Probabilistic Graph Recurrent Imputation

GDL 2022
Group 04
Project 13

Pratyai Mazumder, Wonbin Song, Zechen Wu

Abstract

We have used the GRIN architecture introduced by Cini et al. [1] to capture some of the uncertainties inherent to an imputation task [2] on a spatio-temporal graph. More specifically, we proposed that during imputation, instead of estimating a mere point that maximizes the likelihood of observing the imputed data exactly at that point, we could estimate an interval that achieves a certain likelihood of observing the imputed data within that interval. Then, we adjusted the GRIN-based imputation model for applying simple quantile regression techniques [3] aiming different quantile-points, and constructed an interval of a desired significance level based on those quantile-points. Even such a simplistic interval estimate will enable better statistical decision making for a class problems where it is important to learn the distribution of the imputed data.

1 Introduction

Dealing with missing data is an important aspect of any practical statistical analysis or learning method that works with real-world data. Deep learning on spatio-temporal graphs that model physical or social systems are no different either. A typical and effective approach is *imputation*, which is to fill in with proxy data from the same underlying generating distribution as the dataset[2, p.25]. This allows various standard methods to operate on the dataset without completely discarding the data-points with missing values.

For imputing on spatio-temporal graphs, the GRIN architecture has been developed[1]. GRIN has been concerned with point-estimates from single imputation methods[2, p.68] so far. However, these point-estimates do not account for the imputation uncertainty, and can distort the statistical properties of the learned model (e.g. underestimation of standard errors, too narrow confidence intervals etc.)[2, p.81]. This effect can be combated by using methods like resampling methods (e.g. Bootstrap or Jackknife standard errors), multiple imputations etc.

This report develops an imputation method based on quantile regression using the GRIN architecture for *missing at random* (MAR)[2, p.14] spatio-temporal data – in particular, its implementation and application on a benchmark dataset. The current implementation of the proposed imputation method builds on TSL[4], and focuses on a single, but probabilistic imputation with some uncertainty. It lends itself to multiple imputation quite easily too.

2 Related works

2.1 Time-series Imputation

There is a large literature on the imputation of time series. Recently, many researchers proposed deep learning approaches to impute missing values in time series. GRU-D[5] is a GRU[6]-based model which imputes missing values in multivariate time series by incorporating two representations of missing patterns inside the GRU architecture. M-RNN[7] imputes missing values by training both an interpolation block and an imputation block simultaneously. BRITS[8] is a bidirectional RNN-type model which takes account for the correlation between missing variables to imputes missing values in multivariate time series. GAIN[9] is a generative model for missing value imputation which adopt the GAN[10] architecture.

2.2 Quantile Regression and Uncertainty

Although deep learning method show its power and potential in modeling spatio-temporal data, but the most approaches are focused on conditional expectations of the output variables being modeled. A multi-output multi-quantile deep learning approach[11] was invented to provide a more comprehensive perspective of the predictive density in spatio-temporal problems

Aleatoric uncertainty[12] describes the variance of the conditional distribution of the target variable given features. This type of uncertainty arises due to unmeasured variables or measurement errors, and cannot be reduced by collecting more data from the same distribution. To estimate aleatoric uncertainty, a mothod called ”simultaneous quantile regression”[12] was proposed , which use a loss function to learn all the conditional quantiles of a given target variable. These quantiles lead to well-calibrated prediction intervals.

In machine Learning, a random forest classifier[13] can be trained to predict missing values. This method is a very effective imputation method. but for dealing with probability, Bayesian inference has been exploited, DNN trained using dropout is interpreted as a Bayesian method. A new method converts a neural network trained using dropout to the corresponding Bayesian neural network with variance propagation[14].

3 Preliminaries

Please check out the original GRIN paper (Cini et al. [1]) for a detailed background on imputation on spatio-temporal graphs. We briefly reintroduce the relevant concepts here.

3.1 Sequences of Graphs

GRIN is concerned with sequences of weighted directed *homogeneous fixed* graphs $G_t = (\mathbf{X}_t, \mathbf{W}_t)$, where $\mathbf{X}_t$ are d-dimensional feature vectors per node, $\mathbf{W}_t$ are adjacency-matrix scalars. Since the graph is *fixed*, the graph topology, i.e. $\mathbf{W}_t$ does not change over time – although $\mathbf{X}_t$ does. And since it is homogeneous, all the nodes are of the same kind. Also, note that at each time step, *typically* all of the node features are expected to be available (unless its missing values – the cases we will need to impute). Essentially, graph sequences of such kind are represented as homogeneous multivariate time-series.

3.2 Multivariate Time-series Imputation

Given the multivariate time-series representation of a graph $G_t = (\mathbf{X}_t, \mathbf{W}_t)$, we can model the *missing-at-random* values with a binary mask $M_t \in \{0,1\}^{N \times d}$, where a $m_{i,j} = 0$ indicates a missing value, which masks the corresponding entries from the ground truth data $\tilde{\mathbf{X}}_t$ to produce the *observable* dataset $\mathbf{X}_t$. We also must change our training loss function for the time-interval $[t, t+T]$ accordingly (where we do have access to $\tilde{\mathbf{X}}_t$):

$$\mathcal{L}_{[t,t+T]}(\hat{\mathbf{X}}, \tilde{\mathbf{X}}, \overline{\mathbf{M}}) = \sum_{h=t}^{t+T} \sum_{i=1}^{N} \frac{\langle \overline{m}_{i,h}, \rho(\hat{x}_{i,t}, \tilde{x}_{i,t}) \rangle}{||\overline{m}_{i,h}||^2}$$

Where $\overline{\circ}$ operator indicates the logical complement, and $\rho(\hat{x}, x)$ is the element-wise loss function. In practice, $\tilde{\mathbf{X}}_t$ is never known, and different methods are applied to train the models based on the observable data $\mathbf{X}_t$.

3.3 Quantile Regression (QR)

Let X be a random variable with cumulative distribution function (CDF) $F(x) = P(X \le x)$. Then the τ^{th} quantile of random variable X is defined as

$$F^{-1}(\tau) = \inf\{x : F(x) \ge \tau\} \quad ; \tau \in (0,1)$$

Quantile regression minimizes the loss function[3]

$$\rho_\tau(x) = (\tau - \mathbb{1}_{x \le 0})x$$

To see this let $\hat{x}$ minimizes the expected loss function

$$\begin{aligned}\mathbb{E}[\rho_\tau(x - \hat{x})] = (1-\tau) \int_{-\infty}^{\hat{x}} (x - \hat{x}) dF(x) \\ - \tau \int_{\hat{x}}^{\infty} (x - \hat{x}) dF(x).\end{aligned}$$

Taking derivative with respect to $\hat{x}$, we have

$$\begin{aligned}0 &= (1-\tau) \int_{-\infty}^{\hat{x}} dF(x) - \tau \int_{\hat{x}}^{\infty} dF(x) \\ &= F(\hat{x}) - \tau\end{aligned}$$

that is, $\hat{x} = F^{-1}(\tau)$ minimizes the expected loss function.

The usual quantile regression model is

$$Y_i = Z_i^T \beta + \epsilon_i, \ \epsilon_i \text{ i.i.d. noise.}$$

In quantile regression problem we want β that minimizes $\mathbb{E}[\rho_\tau(Y_i - Z_i^T \beta)]$. The solution is found by

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \rho_\tau(Y_i - Z_i^T \beta).$$

3.4 Multiple Imputation

As mentioned earlier, imputing with point-estimates (mean, median etc.) may distort some important statistical property important for the consumer system of the imputed data[2, p.81] and multiple imputation is a typical way to deal with this. Multiple imputation (MI)[15] uses several suitable values which represent a probabilistic distribution to replace missing or insufficient values in the dataset. MI essentially imputes the dataset D times, each time drawing from the posterior predictive distribution of the target Y, to produce D completed datasets[2, p.96].

Approximating this posterior predictive distribution (PPD) is not trivial, and a simple approach taken in this work is described in section 4.1.

4 Methodology

4.1 Approximating the PPD for MI

Our approach is to estimate a narrow interval, instead of just a point value, in which it is sufficiently likely to find the true value. Once such an interval $[L, U]$ is estimated, the distribution can be approximated in different ways; for example:

- A Dirac delta at the center

$$pdf(y) = \delta\Big(y - \frac{L+U}{2}\Big)$$

- A piece-wise constant

$$pdf(x) = \begin{cases} \frac{1}{U-L} & ; L \le y \le U \\ 0 & ; otherwise \end{cases}$$

- A truncated normal

$$pdf(y) = \frac{1}{\sigma} \frac{\phi(\frac{y-\mu}{\sigma})}{\Phi(\frac{U-\mu}{\sigma}) - \Phi(\frac{L-\mu}{\sigma})}$$

where $\phi(\xi)$ is the PDF of the standard normal $\mathcal{N}(0,1)$ and $\Phi(\xi)$ is its CDF.

All these approximations will avoid generating outliers, while selecting a plausible value with a reasonable variance. All three were implemented in the course of this work and are demonstrated in section 6.

Alternative ideas to this interval estimation exist (e.g. estimating the parameters of an assumed distribution). However, they were not attempted in the scope of this report.

4.2 Estimating the interval bounds with QR

So, how do we estimate this *narrow interval of sufficient likelihood*? More precisely, the task is to find the narrowest possible interval which has a desired likelihood (e.g. significance level $\alpha = 0.05$) of containing the data. There are different ways to estimate a confidence interval, although finding the narrowest possible is usually quite difficult. The simplest way is to estimate the interval bounds $[F^{-1}(\frac{\alpha}{2}), F^{-1}(1 - \frac{\alpha}{2})]$, which still gives reasonably narrow interval for many typical scenarios.

Two approaches were considered for estimating the interval bounds:

- Learn separate models for the upper and the lower bounds of the interval given then target quantiles. Then build a predictor that uses these two models to estimate the interval and impute accordingly.
- Learn a single simultaneous quantile regression model[16] for any target quantile. Then use this model to predict the interval bounds and impute accordingly.

Current work only used the first approach, which is also the simplest.

4.3 Metrics for Interval Estimation : PICP and MPIW

To quantify what we mean by *reasonably narrow interval* with *sufficient likelihood*, the following two metrics[17] have been used.

Prediction Interval Coverage Probability (PICP)
This is the frequency of the true value actually being contained in the predicted interval. For a given interval size, a higher PICP is desirable. However, we also want the interval to be narrow, since a 100% PICP is easily achieved with arbitrarily wide intervals.

$$PICP = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{L_i \leq y_i \leq U_i}$$

Mean Prediction Interval Width (MPIW)
This is the mean width of all the predicted intervals, given a target PICP. Since we want to find the narrowest possible interval, minimizing MPIW is a goal in general.

$$MPIW = \frac{1}{n}\sum_{i=1}^{n}(U_i - L_i)$$

5 Implementation

5.1 Datasets

The METR-LA dataset[18] was used for the demonstration, which is already included in TSL[4]. The structure of the dataset and the configuration of its loader for our experiment is largely unchanged from the original GRIN paper[1]. To paraphrase – METR-LA contains 4 months of sensor readings from 207 detectors in the Los Angeles County Highway with a sampling rate of 5 minutes. Input sequences of 24 steps, corresponding to 2 hours of data, were used. A thresholded Gaussian kernel applied to geographic distances. A training-validation-testing split of 70%-10%-20% were used. Figure 1 shows some examples of marginal distributions of the dataset from an exploratory perspective.

However, while this setup was used for both learning the models for quantile predictions, and testing the final imputer, the demonstrated work in this report *does not* maintain a complete separation of test data for the final imputer and the training data for the QR. This was due to technical difficulties and time-constraint – not an ideal choice.

5.2 Hyperparameters

All the hyperparameters were largely unchanged from the GRIN paper[1, p15], and can be found in the file `imputation/config/grin.yaml`. The encoder-decoder network and the MLP both have 64 neurons, and the learnable embedding size was 8. The single-layered message passing operation was a diffusion convolution with a kernel size of 2 in the encoder. Dropout was not used.

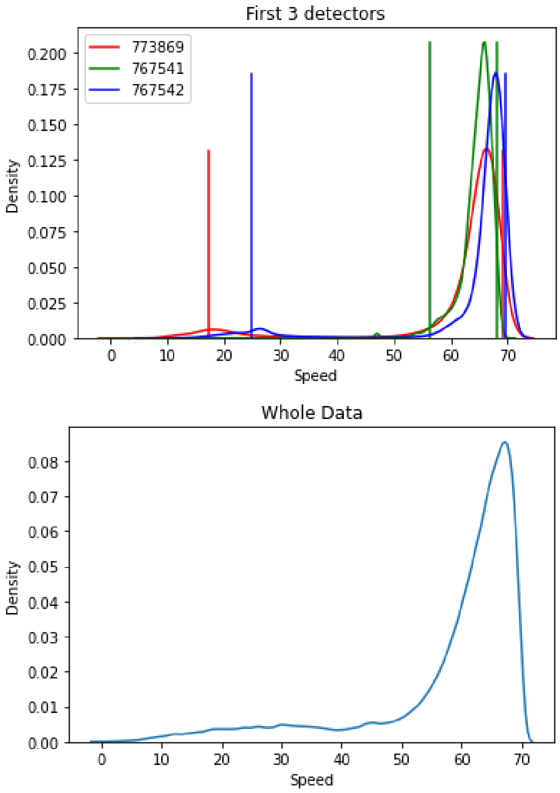

Figure 1. The distribution of the dataset (METR-LA). The vertical lines in the upper plot indicate 0.025 and 0.975 quantiles *marginal* distribution (i.e. for *all* observations on that node).

5.3 Experimental Setup

All relevant code and notebooks related to this work are to be found in the GitHub repository[19] for this project.

The implementation structure largely reuses TSL[4] repository as is, and only adds these additional libraries–

- `imputation/qloss.py` contains the metrics related to quantile regression and interval prediction.
- `imputation/qimputer.py` contains a wrapper class to impute with pretrained quantile model.
- `imputation/run_quantile_training.py` recycles the `run_imputation.py` script from TSL to train and store quantile models.
- `imputation/qpredictor_experiments.py` computes test metrics for the imputation with pretrained models.

Notably, the training and the test parts of the workflows are separated out, since the training takes a long time and the test metrics of the quantile metrics are not particularly relevant for the final imputation itself.

5.4 Computational Requirements

The training and testing was done mostly on Google Colab[20] platform on shared GPU. With the given hyperparameters, the training for each quantile model took roughly 160 epochs[*], 245 iterations per epoch, and roughly 1.4 seconds per iteration, totalling around 15 hours per model. 12 such models were trained to produce the results in section 6.

6 Results

Table 1. Error statistics of $1\times$ imputation for different target intervals for METR-LA dataset with *points missing at random*.

Methods	MAE	MRE	MAPE	PICP	MPIW
Target 68% prediction interval					
`Center`	1.9489	0.0337	0.0454	0.6048	2.1866
`Uniform`	2.1834	0.0378	0.0506		
`Normal`	2.1759	0.0377	0.0505		
Target 95% prediction interval					
`Center`	2.3885	0.0414	0.0535	0.9284	6.007
`Uniform`	3.2561	0.0564	0.0728		
`Normal`	3.2311	0.0560	0.0723		
Target 99.7% prediction interval					
`Center`	3.8741	0.0671	0.0796	0.9913	9.365
`Uniform`	5.3719	0.093	0.1141		
`Normal`	5.3203	0.0921	0.113		

Table 2. Error statistics of $1\times$ imputation for different target intervals for METR-LA dataset with *blocks missing at random*.

Methods	MAE	MRE	MAPE	PICP	MPIW
Target 68% prediction interval					
`Center`	2.1171	0.0367	0.0502	0.6156	1.6464
`Uniform`	2.3564	0.04081	0.0552		
`Normal`	2.3492	0.0407	0.0551		
Target 95% prediction interval					
`Center`	2.7266	0.0472	0.0635	0.9264	4.6182
`Uniform`	3.6134	0.0626	0.0823		
`Normal`	3.5895	0.0622	0.0818		
Target 99.7% prediction interval					
`Center`	5.0352	0.0872	0.1086	0.9915	7.3487
`Uniform`	6.5893	0.1141	0.1433		
`Normal`	6.5412	0.1133	0.1421		

We simulated our method in two different settings, block missing and point missing, as in the GRIN[1] paper. In each setting, we constructed $1-\alpha$ prediction intervals with $alpha = 0.32, 0.05, 0.003$. We performed three types of imputation for each interval: 1) Center, choose middle point of the interval, i.e., the mean of two quantiles; 2) Uniform, choose the value uniformly in the interval; 3) Truncated normal, choose the value normally in the interval. We used mean absolute error (MAE), mean relative error (MRE), and mean absolute percentage error (MAPE) as performance metrics for imputation accuracy, and prediction interval coverage probability (PICP) and mean prediction interval width (MPIW) as performance metric for prediction interval quality.

Table 1 and 2 summarize our experiment. Our best result of probabilistic imputation (from center of 68% prediction interval) achieved accuracy very close to that of GRIN's. As for PICP and MPIW the higher the PICP, the higher the MPIW as expected. Well calibrated prediction interval has PICP $\approx 1-\alpha$, and our results are slightly less than the target $1-\alpha$.

7 Discussion and conclusion

In this work, we developed a quantile regression based imputer using the GRIN[1] model that propagates reconstruction uncertainty through time and space. Our method constructs prediction intervals of imputed values, and perform imputation on the constructed interval. Our probabilistic imputation compare to the GRIN[1] model has less reconstruction accuracy, but it has better reconstruction accuracy than other state-of-arts baselines. As indicated in section 5.1 our work does not maintain a complete separation of train/test set, i.e., some of the data in training set are also included in the test set. If one can deal with this issue, the result would be more credible.

As a future work, one direction might be jointly estimate all the conditional quantiles using simultaneous quantile regression [16] and construct well-calibrated prediction interval.

8 Further work

References

[1] Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the g_ap_s: Multivariate time series imputation by graph neural networks. In *International Conference on Learning Representations*, 2021.

[2] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[3] Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4): 143–156, 2001.

[4] Andrea Cini and Ivan Marisca. Torch Spatiotemporal, 3 2022. URL `https://github.com/TorchSpatiotemporal/tsl`.

[5] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. Estimating missing data in temporal data

[*] Due to early stopping. Limit was set to 300 epochs.

streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490, 2018.

[8] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.

[9] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pages 5689–5698. PMLR, 2018.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[11] Filipe Rodrigues and Francisco C. Pereira. Beyond expectation: Deep joint mean and quantile regression for spatio-temporal problems, 2018. URL https://arxiv.org/abs/1808.08798.

[12] Natasha Tagasovska and David Lopez-Paz. Frequentist uncertainty estimates for deep learning, 11 2018.

[13] Adam Pantanowitz and Tshilidzi Marwala. Evaluating the impact of missing data imputation. In *Advanced Data Mining and Applications*, pages 577–586. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-03348-3_59. URL https://doi.org/10.1007%2F978-3-642-03348-3_59.

[14] Yuki Mae, Wataru Kumagai, and Takafumi Kanamori. Uncertainty propagation for dropout-based bayesian neural networks. *Neural Networks*, 144:394–406, 2021. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2021.09.005. URL https://www.sciencedirect.com/science/article/pii/S0893608021003555.

[15] Donald B Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.

[16] Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning, 2018. URL https://arxiv.org/abs/1811.00908.

[17] Ying Yin Ting and Jason Ansel. Better prediction intervals with neural networks, Jan 2020. URL https://web.archive.org/web/20200213165932/https://www.godaddy.com/engineering/2020/01/10/better-prediction-interval-with-neural-network/.

[18] 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.

[19] Pratyai Mazumder, Wonbin Song, and Zechen Wu. Probabilistic Graph Recurrent Imputation, 2022. URL https://github.com/pratyai/gdl-2022.

[20] Google Colab. URL https://colab.research.google.com/.