3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–7

The Gaussian Process Latent Autoregressive Model

Rui Xia RUI.XIA@U.NUS.EDU
Wessel Bruinsma WPB23@CAM.AC.UK
William Tebbutt WCT23@CAM.AC.UK
Richard E. Turner RET26@CAM.AC.UK
University of Cambridge

Abstract

Many real-world prediction problems involve modelling the dependencies between multiple different outputs across the input space. Multi-output Gaussian Processes (MOGP) are a particularly important approach to such problems. In this paper, we build on the Gaussian Process Autoregressive Regression (GPAR) model which is one of the best performing MOGP models, but which fails when observation noise is large, when there are missing data, and when non-Gaussian observation models are required. We extend the original GPAR model to handle these settings and provide a variational inference procedure similar to that used in deep Gaussian Processes which replaces the ad hoc denoising approximation used in the original work. We show that the new approach naturally handles noisy outputs, missing data and that it also enables the model to handle heterogeneous non-Gaussian observation models.

1. Introduction

With the growing prevalence of complex decision making systems, there is increasing need for learning systems that predict multiple outputs simultaneously. Multi-output problems appear in many different forms: they can differ in data types of the outputs or in the ways outputs depend on one another. Examples that involve mixed output data types include real-valued multi-target regression (Borchani et al., 2015), multi-label classification (Zhang and Zhou, 2013), and the heterogeneous case where a mix of continuous, categorical, or discrete variables are of interest (Moreno-Muñoz et al., 2018). Complex dependencies between outputs also appear in various ways: one output might depend quite simply on inputs but can depend on certain other outputs in a complex way. The sophisticated dependencies between these outputs require structured modelling. Multi-output GPs (MOGPs) are a powerful and popular approach to multi-output modelling. In this paper, we focus on one of the MOGPs that explicitly treats outputs as inputs, called the Gaussian Process Autoregressive Regression (GPAR), studied by Requeima et al. (2018). We generalise GPAR to deal with noisy or missing values in the outputs in a principled way and enable it to model non-Gaussian likelihoods and even heterogeneous data. We connect the proposed model to Deep Gaussian Process (DGPs) and leverage the approximate inference methods developed for DGPs (Salimbeni and Deisenroth, 2017).

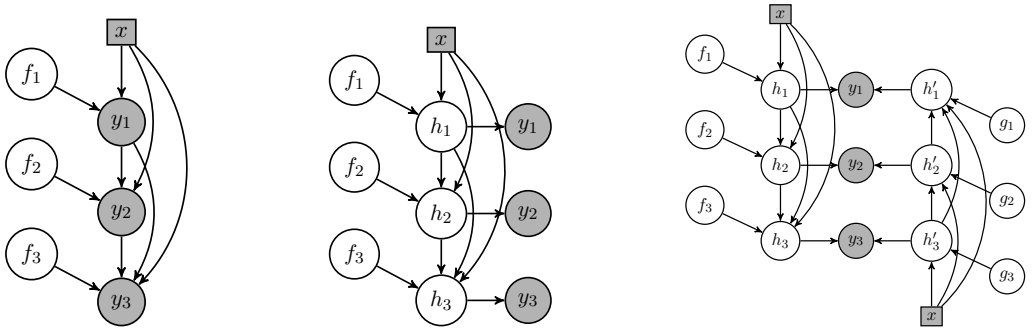
Main Contribution. First, we present the Gaussian Process Latent Autoregressive model (GPLAR), combining ideas from DGPs and MOGPs. Hidden variables are introduced corresponding to noiseless, unobserved but true latent function evaluations that require Bayesian inference. Second, we find that GPLAR can still perform poorly when there are missing values in the first few outputs. We propose a new version of GPLAR inspired by bi-directional Recurrent Neural Networks to address this limitation.

2. Gaussian Process Latent Autoregressive Model

2.1. The GPAR Model

In the multi-output scenario, assume $\mathbf{x}$ and $\mathbf{y} = \{\mathbf{y}_l\}_{l=1}^L$ are the training inputs and associated observations for L outputs. We assume all L outputs share the same input space. We utilize the product rule to decompose the joint distribution over all outputs into a set of univariate conditional distributions. In particular, GPAR factorizes the distribution of L outputs $y_{1:L}(\mathbf{x}) = (y_1(\mathbf{x}), \ldots, y_L(\mathbf{x}))$ as $p(y_{1:L}(\mathbf{x})) = p(y_1(\mathbf{x}))p(y_2(\mathbf{x})|y_1(\mathbf{x})) \ldots p(y_L(\mathbf{x})|y_{1:L-1}(\mathbf{x}))$, so $y_l(\mathbf{x})$ is generated from $y_{1:l-1}(\mathbf{x})$ according to some latent function f_m. GPAR models these latent functions $f_{1:L}$ with GPs, where their kernels $k_{1:L}$ can be linear, nonlinear, or composite. GPAR is a state-of-the-art MOGP model on small multi-output regression problems.

Deficiencies of GPAR. There are limitations in the above formulation of GPAR. A graphical model of three outputs is shown in Fig. 1(a), where observation y_1 is directly used as inputs to the functions f_2 and f_3. Noisy outputs from an earlier stage result in noisy inputs to a subsequent level. The original paper solved this by employing a denoising transformation: the posterior predictive mean of preceding outputs are used as inputs instead of observed values. Furthermore, the inference and learning procedures provided in the original paper are only valid for *closed-downwards* observations, i.e., for every observation $y_{ln} = y_l(\mathbf{x}_n)$, there are also observations $y_{(1:l-1)n}$. For observations that are not closed downwards, the authors propose to impute the necessary observations with posterior predictive means. We will show in the experiments section that this imputation method and GPAR's layer-by-layer fitting procedure performs poorly on closed-upwards observations.



(a) GPAR (b) GPLAR (c) Bi-GPLAR

Figure 1: Graphical models of (a) GPAR, (b) GPLAR, and (c) Bi-GPLAR. Observed variables $y_{1:3}$ are shaded. $f_{1:3}$ denote latent function mappings.

2.2. The GPLAR Model

A more principled approach is to perform Bayesian inference. Instead of directly working on observations, we introduce latent variables $\mathbf{h}_{1:3}$ for each output, graphically shown in Fig. 1(b). Unfortunately, approximate inference is now required to deal with these variables, which we develop next. The same approximate inference procedure will also enable non-Gaussian likelihoods, such as for classification or non-negative data. We call this model the

Gaussian Process Latent Autoregressive (GPLAR) model. We describe the probabilistic model for a GPLAR model with L outputs. The main difference with DGPs is that all previous hidden variables are propagated to the next layer. Like GPAR, the latent functions in GPLAR are modelled with GPs: $p(f_l|\theta_l) = \mathcal{GP}(f_l; \mathbf{m}_l, \mathbf{K}_l)$, for $l = 1, \ldots, L$. These functions are then connected in the following way:

$$p(\mathbf{h}_l|f_l, \mathbf{X}, \mathbf{h}_{1:l-1}, \sigma^2) = \mathcal{N}(\mathbf{h}_l; f_l(\mathbf{X}, \mathbf{h}_{1:l-1}), \sigma_l^2 \mathbf{I}_n), \quad p(\mathbf{y}_l|\mathbf{h}_l) = \mathcal{N}(\mathbf{y}_l; \mathbf{h}_l, \sigma_{y_l}^2 \mathbf{I}_n).$$

Approximate Inference for GPLAR. The posterior distributions over the latent function mappings, $f_{1:L}$, as well as over the intermediate hidden variables $\mathbf{h}_{1:L-1}$ are of interest. Existing work by Salimbeni and Deisenroth (2017); Bui et al. (2016) for approximating these distributions considers variational approximations (Titsias, 2009) over the latent functions but retains the prior conditionals $p(h_{ln}|f_l, h_{(l-1)n})$. Using the same idea, we introduce inducing points to every layer (w.r.t. output dimensions) of GPLAR. The approximate posterior and joint distribution (written in terms of the function values at inducing points $\mathbf{u}$) of a three output GPLAR are as follows:

$$\begin{aligned} q(f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3) = & \, p(f_{1\neq\mathbf{u}_1}|\mathbf{u}_1) p(f_{2\neq\mathbf{u}_2}|\mathbf{u}_2) p(f_{3\neq\mathbf{u}_3}|\mathbf{u}_3) q(\mathbf{u}_1) q(\mathbf{u}_2) q(\mathbf{u}_3) \times \\ & \prod_n \Big[p(h_{1n}|f_1, \mathbf{x}_n) p(h_{2n}|f_2, h_{1n}, \mathbf{x}_n) p(h_{3n}|f_3, h_{2n}, h_{1n}, \mathbf{x}_n) \Big], \quad (1) \\ p(\mathbf{y}, f_1, f_2, f_3, \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3) = & \, p(f_{1\neq\mathbf{u}_1}|\mathbf{u}_1) p(f_{2\neq\mathbf{u}_2}|\mathbf{u}_2) p(f_{3\neq\mathbf{u}_3}|\mathbf{u}_3) p(\mathbf{u}_1) p(\mathbf{u}_2) p(\mathbf{u}_3) \times \\ & \prod_n \Big[p(h_{1n}|f_1, \mathbf{x}_n) p(h_{2n}|f_2, h_{1n}, \mathbf{x}_n) p(h_{3n}|f_3, h_{2n}, h_{1n}, \mathbf{x}_n) \times \\ & \qquad p(y_{1n}|h_{1n}) p(y_{2n}|h_{2n}) p(y_{3n}|h_{3n}) \Big]. \quad (2) \end{aligned}$$

Evidence Lower Bound. By applying the Jensen's inequality, we get a lower bound for the log-marginal likelihood (ELBO):

$$\mathcal{L}_{ELBO} = E_q \left[\log \frac{p(\mathbf{y}, f_{1:3}, \mathbf{h}_{1:3})}{q(f_{1:3}, \mathbf{h}_{1:3})} \right] = -\sum_{l=1}^{3} KL\left[q(\mathbf{u}_l) \| p(\mathbf{u}_l)\right] + \sum_{l,n} E_q \left[\log p(y_{ln}|h_{ln})\right]. \quad (3)$$

The difference between the exact log-marginal likelihood and the ELBO is equal to the KL divergence between the approximate posterior (Eq. 1) and the true one (Eq. 2). Maximizing the ELBO w.r.t. the variational parameters and the hyperparameters, we simultaneously obtain approximations to the log-marginal likelihood and the posterior. The second term in Eq. 3 decomposes over across the training instances and output dimensions l. This term can be rewritten as $\sum_{l,n} E_q \left[\log p(y_{ln}|h_{ln})\right] = \sum_{l,n} \int q(h_{ln}) \log p(y_{ln}|h_{ln}) dh_{ln}$, where $q(h_{ln}) = \int q(h_{ln}|h_{(1:l-1)n}, \mathbf{x}_n) \ldots q(h_{1n}|\mathbf{x}_n) dh_{1n} \ldots dh_{(l-1)n}$. Positing a Gaussian form for each variational distribution, $q(\mathbf{u}_l) = \mathcal{N}(\mathbf{u}_l; \mathbf{m}_l, \mathbf{S}_l)$, we notice that the latent function, f_l, can be analytically marginalized out at each layer: $q(h_{ln}|h_{(1:l-1)n}, \mathbf{x}_n) = \mathcal{N}(\mu_{h_l|h_{1:l-1}}(\hat{\mathbf{x}}_{ln}), \sigma^2_{h_l|h_{1:l-1}}(\hat{\mathbf{x}}_{ln}))$ where,

$$\begin{aligned} \mu_{h_l|h_{1:l-1}}(\hat{\mathbf{x}}_{ln}) &= \mathbf{k}_l(\hat{\mathbf{x}}_{ln}, \mathbf{Z}_l) \mathbf{K}^{-1}_{\mathbf{u}_l\mathbf{u}_l} \mathbf{m}_l, \\ \sigma^2_{h_l|h_{1:l-1}}(\hat{\mathbf{x}}_{ln}) &= k_l(\hat{\mathbf{x}}_{ln}, \hat{\mathbf{x}}_{ln}) - \mathbf{k}_l(\hat{\mathbf{x}}_{ln}, \mathbf{Z}_l) \mathbf{K}^{-1}_{\mathbf{u}_l\mathbf{u}_l} (\mathbf{K}_{\mathbf{u}_l\mathbf{u}_l} - \mathbf{S}_l) \mathbf{K}^{-1}_{\mathbf{u}_l\mathbf{u}_l} \mathbf{k}_l(\mathbf{Z}_l, \hat{\mathbf{x}}_{ln}) + \sigma_l^2 \end{aligned}$$

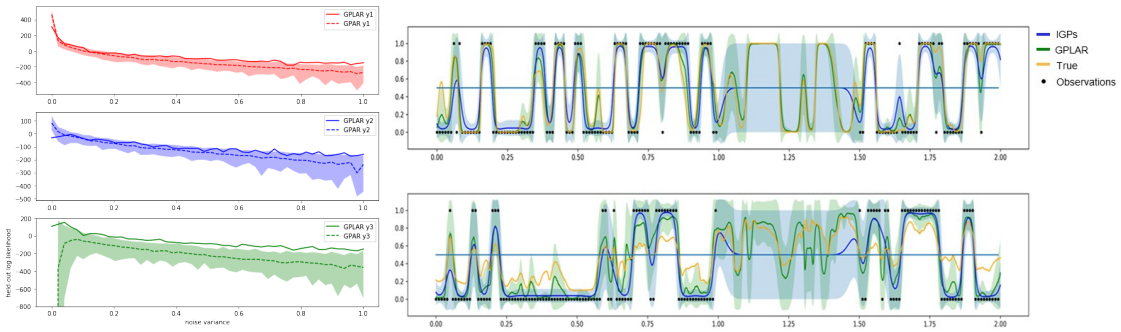
with $\hat{\mathbf{x}}_{ln} = (\mathbf{x}_n, h_{(1:l-1)n})$ is concatenation of the input and previous hidden variables and $\mathbf{Z}_l$ is the location of inducing points at each layer. Notice that for $q(h_{1n}|\mathbf{x}_n)$, the distribution does not marginalise out previous hidden layers, and is therefore simply a Gaussian predictive distribution. However, for $q(h_{2n}|\mathbf{x}_n) = \int_{h_{1n}} q(h_{2n}|h_{1n}, \mathbf{x}_n) q(h_{1n}|\mathbf{x}_n)$, the resulting $q(h_{2n})$ is a complicated infinite mixture of Gaussian densities (Bui, 2018), which can be multi-modal or heavy-tailed. To sample from this, we use a nested simple Monte Carlo method (Salimbeni and Deisenroth, 2017). When further propagating h_{2n} and h_{1n} to the posterior of h_{3n} in GLPAR, samples are drawn from a uniformly weighted mixture of Gaussian: $q(h_{3n}|\mathbf{x}_n) \approx \frac{1}{R}\sum_r q(h_{3n}|h_{2nr}, h_{1nr}, \mathbf{x}_n)$, where $h_{1nr} \sim q(h_{1n}|\mathbf{x}_n), h_{2nr} \sim q(h_{2n}|h_{1nr}, \mathbf{x}_n)$. To obtain low variance gradients, we apply the reparametrisation trick (Kingma and Welling, 2013) to recursively draw samples $h_{lnr} \sim q(h_{ln}|h_{(1:l-1)n}, \mathbf{x}_n)$ as $h_{lnr} = \mu_{h_l|h_{1:l-1}}(\hat{\mathbf{x}}_{ln}) + \epsilon_{lnr} \cdot \sigma^2_{h_l|h_{1:l-1}}(\hat{\mathbf{x}}_{ln})$, $\epsilon_{lnr} \sim \mathcal{N}(0, 1)$. When using a non-Gaussian likelihood, $\log p(y_{ln}|h_{ln})$, additional approximations such as another step of simple Monte Carlo sampling are required.

Treatment of Inducing Inputs. Treatment of the inducing inputs for the first layer is standard. However, for higher layers, choosing the locations for inducing points is less straightforward. These locations also require values for the corresponding previous function values, which should be consistent with the inducing points at those previous layers. Consequently, free optimization of inducing inputs at each layer is no longer appropriate. To remedy this, we take the mean of $q(\mathbf{u}_l)$ as inducing inputs for the next layer $l+1$. The resulting locations for the inducing inputs at each layer l are then $\begin{bmatrix}\mathbf{Z} & \mathbf{m}_1 & \dots & \mathbf{m}_{l-1}\end{bmatrix}$, where $\mathbf{m}_l$ denotes the mean of each variational distribution. In this setup, the inducing inputs are "automatically" optimized since they are variational parameters themselves. Please refer to supplementary material (Xia et al.) for more details.

Bi-directional GPLAR. In its current form, GPLAR has difficulty dealing with many outputs. Although the end-to-end training of all layers fits all layers simultaneously, unlike GPAR, which fits the layers in a greedy fashion, when the outputs dimension becomes large, back-propagation through the autoregressive structure becomes difficult. Moreover for real-world data, dependencies between two outputs is often asymmetric. Hence, an incorrect ordering of the outputs in GPLAR would model dependencies in the wrong directions. To alleviate this, we take inspiration from the bi-directional RNN model. The basic idea is to split each hidden state into two in order to model the forward and the backward direction separately, both of which are connected to the same output. Inspired by this idea, we run another GPLAR model in reverse; the graphical model is shown in Fig. 1(*c*). The hidden variables from both directions at each layer are aggregated and in the case of a regression problem, supplemented with noise to produce the observations. We will demonstrate that this structure can produce a better predictive mean and better calibrated uncertainty estimates.

3. Experiments

Synthetic Data Experiments. We first compare the ability of GPAR and GPLAR to model synthetic data generated from a multi-output GP where dependencies between outputs are nonlinear (see supplementary for full details). We calculate the held-out log-



(*a*) HLL vs noise (*b*) Heterogenous

Figure 2: (a) One trial for GPLAR is denoted by solid line. GPAR is unstable and 95% CI over HLL is indicated by the shaded area and the median is depicted by dashed lines. (b) GPLAR and IGPs predictions on the second labeling task. Black dots are observations. Synthetic GP kernels between outputs are (Up:) Linear, (Down:) Nonlinear.

likelihood (HLL) for a range of values for the variance of noise. It is observed from Fig. 2(*a*) that the higher outputs of GPAR are unstable, suggesting that noisy outputs from previous layers harms predictions for the next output. In contrast, GPLAR's HLL is observed to always overlap or locate higher than the 95% confidence upper bounds of GPAR. In the third output of GPAR, the HLL is extremely negative when the noise variance is close to zero. It turns out that, in this case, the predictive variances for the third output are all near zero. In comparison, the predictive variances of GPLAR are at an appropriate level. These results indicate that GPLAR can better handle noisy observations, and is more robust to under-fitting and over-fitting. Next, we extend GPLAR to non-Gaussian likelihoods and heterogeneous outputs. Data are drawn from 4 synthetic GPs. The last two outputs are converted to binary outputs using the sigmoid function; labels are then generated by drawing from the corresponding Bernoulli distributions. It is observed from Fig. 2(*b*) that information learned from previous tasks helps GPLAR predict the second labeling task. The predictive mean nearly recovers the true underlying process, and the uncertainty is greatly reduced. As expected, independent GPs (IGPs) fail to capture the dependencies and revert to the prior distribution in areas without observations.

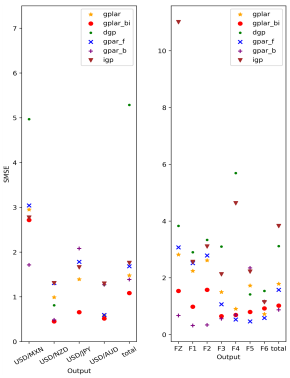
Real-World Data Experiments. In this section, we evaluate GPLAR and bi-GPLAR and compare to other models on two standard datasets where GPAR has been shown to be the state-of-the-art. One is the electroencephalogram (EEG) dataset[1], consisting of one second of measurements from 7 electrodes mounted on a patient's scalp. The second dataset is the exchange rates dataset[2], consisting of one year of exchange rates w.r.t. USD of ten international currencies and three metals in one year. The task is to predict missing values in some of the outputs given all other observations. The results are presented in Table. 3(*a*), and refer to supplementary materials for corresponding figures. For EEG, it is observed that predictions of GPAR over $F1$ are over-confident which leads to large HLL, while the uncertainty over F1 from GPLAR is well-calibrated. As for the exchange rates, although only the SMSE of GPLAR over "USD/AUD" is significantly lower than that of GPAR,

1. The EEG datset is available at https://archive.ics.uci.edu/ml/datasets/eeg+database.
2. The exchange rates dataset is available at http://fx.sauder.ubc.ca.

observe from figures that GPLAR gives predictions with reasonable uncertainty even outside the missing area, while GPAR produces overconfident predictions with a lot more wigglings. We further compare the performance of bi-GPLAR to the that of IGPs, GPAR in single direction, and DGPs on the EEG/exchange-rate dataset averaged over 5 patients/years respectively. The results are shown in Fig. 3(b). Since the EEG datasets contains almost noiseless measurements, GPAR in the forward direction performs better than GPLAR for closed-downwards observations. However, GPAR performs significantly worse on closed-upwards observations. Bi-directional GPLAR gives a performance between GPARs in the two directions but GPAR does take more time. Hence, in situations when noise is not dominant, one should use GPLAR with a natural ordering of outputs. For the exchange rate, the observations are much noisier and sometimes contain severe outliers. Bi-GPLAR gives the best results for all four outputs except for "USD/MXN", when correlations are hard to find for all models. Bi-GPLAR also gives the best average performance over all datasets with closed-upwards or closed-downwards observations, and generally performs better than DGPs and IGPs. Although DGPs model correlation between outputs, the experiments show that DGPs struggle to leverage knowledge of observed outputs for predicting missing outputs. GPLAR can also be applied to non-time-series data. We refer the reader to supplementary materials for experiments on more real-world data including radar image features and latitude-longitude spatial points as input.

Output	SMSE		HLL	
EEG	GPAR	GPLAR	GPAR	GPLAR
FZ	0.1340	0.1273	-135.7	-141.3
F1	0.3285	0.3130	**-663.1**	**-183.1**
F2	0.1536	0.1317	-132.4	-136.6
ER	GPAR	GPLAR	GPAR	GPLAR
USD/CAD	0.0215	0.0439	148.60	153.95
USD/JPY	0.0170	0.0234	843.18	860.95
USD/AUD	**0.2089**	**0.0685**	523.97	464.58

(a) EEG & Exchange Rates



(b) Overall

Figure 3: (a) GPAR vs GPLAR for the EEG and exchange datasets (b) SMSE over missing values of IGPs, GPAR in forward/backward direction, DGPs, GPLAR, and bi-GPLAR. "FZ,F1,F2" and "MXN,NZD" are **closedupwards**, "F5,F6" and "JPY, AUD" are **closed-downwards**, "F3,F4" are **neither**. The "total" column denotes average performance.

4. Conclusion

We have introduced GPLAR, an extension of GPAR that deals with noisy outputs using a fully Bayesian approach, enabling the resulting model to work with non-Gaussian or even heterogeneous likelihoods. We further extended the GPLAR model to a bi-directional version.

References

Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.

Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep gaussian processes for regression using approximate expectation propagation. In *International conference on machine learning*, pages 1472–1481, 2016.

Thang Duc Bui. *Efficient Deterministic Approximate Bayesian Inference for Gaussian Process models*. PhD thesis, University of Cambridge, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Pablo Moreno-Muñoz, Antonio Artés, and Mauricio Alvarez. Heterogeneous multi-output gaussian process prediction. In *Advances in neural information processing systems*, pages 6711–6720, 2018.

James Requeima, Will Tebbutt, Wessel Bruinsma, and Richard E Turner. The gaussian process autoregressive regression model (gpar). *arXiv preprint arXiv:1802.07182*, 2018.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599, 2017.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

Rui Xia, Wessel Bruinsma, Will Tebbutt, and Richard E Turner. Supplementary file for the gaussian process latentautoregressive model. https://github.com/XiaRui1996/GPLAR/blob/master/supp.pdf.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.