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

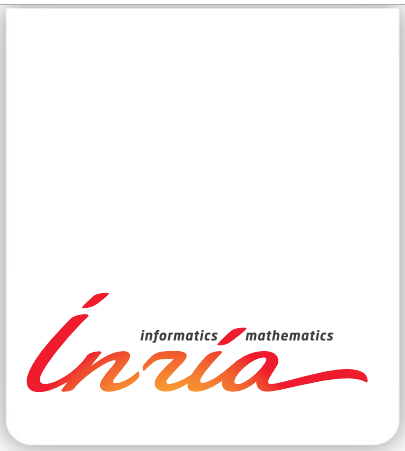
RESEARCH REPORT

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 — December 2018 — 27 pages

Abstract: The probabilistic forecasting method described in this work is designed to leverage spatial and temporal dependency of urban traffic networks in order to provide accurate predictions for an horizon up to a few 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 whole network in one pass with an execution time scaling linearly with the size of the network. The method consists in two learn a sparse Gaussian Copula of traffic variables, compatible with the Gaussian belief propagation algorithm. The model is learned automatically from an historical dataset through an iterative proportional scaling procedure well suited to impose this compatibility constraint. It is tested on three different datasets of increasing sizes ranging from 250 to 2000 detectors corresponding to flow or/and speed and occupancy measurements. The results show very good ability to predict flow variables and a reasonably good performance on speed or occupancy variables. Some element of understanding of the observed performance are given by a careful analysis of the model allowing to some extend to disentangle modelling bias from 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.
- Forecast of the whole network state with linear execution time with the size of the network.

Contents

1 Introduction

The rapidly evolving sector of traffic management information systems makes 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 as fluid as possible. They generally operate on main axes or principal sub-network 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 along them. To be accurate, these flow models must have a high level of detail, that is translated into an enormous amount of parameters to be calibrated. But the calibration of those parameters is so challenging and time consuming that the effectiveness of flow models can decrease sharply. In this respect, data driven models can leverage the calibration effort and provide real-time feed to the flow models to improve its results.

Besides this motivation, the massive increase of traffic data has triggered during the last decade a surge in interest for 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 is potentially offering 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]. A lot of methods are targeted toward independent segment modeling. Methods trying to leverage spatial dependencies are less numerous but many have appeared recently [2]. If we focus more specifically on forecasting models which attempt to address the problem at the network scale, the requirements we can think of for such models to be ideally deployed in online applications are the following

- *accuracy*: predictions should be significantly better than a simple persistent predictor combined with historical day-time dependent average for instance, in use when data are incomplete.

- *missing data*: we cannot expect to have at any time a complete information of the network state, which means that both the learning and the running of the model have to be able to be done in a setting with missing data.

- *scaling*: the model should scale up to high systems size, i.e. networks of the size of an agglomeration, where number of road segments to be tackled can be around one or two hundreds thousands. Actually, if we think in terms of detectors, this requirement might be lower. At the moment, 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 been adapted recently to this context, e.g. for treating floating car data (FCD) at small scale (120 points location in central Rome) [4]. While yielding a good level of interpretability, this type of methods do not seem suitable to scale up to large network sizes. In order to capture local spatial features of traffic patterns, several studies (see e.g. [5]) proposed hybrid machine learning methods involving neural network and L1 regularization of the weight matrix connecting the input to the hidden layer. They remained however limited to scale of the order of a few hundred of 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 performance improvement over traditional methods, though often demonstrated on small scale problems involving again a few hundreds of variables, presumably due to the heavy computationally training procedure [9].

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

This paper proposes a different direction, well adapted to 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 can be conveniently used in particular for data imputation, even when observability of each segment is not known in advance. One difficulty is to learn 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 variable from the observed ones. For instance, it is shown in [11] that a simple Gaussian multivariate model with very few parameters can already perform better than simple interpolation methods at large scale with a cost linear w.r.t. system size, thanks to mean-field techniques. More refined models, like proposed in [12] based on a mixture of sparse Gaussian random variables, can be learned 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. Some years ago we started to investigate [13] 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 [14] or Gaussian copula models, both coming with an associated learning algorithm to generate compliant models, respectively with Generalized Belief Propagation for binary variables [15] and Gaussian Belief Propagation for real-valued ones [16]. Our purpose here is to wrap up some of the techniques developed in these past works by focusing more specifically on the Gaussian copula as a forecasting approach in a setting with missing data. We perform comprehensive experimental tests on various real traffic dataset in order to illustrate in various conditions the effectiveness of this method.

Our approach is designed to have the following desirable features which to our knowledge are not altogether met by any other method at the moment:

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

To some extent, these features meet the requirements listed earlier for a network-wide forecasting method. Concerning the scaling to large systems sizes, we are actually hampered by the learning 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 learning method on its side has a cubic scaling with systems size. This means that we can learn 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 can be possibly proposed, but they shall not be discussed in the present paper.

The paper is organized as follows: the needed materials like the Gaussian belief propagation algorithm in Section 2.1 used for inference, the $\star$-IPS algorithm used for constructing the model in Section 2.2 and the Copula encoding 2.3 are given first before describing the workflow of the method in Section 2.4. Results of experiments performed on three different traffic dataset with various features are then reported and analyzed in Section 3. Comparison will be made in particular with results from single detector times series [17] forecast models to measure the advantage taken from spatial dependencies. In Section 4 is performed a statistical analysis of the errors of the model in which we try to separate intrinsic uncertainty to the traffic phenomena from systematic errors based on modeling issues.

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) [18] does this for all variables in one single procedure, by remarking 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 among them as a fixed point equation. Iterating the following message updates

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

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

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

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. Indeed, 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 [13, 14]. Our focus here is on Gaussian variables, for which the algorithm takes on a specific form, referred to as Gaussian Belief Propagation (GaBP) [19]. The factors are naturally parameterized as

$$\psi_{ij}(x_i, x_j) = \exp\big(-A_{ij}x_i x_j\big),$$

$$\phi_i(x_i) = \exp\big(-\frac{1}{2}A_{ii}x_i^2 + h_i x_i\big).$$

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

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

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

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

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

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 reads

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

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 [20], with computational time $O(N\log(N))$ roughly linear with 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 which is compatible with GaBP. By

combining various methods proposed in the context of sparse inverse covariance matrix estimation [21, 22, 23], a way to do that as been elaborated in [16] in the form of the ⋆-IPS algorithm. 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. Indeed, 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 [24], can guarantee the compatibility with GaBP-based inference. ⋆-IPS incorporates these explicitly, by combining an approach based on the iterative proportional scaling (IPS) procedure [25], with block-updates techniques used in [21, 22]. 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 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 the distinction between frustrated/non-frustrated loops, i.e. loops for along which the product of partial covariances $(-A_{ij})$ is negative.
- Spectral constraints like walk-summability [resp. weak walk-summability] involve definite positiveness of matrix $\mathbf{diag}(A) - |A - \mathbf{diag}(A)|$ [resp. matrix $2\,\mathbf{diag}(A) - A$], where $\mathbf{diag}(A)$ 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 [21] and refined in [22]. In practice, ⋆-IPS alternates many link additions, corresponding to significant mean connectivity increase, with block coordinate descent procedures. Overall, sparse precision matrix of good likelihood are generated in $O(N^3)$ steps, with the advantage of having available all the optimization path, by mean of many graphical models of intermediate connectivity. Note finally that we have discarded the standard way for generating sparse precision matrix based on Lasso penalty [22, 26]. 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 [16].

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 $\delta_t = 15\,\text{min}$ 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

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

which represents a centered and normalized variable for given τ and ℓ. From this transformed histrical 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 of 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 : the creation of the model, and its use to produce detailed forecasts. We give below the details of our implementation.

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.

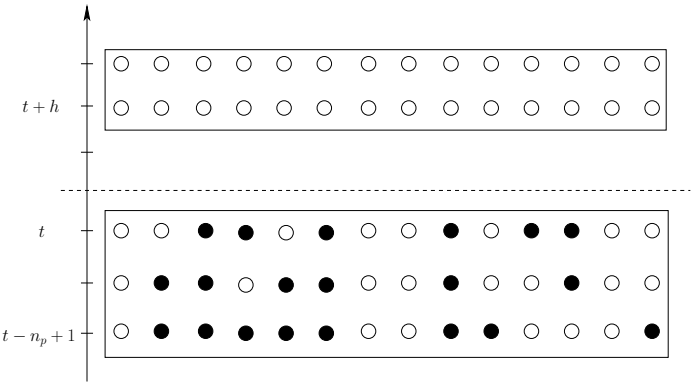


Figure 1: Time layers setting of one basic space time configuration. Each circle represents one single variable at a given time, filled circles correspond to observations.

(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 needed 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 the cdf F_i of $U_i^{t,\ell}$ aggregated over all t is computed and approximate it by an invertible monotonous linear piece wise 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 data is incomplete, some regularization can be required if some modes are associated to with large negative eigenvalues. In such case the eigenvalue is replaced by its absolute value.

(MB3) MRF model The model is built using the $\star$-IPS algorithm [16], which takes as input the regularized covariance matrix and generates an almost continuous set of models with increasing mean connectivity $\bar{d}$. The model of choice $\bar{d} = d^\star$ is picked 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).

2.4.2 Reconstruction and prediction (RP, 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

(RP1) 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. Neighbor 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 unknown (In the experiments presented later only the strong beliefs will be used).

(RM2) 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 on 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 a way or another. Indeed h is imposed by the task and n_f is an optional choice, with 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 simplifies greatly their tuning.

Complexity and affordable systems sizes Since $\star$-IPS has a cubic complexity, the model building task (MB) has a running time which 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 to be able to learn 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 reconstruction and prediction task (RP), in our setting GaBP can converge on networks of size 10^5 within a few tens 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 a few 10^5 of detectors.

3 Experiments

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

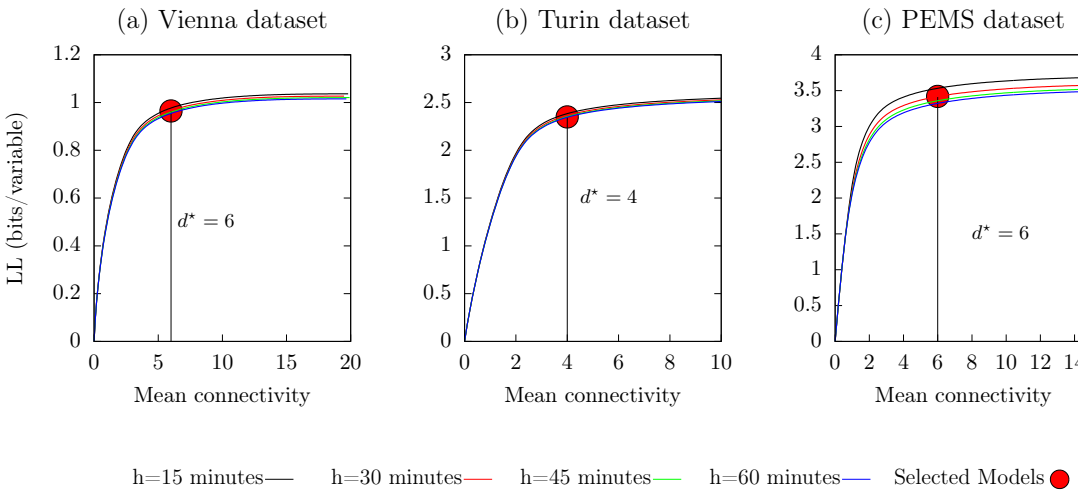


Figure 2: Log likelihood of the sparse Copula models as a function of mean connectivity at various forecast horizons for Vienna, Turin and PeMS datasets

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.

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

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

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

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

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

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 statistics of interest is thus the percentage of GEH < 5.

The expectation 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.

	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 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$^{\text{st}}$.

3.2 Baseline predictors

In order to provide some reference points of comparison to 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 random walk predictor. It is equal to the last observed value within the past window time of the detector. If no observation is found within the past window time, 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 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 network (NN) and clustering of day profiles have been previously tested [17] 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 concern urban traffic in the agglomeration of Vienna (Austria). It consists of flow data collected during 4 years (2011–2014) from 292 count locations (CLOC) measuring vehicles/15 minutes every 15 minutes. Since these detectors are quite often down, we select arbitrarily those which are up during at least 10% of the time. This leads to retain 266 CLOCs out of 292. With this setting, at any given time, we have typically around 65% of detectors which are up. This means that, in addition to forecasting a certain number of layer 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 $N_t = 96$

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

The settings for $\star$-IPS remain the same through all experiments on this dataset. Our regularization option consists to avoid frustrated short loops of size up to 5 (see Section 2.2) and the selected model is always of mean connectivity $d = 6$ at a point where the log likelihood slope starts to flatten (see Figure 2).

For this setting, several architectures lead to similar performance, 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 on Figure 3 are obtained with a model with 4 past layers and one single specialized future layer. Results summarized in Table 2 are 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 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.

Since 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 observation available some variables of the past and present layers in order to have a given density ρ of observed variables per layer as sketched on Figure 1. The result is shown on Figure 3 (left). This plot shows that already a lot of information is gained by observing only a small fraction of variables, say $5 - 10\%$ and after that the forecasting and reconstruction error continue to decrease more slowly but steadily, while for the k-NN model it saturates after $\rho \geq 0.2$. The forecasting performance of the model as a function of the horizon is shown on Figure 3 (right). With respect to the Mean(t) predictor, we see the error reduced by 33% on the short-term prediction. In addition, our model is providing 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 the 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 on Figure 4. We can see that GaBP follows very well without any delay the changes in traffic conditions and realizes 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 in some sense the benefits of leveraging spatial dependencies. From these results, we see some gain is visible at all horizons regarding all metrics of evaluation, rather marginal for $h = 15'$ but rapidly sizable when the horizon increases. By looking more specifically at individual CLOC errors, those having less than 70% of GEH$<$5 are either bad behaved sensors clearly sending corrupted data, either sensors for which a systematic bias is present in our Mean(t) baseline reflecting possibly a long lasting modification in the traffic conditions for the corresponding segment.

3.4 Turin dataset

The second data set is again about urban traffic, in the agglomeration of Turin (Italy). 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 on Figure 5. Two years of data (2015–2016) have

	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 [17]).

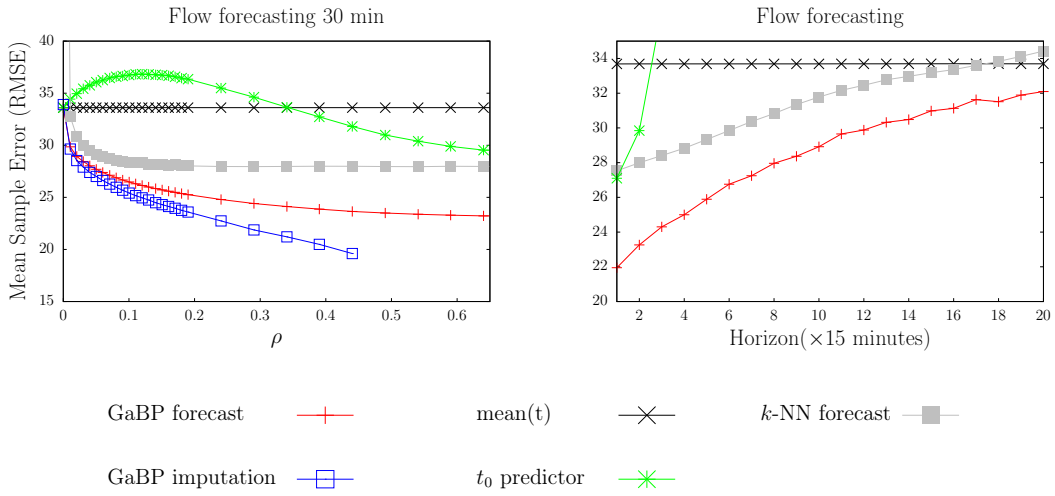


Figure 3: Vienna dataset: (left panel) average flow forecasting error for a given time lag of 30 minutes, as a function of the fraction ρ of observed variables in the past time layers, averaged over 5000 test samples. The reconstruction error for missing data is also shown (in blue). A point of comparison 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).

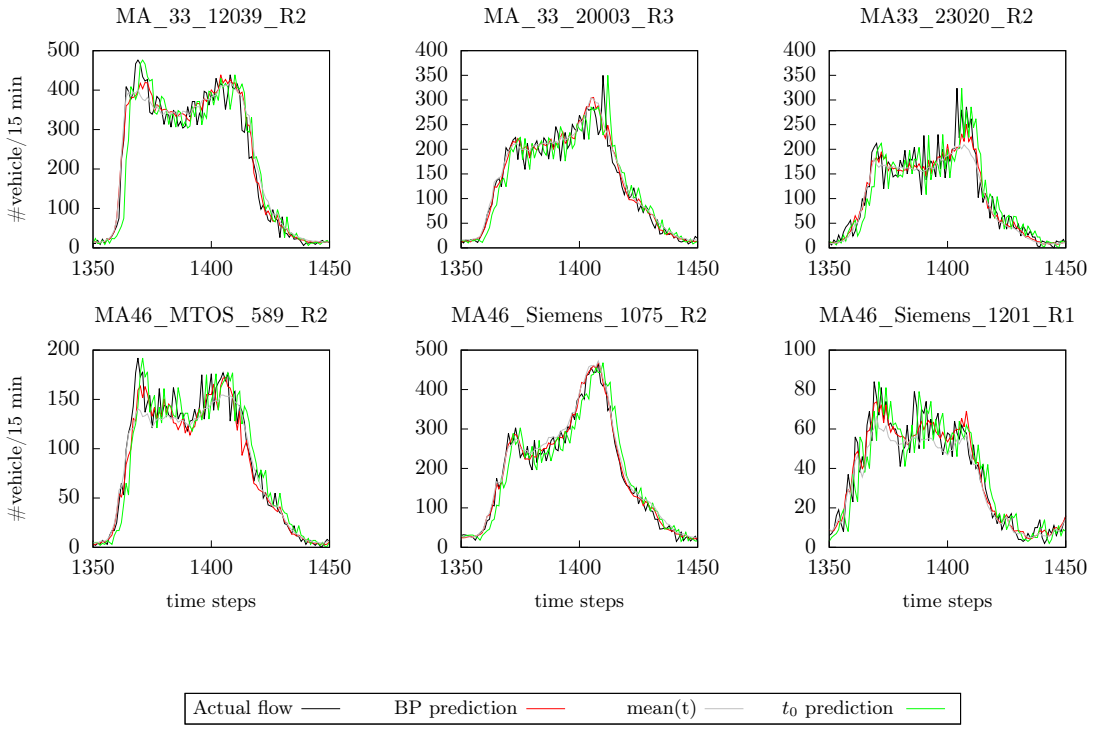


Figure 4: Vienna dataset: excerpt of flow time-series along with GaBP and t_0 prediction for 30 minutes horizon for 6 different CLOCs.

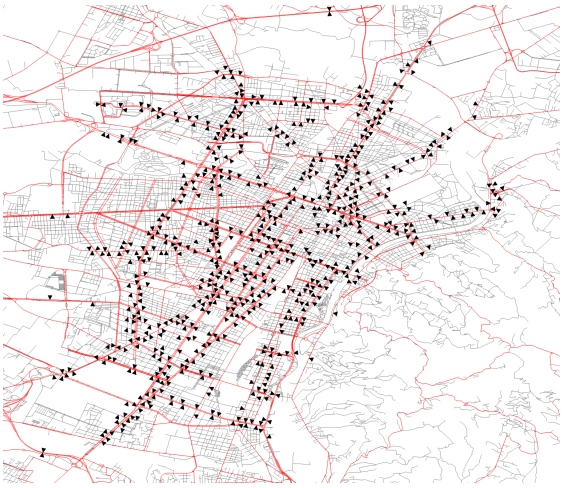


Figure 5: Location of the CLOCs in the center of Turin

	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.

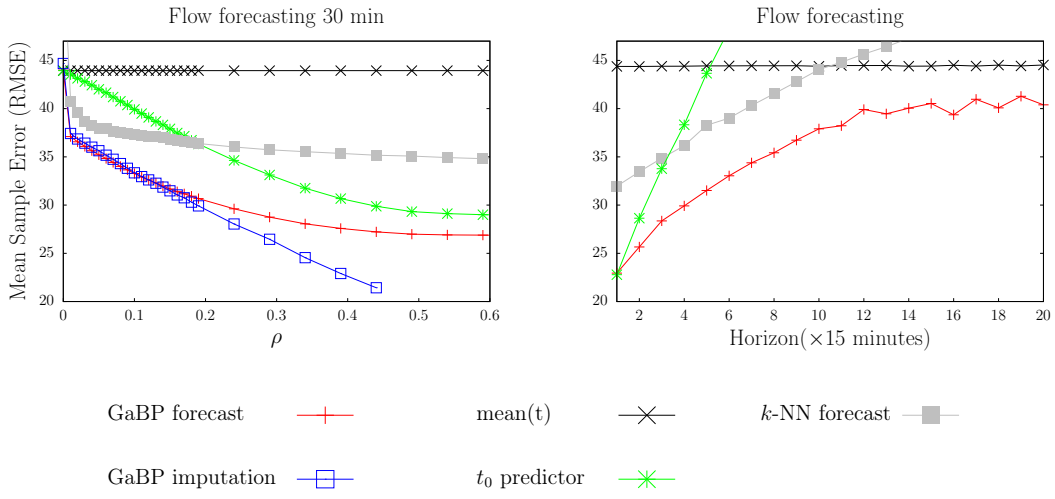


Figure 6: Turin dataset: (left panel) average flow forecasting error for a given time lag of 30 minutes, as a function of the fraction ρ of observed variables in the past time layers, averaged over 5000 test samples. The reconstruction error for missing data is also shown (in blue). A point of comparison 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.

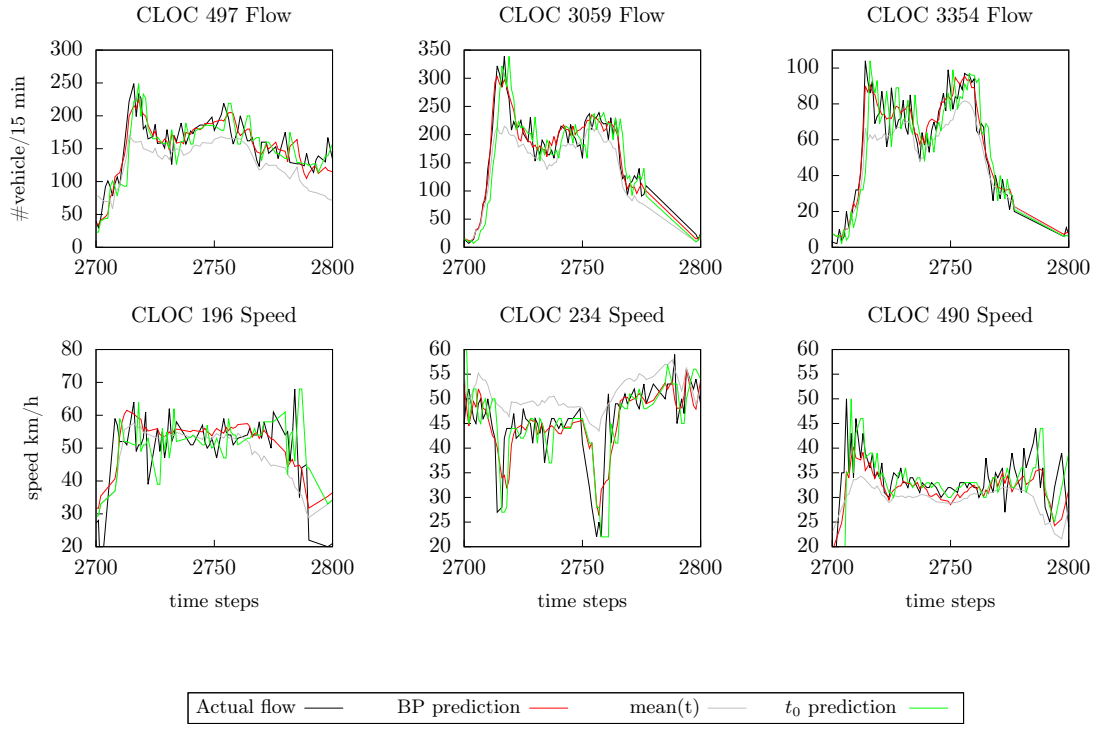


Figure 7: Turin dataset: excerpt of flow time-series along with GaBP and t_0 prediction for 30 minutes horizon for 6 different CLOCs of flows or speed detectors

been collected. Like for the previous dataset, we select detectors based on an activity threshold of 10%. We end up with 685 detectors, out of which 566 correspond to flow and 119 to speed measurements. With this setting, the ratio of available data is around a 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 more noisy Mean(t) baseline (around 45 in RMSE instead of 40 for Vienna). Figure 6 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 size equal to 2740 variables. Table 3, by contrast, compares different methods to a multi-steps a head 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 1h horizon. All indicators for the flow are very similar to the ones 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 30 minutes 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 amount of speed measurements leads to a very noisy signal.

On Figure 7 are shown 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 on 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 San Francisco Bay area during the first 11 months of 2013. These data are provided without any missing value, thanks to an automatized interpolation procedure [27]. For each station are reported as many values as freeways lines and an aggregated value coming with an observation rate. This rate corresponds to the percentage of actual observations composing 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 one, we retained only data coming with a high observation rate ($> 80\%$), other data being considered as missing. On top of this filtering procedure, we select detectors being 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. Forecast concerning speed or occupancy are performed on raw measurements. The first 8 months of data are used for training 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. Concerning the architecture of the model, we take again $(n_p, n_f) = (3, 1)$ 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 learn the model (model building task described in Section 2.4), while each forecast for the test is performed within ~ 0.1 seconds (prediction and reconstruction 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, forecast is now done on flow, speed and occupancy, while previous ones where mostly about flow. Looking first at the

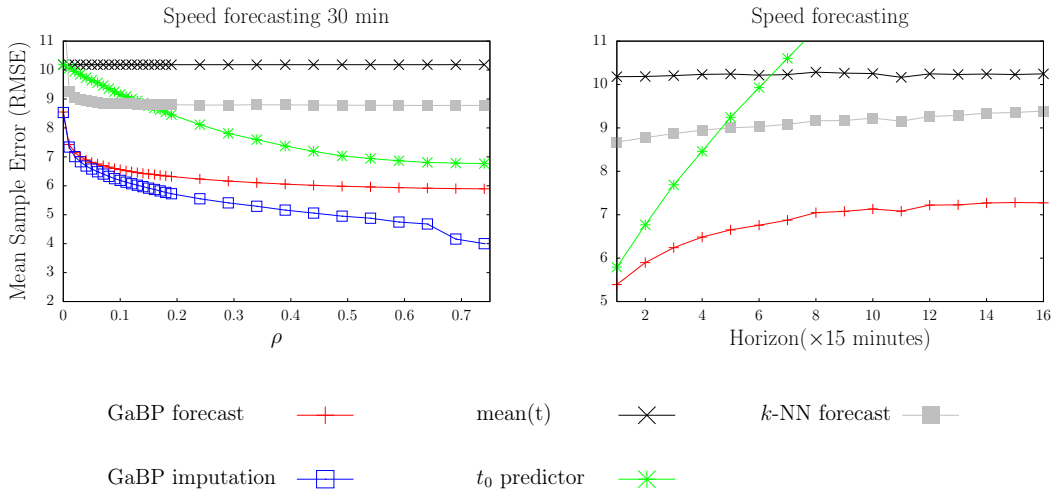


Figure 8: PeMS dataset: average speed forecasting error for a given time lag of 30 minutes, as a function of the fraction ρ of observed variables in the past time layers, averaged over 1000 test samples (left). Average speed forecasting error as a function of time lag (right) at maximum possible observation rate $\rho \sim 0.75$, (averaged over 1000 test samples). The reconstruction error for missing data is also shown (in blue). A point of comparison is given by the k-NN predictor ($k = 50$).

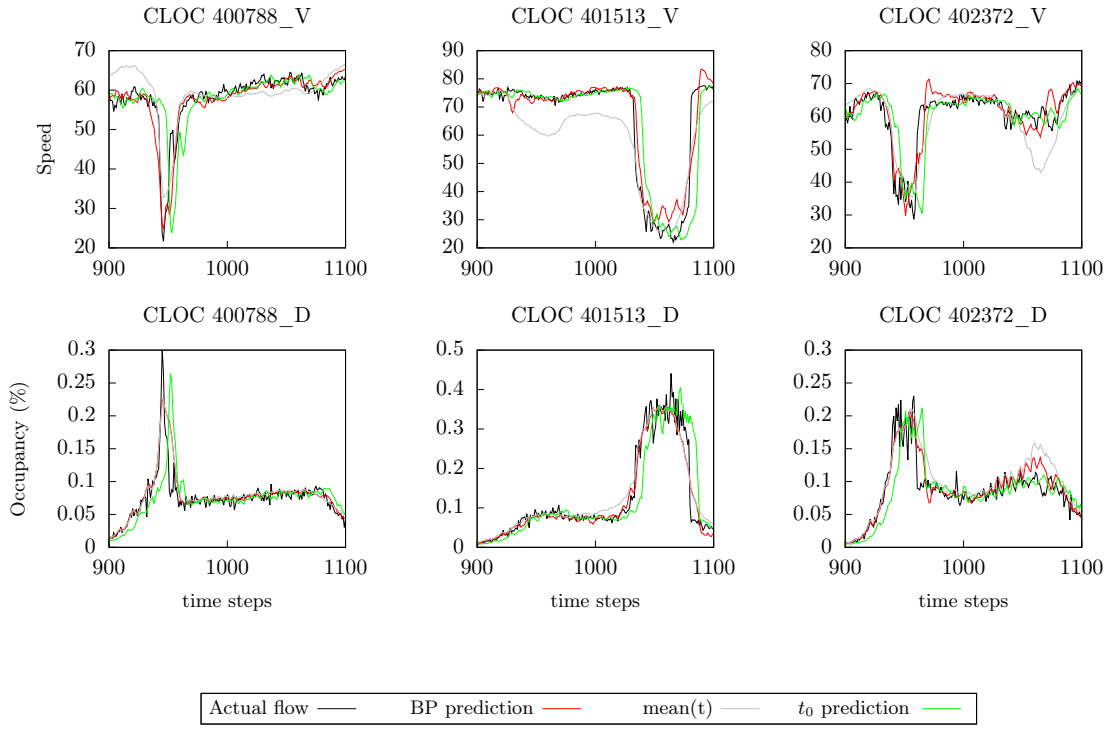


Figure 9: PeMS dataset: excerpt of speed time-series along with GaBP and t_0 prediction for 30 minutes horizon for 6 different speed detectors

	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 both flow, speed and occupancy measurements.

left of Figure 8, we see that most of information needed to perform the forecast is obtained from the first $\rho \sim 30-50\%$ of observed segments. After that, all the curves remains mostly flat, which was not the case before. This behavior seems to be specific to speed and occupancy, and is not exhibited by to flow variables. Regarding dependence w.r.t. horizon of prediction, the right panel of Figure 8 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 we see in Table 4 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 that for urban roads in Vienna and Turin, which certainly bias the RMSE and MAE but also GEH toward larger values. There are however also some modelling 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 mention also the difficulties encountered with k-NN, especially for this PeMS dataset. It is expected as the dimension increases. For the experiments of Figure 8, 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 less past layers, as is done to obtained 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 help of a single model does not leads to significant improvements w.r.t. separated models. Designing 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 on Figure 9, we see that the GaBP predictor is able to anticipate to some extent suddent changes, as seen by looking at the t_0 predictor by contrast.

4 Error analysis and limits of the model

In this section, in order to estimate the margin and possible directions for improvement, we discuss first 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 phenomena, which cannot be predicted. When looking at Figure 4, 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 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 instead 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 $\pm$ 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 instance on the left of Figure 10, these confidence intervals are quite consistent and meaningful and may actually help to identify detector errors. Indeed, on 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 times steps. Then it delivers a clearly underestimated observation of traffic flow, 2 std away from of our confidence interval. These confidence 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 taken aside, there are systematic errors that our model makes which we would like to identify and possibly 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 is our model, 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 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. On Figure 10 is shown for the various datasets how this distributions compare 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 std. 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.

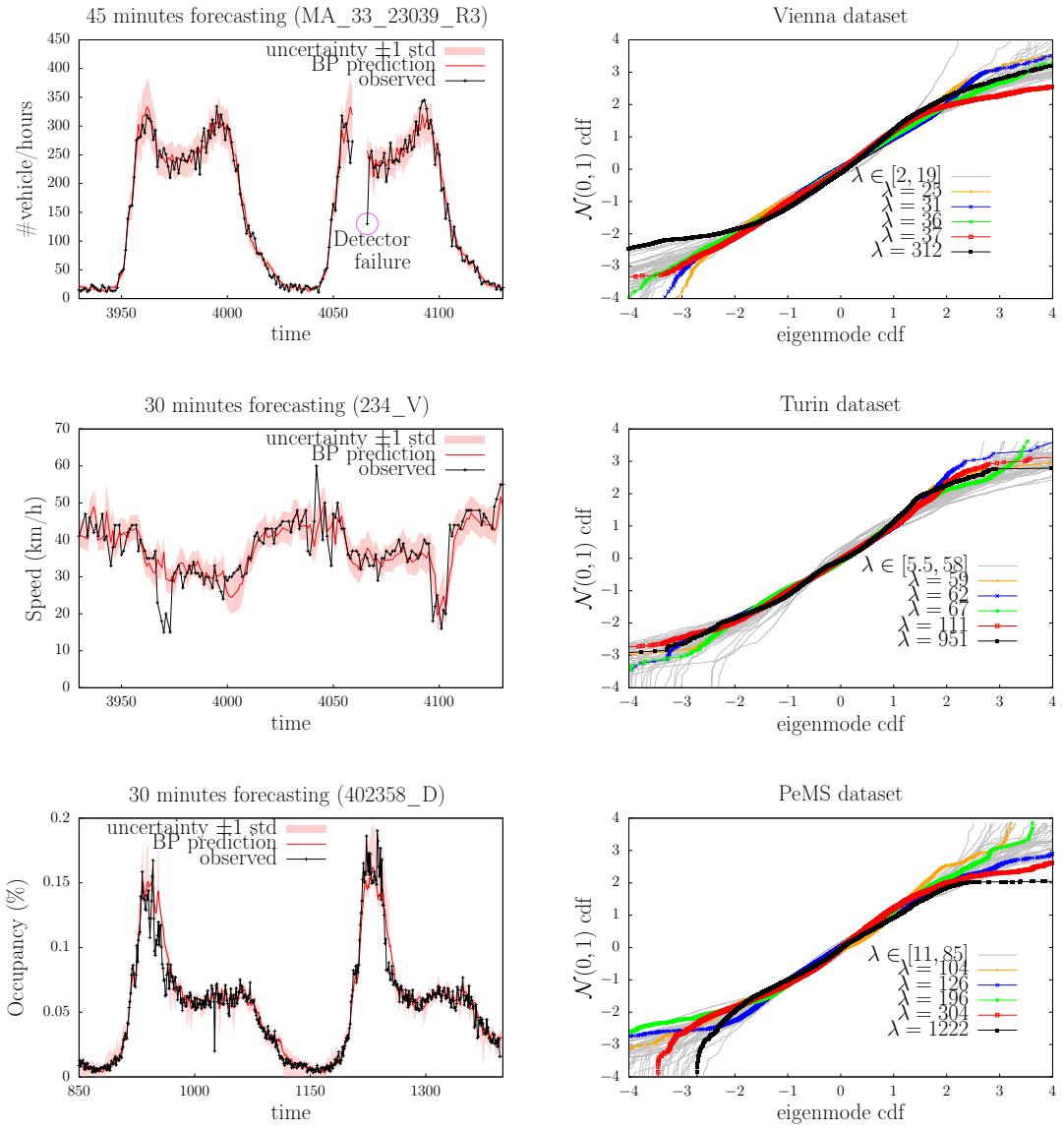


Figure 10: (left panel) Flow, speed and occupancy forecast, including uncertainty estimate delivered by GaBP, for a given CLOC respectively in Vienna, Turin and PeMS datasets. The confidence interval corresponds to 1 std in copula space, corresponding i.e. to 84% of confidence that the true value is 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)

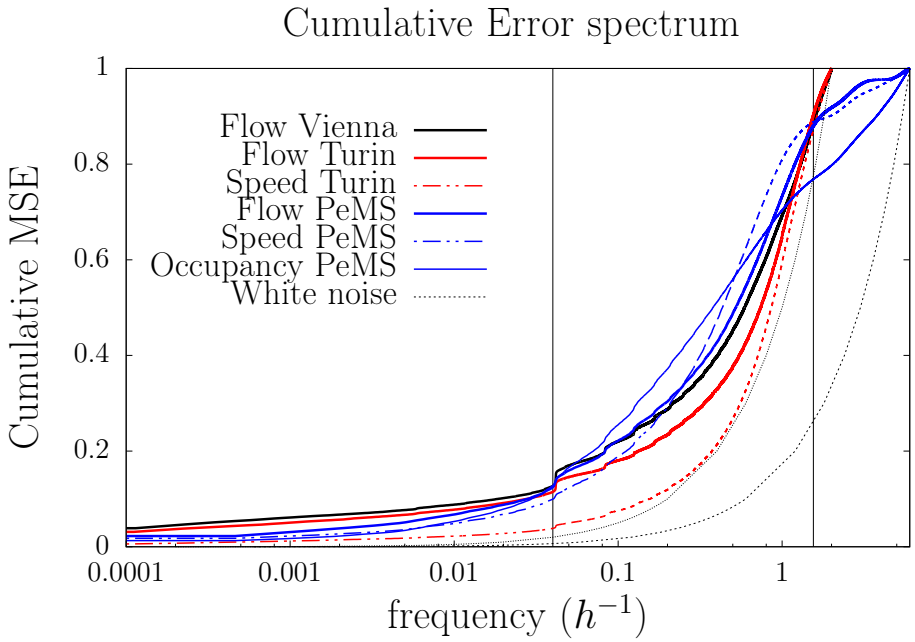


Figure 11: Cumulative power spectrum of the mean errors signal of the GaBP predictor, for the different datasets and type of variables. Vertical lines indicate one day and one hour time scales. Dotted lines indicate a white noise reference for Vienna, Turin and PeMS.

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 11 shows this for the different datasets and types of variables compares this to a white noise predictor, that is a predictor which would be able to predict the observed values with random decorrelated errors of same variance as the GaBP predictor error. 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 of time, 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 seems rather uniform beyond a point corresponding to $f = 1.55h^{-1}$, 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 effectively observe such a high frequency noise on most of the signals, particularly pronounced on the PeMS occupancy variables. We don't know whether something can be done about that or if this noise corresponds to natural and uncorrelated fluctuations of the traffic phenomena. A more careful analysis will tell, but clearly in our approach these contributions are interpreted as random noise by our model. Then 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 on Figure 11, this corresponds to 70 to 80% of the total error. It is in this region that we

can mostly pinpoint modelling 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 instead, the discrepancy increases drastically in this region, which confirms a global modelling 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 in the year). Also, the part of the spectrum corresponding to daily time scales indicates that the day profiles 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.

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 becomes 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 which 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 properly 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 to estimate 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 help of the soft belief setting mentioned in Section 2.4 might help us to improve the accuracy of the forecast by discarding automatically 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 with us their data.

References

[1] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John 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] Alireza Ermagun and David Levinson. Spatiotemporal traffic forecasting: review and proposed directions. *Transport Reviews*, 0(0):1–29, 2018.

[3] Marco Lippi, Matteo Bertini, and Paolo 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] Gaetano Fusco, Chiara Colombaroni, and Natalia 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] Haiyang Yu, Zhihai Wu, Shuqin Wang, Yunpeng Wang, and Xiaolei Ma. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, 17(7), 2017.

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

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

[10] Yanjie Duan, Yisheng Lv, Yu-Liang Liu, and Fei-Yue Wang. An efficient realization of deep learning for traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 72:168 – 181, 2016.

[11] Shun Kataoka, Muneki Yasuda, Cyril Furtlehner, and Kazuyuki Tanaka. Traffic data reconstruction based on markov random field modeling. *Inverse Problems*, 30(2):025003, 2014.

[12] Yusuke Hara, Junpei Suzuki, and Masao 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.

[13] 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.

[14] Victorin Martin, Jean-Marc Lasgouttes, and Cyril Furtlehner. Latent binary MRF for online reconstruction of large scale systems. *Annals of Mathematics and Artificial Intelligence*, pages 1–32, 2015.

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

[16] 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.

[17] 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.

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

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

[20] Yair Weiss and William T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Comput.*, 13(10):2173–2200, 2001.

[21] 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.

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

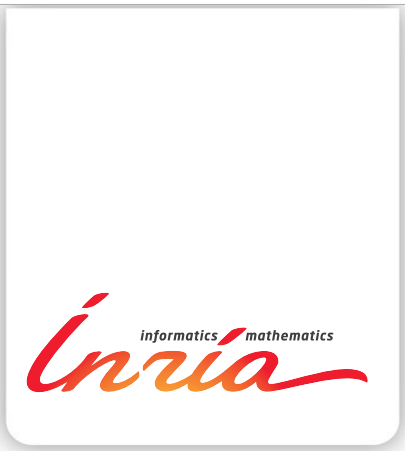
[23] Katya Scheinberg and Irina Rish. Learning sparse Gaussian Markov networks using a greedy coordinate ascent approach. In *ECML-PKDD*, 2010.

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

[25] TP Speed and HT Kiiveri. Gaussian Markov distributions over finite graphs. *The Annals of Statistics*, 14(1):138–150, 1986.

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

[27] 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