffic management systems for instance, and to provide some understanding of the statistical properties of traffic. With the advent of big data in the domain of traffic management, it is becoming possible to analyze these properties in detail at the macroscopic level. In our analysis of the model, we left aside the structure of the network that is generated, which could potentially let us visualize the information flow among the variables. The focus might concentrate more on this in some future work with help of dedicated data analysis techniques of graph structured data. Various sources of improvement can also be expected, like for instance, working directly with variables combining flow speed and occupancy when available, in order to model more accurately the statistical properties of the fundamental diagrams at the level of each segment. Another line of research concerns the question of corrupted data, which was merely touched upon in this paper. Our model offers the possibility via the message passing structure of estimating whether an input variable should be taken more or less seriously depending on the information coming from other variables. Integrating this information in the belief propagation schema with the help of the soft belief setting mentioned in Section 2.4 might help us to improve the accuracy of the forecast by automatically discarding suspicious observations.

Acknowledgments

The authors would like to thank the municipality of Vienna (Austria) and the Italian company 5T, which manages the mobility in Piemonte region (including Turin), for sharing their data with us.

References

[1] E.I. Vlahogianni, M.G. Karlaftis, and J.C. Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43, Part 1:3 – 19, 2014. Special Issue on Short-term Traffic Flow Forecasting.

[2] A. Ermagun and D. Levinson. Spatiotemporal traffic forecasting: review and proposed directions. *Transport Reviews*, 0(0):1–29, 2018.

[3] M. Lippi, M. Bertini, and P. Frasconi. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 14:871–882, 06 2013.

[4] G. Fusco, C. Colombaroni, and N. Isaenko. Short-term speed predictions exploiting big data on large urban road networks. *Transportation Research Part C: Emerging Technologies*, 73:183 – 201, 2016.

[5] S. Sun, R. Huang, and Y. Gao. Network-scale traffic modeling and forecasting with graphical lasso and neural networks. *Journal of Transportation Engineering*, 138(11):1358–1367, 2012.

[6] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2015.

[7] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, 17(7), 2017.

[8] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

[9] N.G. Polson and V.O. Sokolov. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 79:1 – 17, 2017.

[10] Y. Duan, Y. Lv, Y.L. Liu, and F.Y. Wang. An efficient realization of deep learning for traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 72:168 – 181, 2016.

[11] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transportation Research Part C: Emerging Technologies*, 105:297 – 322, 2019.

[12] S. Kataoka, M. Yasuda, C. Furtlehner, and K. Tanaka. Traffic data reconstruction based on Markov random field modeling. *Inverse Problems*, 30(2):025003, 2014.

[13] Y. Hara, J. Suzuki, and M. Kuwahara. Network-wide traffic state estimation using a mixture Gaussian graphical model and graphical lasso. *Transportation Research Part C: Emerging Technologies*, 86:622 – 638, 2018.

[14] C. Furtlehner, J.M. Lasgouttes, and A. de La Fortelle. A belief propagation approach to traffic prediction using probe vehicles. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 1022–1027. IEEE, 2007.

[15] V. Martin, J.M. Lasgouttes, and C. Furtlehner. Latent binary MRF for online reconstruction of large scale systems. *Annals of Mathematics and Artificial Intelligence*, pages 1–32, 2015.

[16] C. Furtlehner and A. Decelle. Cycle-based cluster variational method for direct and inverse inference. *Journal of Statistical Physics*, 164(3):531–574, 2016.

[17] V. Martin, C. Furtlehner, Y. Han, and J.M. Lasgouttes. GMRF estimation under topological and spectral constraints. In *ECML PKDD Proceedings,Part II*, pages 370–385, 2014.

[18] A. Attanasi, L. Meschini, M. Pezzulla, G. Fusco, G. Gentile, and N. Isaenko. A hybrid method for real-time short-term predictions of traffic flows in urban areas. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 878–883, 2017.

[19] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*. Morgan Kaufmann, 1988.

[20] D. Bickson. *Gaussian Belief Propagation: Theory and Application.* PhD thesis, Hebrew University of Jerusalem, 2008.

[21] Y. Weiss and W.T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Comput.*, 13(10):2173–2200, 2001.

[22] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *JMLR*, 9:485–516, 2008.

[23] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[24] K. Scheinberg and I. Rish. Learning sparse Gaussian Markov networks using a greedy coordinate ascent approach. In *ECML-PKDD*, 2010.

[25] D.M. Malioutov, J.K. Johnson, and A.S. Willsky. Walk-sums and Belief Propagation in Gaussian graphical models. *JMLR*, 7:2031–2064, 2006.

[26] T.P. Speed and H.T. Kiiveri. Gaussian Markov distributions over finite graphs. *The Annals of Statistics*, 14(1):138–150, 1986.

[27] C. Hsieh, M. A. Sustik, I. S. Dhillon, and K. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. In *NIPS*, 2011.

[28] C. Chen, J. Kwon, J. Rice, A. Skabardonis, and P. Varaiya. Detecting errors and imputing missing data for single-loop surveillance systems. *Transportation Research Record*, 1981:160–167, 2003.

RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399