Visualizing Intrinsic Space for Spatial Data via Input Regularized Gaussian Process Latent Variable Models

Tomoharu Iwata and Naonori Ueda
NTT Communication Science Laboratories, Kyoto, Japan

Abstract— *We propose the input-regularized Gaussian process latent variable model for visualizing a latent intrinsic input space that improves interpolation performance in regression tasks. The proposed model assumes that a latent location is associated with each observed input location, and the covariance function is determined by distance between the latent locations. The latent locations are estimated so that the output covariance of the given data is appropriately captured by the latent locations while preserving the neighbor relationships between the observed input space and the latent space by input regularization. When the input regularization is omitted, the proposed model reduces to the Gaussian process latent variable model. When the input regularization is strong enough to perfectly preserve the neighbor relationships, the proposed model becomes Gaussian process regression. The degree of the regularization is controlled by a hyperparameter, which can be automatically selected by cross-validation using the given data. We demonstrate the effectiveness of the proposed model with real-world spatial data sets in terms of interpolation performance of multiple output values.*

Keywords: Gaussian processes, latent variable models, visualization, spatial data

1. Introduction

Analyzing spatial data is an important task in a wide variety of fields such as geology, ecology, climatology, sociology and urban planning. Gaussian process regression [1], or which is known as Kriging [2] in geostatistics, is a representative method for analyzing and interpolating spatial data. In Gaussian process regression, a covariance function plays a crucial role to define its behavior. With spatial data, kernels that solely depend on distance between two locations, e.g. Gaussian kernels, are usually used for covariance functions, since closely located points are assumed to have similar output values. However, some closely located locations can have different output values, and distant locations can have correlated output values. For example, weather at coastal area would be different from inland area in the same city, people's cultural behavior in two cities divided by a river would be dissimilar, and seismic activity would be related in distant areas when they share a fault line. In other words, the observed input space would be different from its intrinsic space that reflects relationships among locations. Revealing the intrinsic space is beneficial to understand the given spatial data as well as to improve interpolation performance.

In this paper, we propose the input-regularized Gaussian process latent variable model (IGPLVM) that visualizes the latent intrinsic space. The proposed model assumes that each observed input location has its own latent location, and the covariance function is determined by distance between the latent locations. Since neighbor relationships in the latent space should be similar to those in the observed input space as long as the output covariance of the given data is captured by the latent locations, latent locations are regularized to preserve the neighbor relationships. When the regularization is omitted, the proposed model becomes the Gaussian process latent variable model (GPLVM) [3], [4], which is an unsupervised method that learns latent locations using output values without input information. When the regularization is strong enough to perfectly preserve the neighbor relationships, it corresponds to Gaussian process regression, where input locations are assumed to be noise free. The proposed model adaptively uses input information so that output variables are modeled properly.

The proposed model improves interpolation performance by flexibly defining the covariance function by adjusting latent locations. Since the proposed model learns a common latent intrinsic space shared by multiple output variables, it can be used for multi-task learning. The latent space learned by using output values in a task helps to interpolate output values in another related task, even if data are too sparse to learn latent locations with a single task. Note that, although we motivate the proposed model for analyzing spatial data, the proposed model is applicable to other data for any regression problems where input and output values are contained, such as time-series, spatio-temporal, high dimensional input, and multiple output data.

The remainder of this paper is organized as follows. In Section 2, we present our task and introduce Gaussian process regression, on which the proposed model is based. In Section 3, we formulate the proposed model and provide its learning procedures. We review related work in Section 4. In Section 5, the effectiveness of the proposed method is demonstrated by experiments with real-world spatial data sets in terms of interpolation performance of multiple output values. Finally, we present concluding remarks and a discussion of future work in Section 6.

2. Preliminaries

Suppose that we have a set of input and output instances, $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, where $\mathbf{x}_n \in \mathbb{R}^L$ and $\mathbf{y}_n \in \mathbb{R}^D$. For example, in a case of spatial climate data, $\mathbf{x}_n$ is th nth location vector such as latitude and longitude, and $\mathbf{y}_n$ is an observed weather condition vector at the location such as temperature, humidity and precipitation.

With standard regression methods, an output $\mathbf{y}_n$ is assumed to be generated by mapping input $\mathbf{x}_n$ using nonlinear functions,

$$y_{nd} = f_d(\mathbf{x}_n) + \epsilon, \tag{1}$$

where $f_d(\cdot)$ is the nonlinear function for the dth feature, and $\epsilon \sim \mathcal{N}(0, \beta^{-1})$ is a Gaussian noise. Gaussian process regression uses a Gaussian process for a prior distribution of the nonlinear function,

$$f_d(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{2}$$

where $m(\mathbf{x})$ is a mean function, which is often set to zero, and kernel function $k(\mathbf{x}, \mathbf{x}')$ specifies the covariance of outputs between two locations as follows,

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f_d(\mathbf{x}) - m(\mathbf{x}))(f_d(\mathbf{x}') - m(\mathbf{x}'))]. \tag{3}$$

Since the kernel determines the behavior of the nonlinear functions, it is important to use an appropriate kernel for the given data. For real-valued input data, such as time-series and spatial data, kernels that are negatively correlated to distance between two locations $\parallel \mathbf{x} - \mathbf{x}' \parallel$ are often used, such as Gaussian kernels. However, output values can be different even if two locations are close together, and they can be similar even if two locations are far apart.

3. Input-regularized Gaussian Process Latent Variable Models

We propose the input-regularized Gaussian process latent variable model (IGPLVM), which is a method to obtain kernels that appropriately capture the output covariance between inputs by distorting the input space in the Gaussian process regression framework. The distorted input space reveals intrinsic characteristics of the input space.

The proposed model assumes that an output $\mathbf{y}_n$ is generated from its latent intrinsic location $\mathbf{z}_n \in \mathbb{R}^K$, instead of input location $\mathbf{x}_n$, as follows,

$$y_{nd} = f_d(\mathbf{z}_n) + \epsilon. \tag{4}$$

The dimensionality of the latent space K can be different from that of the input space L. Then, the probability of the output observations $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_N)^\top$ given the latent locations $\mathbf{Z} = (\mathbf{z}_1, \cdots, \mathbf{z}_N)^\top$, by integrating out the nonlinear functions, is given by

$$p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\theta}) = (2\pi)^{-\frac{DN}{2}} |\mathbf{K}|^{-\frac{D}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{Y}^\top \mathbf{K}^{-1} \mathbf{Y})\right), \tag{5}$$

where $\mathbf{K}$ is the $N \times N$ covariance matrix defined by the kernel function $k(\mathbf{z}_n, \mathbf{z}_{n'})$, and $\boldsymbol{\theta}$ is the kernel hyperparameter vector. In this paper, we use a Gaussian kernel with an additive noise term,

$$k(\mathbf{z}_n, \mathbf{z}_{n'}) = \alpha \exp\left(-\frac{1}{2\ell^2} \parallel \mathbf{z}_n - \mathbf{z}_{n'} \parallel^2\right) + \delta_{nm}\beta^{-1}, \tag{6}$$

where $\delta_{nm} = 1$ if $n = m$ and $\delta_{nm} = 0$ otherwise, and $\boldsymbol{\theta} = (\alpha, \ell, \beta)$ are the kernel parameters. Note that the GPLVM [3] finds latent locations $\mathbf{Z}$ that minimizes the negative log likelihood of (5),

$$E_{\mathbf{Y}} = -\log p(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\theta}), \tag{7}$$

where input information $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_N)^\top$ is not used.

We assume that neighbor relationships in the latent space should be similar to those in the input space as long as the output covariance that is specified by the latent space is appropriate to the given data. The proposed method models the neighbor relationships by defining a probability of being selected as neighbors as defined in stochastic neighbor embedding [5], [6]. In the latent space, the probability that n selects n' as its neighbors is given by

$$p(n'|n, \mathbf{Z}) = \frac{\exp(-\frac{1}{2} \parallel \mathbf{z}_n - \mathbf{z}_{n'} \parallel^2)}{\sum_{n'' \neq n} \exp(-\frac{1}{2} \parallel \mathbf{z}_n - \mathbf{z}_{n''} \parallel^2)}, \tag{8}$$

where locations with small Euclidean distance $\parallel \mathbf{z}_n - \mathbf{z}_{n'} \parallel$ in the latent space are likely to be selected as its neighbors. Similarly, in the input space, the neighborhood probability is given by

$$p(n'|n, \mathbf{X}) = \frac{\exp(-\frac{1}{2} \parallel \mathbf{x}_n - \mathbf{x}_{n'} \parallel^2)}{\sum_{n'' \neq n} \exp(-\frac{1}{2} \parallel \mathbf{x}_n - \mathbf{x}_{n''} \parallel^2)}. \tag{9}$$

The neighbor relationships are preserved when these two probabilities are matched. This is achieved by minimizing the following sum of Kullback-Leibuler divergences between the probabilities,

$$E_{\mathbf{X}} = \sum_{n=1}^{N} \sum_{n' \neq n} p(n'|n, \mathbf{X}) \log \frac{p(n'|n, \mathbf{X})}{p(n'|n, \mathbf{Z})}. \tag{10}$$

The proposed method finds latent locations that properly capture the output covariance while preserving the neighbor relationships by minimizing the following sum of (7) and (10),

$$E = E_{\mathbf{Y}} + \lambda E_{\mathbf{X}}, \tag{11}$$

where $\lambda > 0$ is a hyperparameter that controls how the neighbor relationships are preserved. When $\lambda = 0$, it corresponds to GPLVM. When $\lambda = \infty$ and dimensionality of the latent space is the same with the input space $K = L$, it corresponds to Gaussian process regression since the latent locations become the same with the input locations $\mathbf{Z} = \mathbf{X}$. The proposed method can be seen as a multi-task learning method based on Gaussian process regression. Since the latent locations are learned by using all of the output variables, the learned covariance matrix can improve multi-task regression performance when the output variables are related.

A local optimum solution of latent locations $\mathbf{Z}$ and kernel hyperparameters $\boldsymbol{\theta}$ is obtained by minimizing (11) using gradient-based optimization methods such as the quasi-Newton method [7]. The gradients of the GPLVM term $E_{\mathbf{Y}}$ with respect to a latent location are calculated by

$$\frac{\partial E_{\mathbf{Y}}}{\partial \mathbf{K}} = \frac{1}{2}D\mathbf{K}^{-1} - \frac{1}{2}\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\mathbf{K}^{-1}, \tag{12}$$

$$\frac{\partial k(\mathbf{z}_n, \mathbf{z}_{n'})}{\partial \mathbf{z}_n} = -\frac{\alpha}{\ell^2}\exp\left(-\frac{1}{2\ell^2} \parallel \mathbf{z}_n - \mathbf{z}_{n'} \parallel^2 (\mathbf{z}_n - \mathbf{z}_{n'})\right), \tag{13}$$

using the chain rule. The gradients of the regularization term $E_{\mathbf{X}}$ with respect to a latent location are calculated by

$$\frac{\partial E_{\mathbf{x}}}{\partial \mathbf{z}_n} = \sum_{n' \neq n} \Big(p(n'|n, \mathbf{X}) - p(n'|n, \mathbf{Z})\Big)(\mathbf{z}_n - \mathbf{z}_{n'}), \tag{14}$$

where a sum of forces pulling or pushing $\mathbf{z}$ depends on difference of the neighborhood probability $p(n'|n, \mathbf{X}) - p(n'|n, \mathbf{Z})$.

We can select the hyperparameter value λ by cross-validation on an interpolation problem of the output matrix $\mathbf{Y}$. The cross-validation procedure is as follows. The elements of the output matrix are randomly split into multiple subsets. While the elements in a subset are supposed to be missing, the latent locations and kernel hyperparameters are estimated with a fixed hyperparameter value λ. The interpolation performance of the learned model is evaluated by the rooted mean squared error for the missing elements. The error is averaged over different subsets, and a hyperparameter value that achieved the lowest error is selected. When y_{nd} is missing, its estimated value is given by $\hat{y}_{nd} = \mathbf{k}_n^{\top}\mathbf{K}^{-1}\mathbf{y}_d$, where $\mathbf{k}_n = (k(\mathbf{z}_n, \mathbf{z}_1), \cdots, k(\mathbf{z}_n, \mathbf{z}_N))^{\top}$ and $\mathbf{y}_d = (y_{1d}, \cdots, y_{Nd})^{\top}$, and they are calculated using the observed data. Note that even when all of the output variables are missing with an instance, the latent location can be estimated by using its neighbor relationships.

The proposed model can be seen as a probabilistic generative model. An output value y_{nd} is generated from a Gaussian distribution with mean $f_d(\mathbf{z}_n)$ using the latent location $\mathbf{z}_n$, where the nonlinear function $f_d(\cdot)$ is generated from a Gaussian process prior. The input locations $\mathbf{x}_n$ are not directly generated, but we can consider that the neighbors of the input locations are generated by a multinomial distribution, where the multinomial parameters are defined by latent locations $\mathbf{Z}$ as in (8).

Although we considered that the input variables are real-valued, the proposed method is applicable to other types of input variables that can calculate similarity between two input locations to define neighbor relationships in (9). For example, we can use normalized tree kernels and graph kernels for tree and graph data, respectively, instead of normalized Gaussian kernels in (9).

4. Related Work

With Gaussian process regression, the covariance function is usually defined parametrically, e.g. Gaussian kernels, and its hyperparameters are estimated, e.g. length-scale and noise variance in Gaussian kernels, by maximizing the marginal likelihood. However, the hyperparameter tuning would not be enough to capture the covariance structure when the intrinsic space is different from the observed input space. The proposed model can define the covariance function by flexibly tuning latent locations. A complex form of covariance functions is learned by using multiple kernels [8], by concatenation different kernels [9], and by modeling a spectral density with Gaussian mixtures [10]. These methods are not designed for revealing the latent intrinsic space.

The manifold Gaussian process [11] is related to our work. It transforms input locations by multi-layer neural networks so as to model complex nonsmooth functions. Since the proposed model optimizes a latent location for each input location nonparametrically so as to preserve the neighbor relationships, it can be used for visualizing the latent intrinsic space. In addition, by tuning a single hyperparameter, the proposed model can become the GPLVM and Gaussian process regression, which are successfully used in a wide variety of applications [12], [13], [14], [15]. The warped Gaussian process [16] transforms the output space. Gaussian process regression with input noise [17] assumes that each input location is independently corrupted by Gaussian noise, and does not give latent locations explicitly. On the other hand, with the proposed model, distortion of each location is not independent since latent locations are estimated so as to preserve the neighbor relationships. This enables us to estimate latent locations even without output values.

The GPLVM [3] is an unsupervised method for imputating output values by assuming latent locations. The GPLVM is not designed for regression tasks, and it does use input information. The discriminative GPLVM [18] is a type of regularized GPLVMs. It is used for classification, and outputs are required to be categorical variables. The Gaussian process latent variable set model [19] is a regression method that learns latent locations to model a flexible covariance function. It assumes bag-of-features input, and each feature has a latent location. On the other hand, the proposed model can take any input type, such as real-valued, tree, graph, bag-of-features data, as long as similarity can be calculated, and each instance has a latent location.

Many visualization methods have been proposed, such as multidimensional scaling, principle component analysis, GPLVM, Isomap [20], Laplacian eigenmap [21] and stochastic neighbor embedding (SNE) [5]. They embeds high dimensional unlabeled data into a low dimensional space. Supervised visualization methods have been also proposed, such as canonical correlation analysis (CCA), kernel CCA [22], Fisher linear discriminant analysis and discriminant GPLVM. For example, CCA embeds input and output pairs into a low dimensional space so as to maximize the correlation. The proposed model is a supervised visualization method that reveals an intrinsic input space for better output interpolation. The proposed model can be seen as a combination of GPLVM and SNE, where GPLVM is used for modeling output values, and SNE is used for preserving the neighbor relationships of input locations. In particular, the objective function is simply defined by a weighed sum of the objective functions of GPLVM and SNE.

5. Experiments

5.1 Data

We evaluated the proposed method by using a real-world spatial data set: the comprehensive climate data of North America (NA) [1]. The NA data set consists of monthly climate reports from 1990 to 2002 [23], [24]. We used 16 output variables, such as carbon dioxide and temperature, which were interpolated on 2.5×2.5 degree grid with 125 locations. We normalized all output values to mean zero and unit variance, and conducted experiments with data for each month, where the input variables were latitude and longitude.

5.2 Comparing Methods

We compared the proposed IGPLVM (Proposed) with the following five methods: GP, MTGP, GPLVM, MF and KNN. GP is a Gaussian process regression method, which assumes that output values are determined by the input locations using nonlinear functions with Gaussian process priors. MTGP is a multi-task Gaussian process regression method [25], which learns relationships between output variables. We used the code provided by the authors[2]. GPLVM is a Gaussian process based nonlinear matrix imputation method [3]. MF is a matrix factorization method [26]. It imputates missing values by the product of two low-rank matrices. Both GPLVM and MF do not use input information. KNN is a k-nearest neighbor regression method, which estimates a missing value by the average value of its four neighbors. The dimensionality of the latent space with the proposed method, GPLVM and MF was set at $K = 2$, which is the same with that of the input space. With the proposed method, we selected a hyperparameter λ for each output variable by five-fold cross-validation from $\{0, 1, 10, 10^2, 10^3, 10^4, 10^5\}$. The latent locations were initialized by the input locations.

5.3 Results

We measured the effectiveness of the proposed method by interpolation tasks. Ten percent of output values were randomly selected as test data. The performance was evaluated by rooted mean squared error (RMSE). Table 1 shows the RMSE with the NA data set. The proposed method achieved the lowest average RMSE. This result indicates that the output covariance is properly modeled by distorting the input space with the proposed method. GP achieved low RMSE with output variables whose covariance is determined by the input locations, such as WET and DTR. GPLVM achieved low RMSE with output variables which can be estimated easily from other output variables. Since the proposed method can become the GP and GPLVM by controlling the hyperparameter for each output variable, the RMSE of the proposed method was low for all output variables.

The computational time of the proposed method was 18 seconds with a one-month data using a computer with Xeon 7350 2.93GHz CPU.

Figure 1 shows the original input space and the estimated latent space by the proposed method. When $\lambda = 0$, the latent locations were different from the input locations, since the latent locations did not regularized by the input locations at all. When $\lambda = 10^5$, the latent space was the same with the input space, since the effect of preserving neighbor relationships became dominant. With $\lambda = 10$ and $\lambda = 10^3$, the some neighbor relationships were preserved but some latent locations were transformed so as to model the output covariance. The latent locations of west coast were separated from the other locations (b,c,d), because west coast exhibits different weather from the other area.

6. Conclusion

In this paper, we have proposed a probabilistic model for discovering a latent intrinsic space. The proposed method is based on Gaussian processes, where a latent location is associated with each input location, and output values are determined by the latent locations. The latent locations are estimated so as to preserve the neighbor relationships as well as to capture the output covariance of the given data. Although our results have been encouraging, our framework can be further improved upon in a number of ways. Firstly, we would like to estimate the hyperparameter for controlling input regularization by using the variational Bayesian framework [27]. Secondly, we plan to estimate relationships among output variables using multi-task learning techniques [25]. With the proposed method a single latent space is shared by all output variables. By using estimated task relationships, we can obtain multiple latent spaces that capture the characteristics of individual output variables, and it leads to better interpolation performance for missing values. Finally, we would like to extend our method more scalable by using scalable Gaussian process techniques [28], [29].

[1] http://www-bcf.usc.edu/~liu32/data/NA-1990-20002-Monthly.csv

[2] https://github.com/ebonilla/mtgp

Table 1: RMSE on interpolation tasks with the NA data set for each output variable averaged over all locations and all timestamps. The last row shows the RMSE averaged over all output variables. Values in bold typeface are statistically better at the 5% level from those in normal typecface as indicated by a paired t-test.

	Proposed	GP	MTGP	GPLVM	MF	KNN
CO2	**0.0532**	0.0536	1.3110	0.2342	0.5448	0.1474
CH4	**0.0520**	0.0528	1.3751	0.1966	0.4935	0.1473
CO	**0.0496**	0.0501	1.3807	0.2007	0.5150	0.1411
H2	**0.0479**	0.0484	1.3669	0.1614	0.4116	0.1358
WET	**0.3672**	**0.3684**	1.3502	0.5033	0.6242	0.3773
CLD	**0.2483**	0.2543	1.3201	0.3580	0.5253	0.2817
VAP	**0.1911**	0.2071	1.3712	0.2511	0.3957	0.2495
PRE	0.5289	0.5201	1.3420	0.5991	0.6857	**0.5098**
FRS	0.3502	0.4480	1.3528	**0.3371**	0.5613	0.4409
DTR	**0.4579**	**0.4611**	1.3947	0.5098	0.5873	0.4704
TMN	**0.1832**	0.3315	1.3694	0.2039	0.3661	0.3531
TMP	**0.1568**	0.3190	1.3786	0.1762	0.3294	0.3419
TMX	**0.1866**	0.3319	1.3706	0.2082	0.3435	0.3578
GLO	**0.1901**	**0.1901**	1.3755	0.2755	0.3618	0.2311
ETR	**0.0641**	0.0647	1.3677	0.1376	0.3714	0.1733
ETRN	**0.0568**	0.0572	1.3034	0.1239	0.3190	0.1547
Average	**0.1990**	0.2349	1.3581	0.2798	0.4647	0.2821

Acknowledgments

A part of the research results has been achieved by “Research and Development on Fundamental and Utilization Technologies for Social Big Data,” the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN.

References

[1] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[2] N. Cressie, *Statistics for spatial data*. Wiley New York, 1993.

[3] N. D. Lawrence, “Gaussian process latent variable models for visualisation of high dimensional data,” in *Advances in Neural Information Processing Systems*, 2004, pp. 329–336.

[4] N. Lawrence, “Probabilistic non-linear principal component analysis with Gaussian process latent variable models,” *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[5] G. E. Hinton and S. T. Roweis, “Stochastic neighbor embedding,” in *Advances in Neural Information Processing Systems*, 2002, pp. 833–840.

[6] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. 2579–2605, p. 85, 2008.

[7] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical Programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[8] A. Melkumyan and F. Ramos, “Multi-kernel Gaussian processes,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2011, pp. 1408–1413.

[9] D. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, “Structure discovery in nonparametric regression through compositional kernel search,” in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 1166–1174.

[10] A. G. Wilson and R. P. Adams, “Gaussian process kernels for pattern discovery and extrapolation,” in *Proceedings of the International Conference on Machine Learning*, 2013.

[11] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth, “Manifold gaussian processes for regression,” *arXiv preprint arXiv:1402.5876*, 2014.

[12] J. Wang, A. Hertzmann, and D. M. Blei, “Gaussian process dynamical models,” in *Advances in Neural Information Processing Systems*, 2005, pp. 1441–1448.

[13] S. Hou, A. Galata, F. Caillette, N. Thacker, and P. Bromiley, “Real-time body tracking using a Gaussian process latent variable model,” in *Proceedings of the 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[14] M. A. Alvarez, D. Luengo, and N. D. Lawrence, “Linear latent force models using Gjaussian processes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2693–2705, 2013.

[15] S. Bratieres, N. Quadrianto, S. Nowozin, and Z. Ghahramani, “Scalable Gaussian process structured prediction for grid factor graph applications,” in *Proceedings of the 31st International Conference on Machine Learning*, 2014, pp. 334–342.

[16] E. Snelson, C. E. Rasmussen, and Z. Ghahramani, “Warped Gaussian processes,” in *Advances in Neural Information Processing Systems*, 2004, pp. 337–344.

[17] A. McHutchon and C. E. Rasmussen, “Gaussian process training with input noise,” in *Advances in Neural Information Processing Systems*, 2011, pp. 1341–1349.

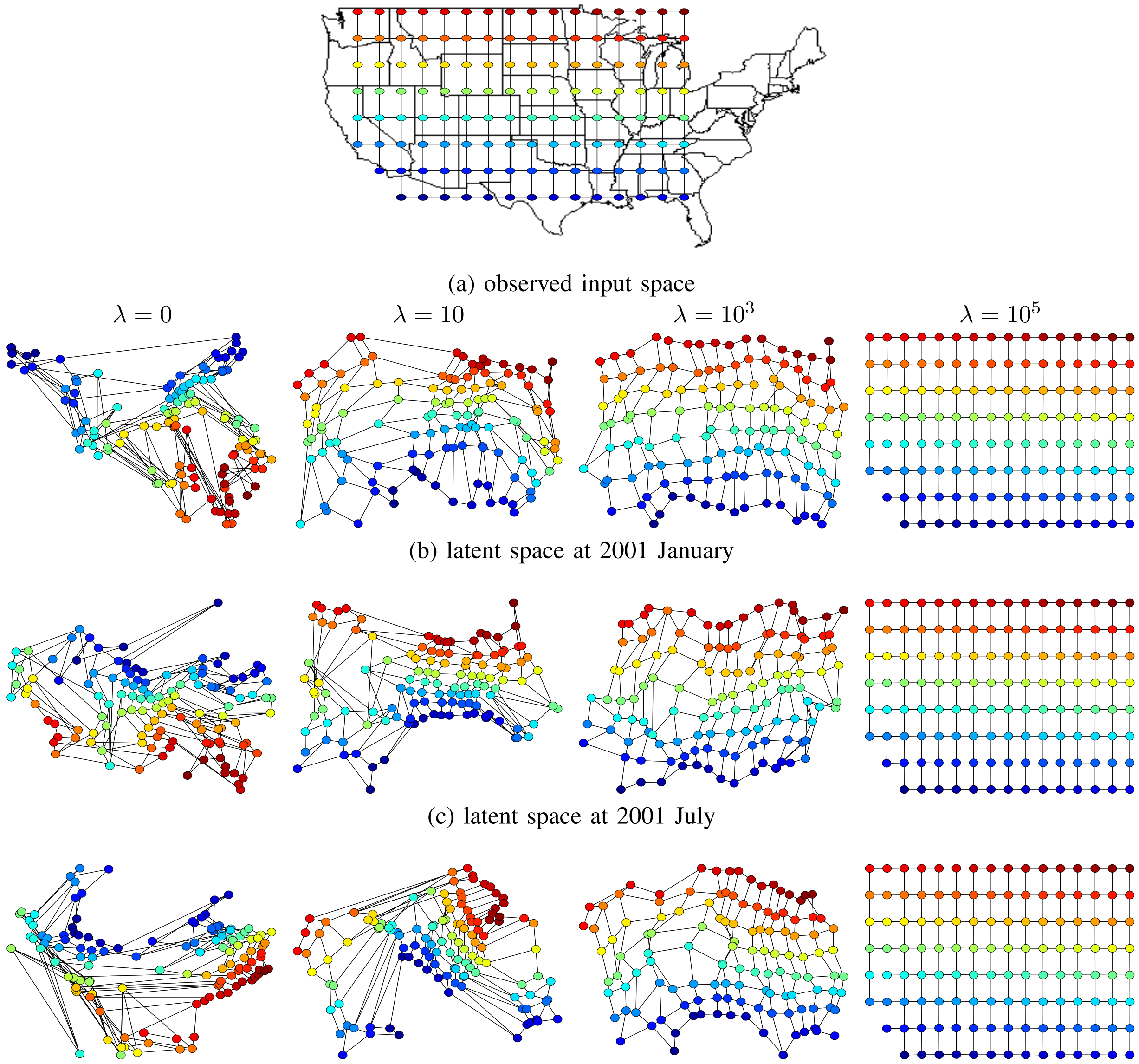
[18] R. Urtasun and T. Darrell, “Discriminative Gaussian process latent variable model for classification,” in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 927–934.

[19] Y. Yoshikawa, T. Iwata, and H. Sawada, “Non-linear regression for bag-of-words data via Gaussian process latent variable set model,” in *Proceedings of th 28th AAAI Conference on Artificial Intelligence*, ser. AAAI, 2015.

[20] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[21] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[22] S. Akaho, “A kernel method for canonical correlation analysis,” in *In Proceedings of the International Meeting of the Psychometric Society*, 2001.



(a) observed input space

(b) latent space at 2001 January

(c) latent space at 2001 July

(d) latent space at 2001 October

Fig. 1: The observed input space (a), and the latent space estimated by the proposed method (b,c,d) with the NA data set. The locations that are closely located at the input space are connected by edges. The color represents the locations in the input space.

[23] M. T. Bahadori, Q. R. Yu, and Y. Liu, "Fast multivariate spatio-temporal analysis via low rank tensor learning," in *Advances in Neural Information Processing Systems*, 2014, pp. 3491–3499.

[24] A. C. Lozano, H. Li, A. Niculescu-Mizil, Y. Liu, C. Perlich, J. Hosking, and N. Abe, "Spatial-temporal causal modeling for climate change attribution," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009, pp. 587–596.

[25] E. Bonilla, K. M. Chai, and C. Williams, "Multi-task Gaussian process prediction," in *Advances in Neural Information Processing Systems*, 2008.

[26] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[27] M. Titsias and N. Lawrence, "Bayesian gaussian process latent variable model," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 844–851.

[28] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems 18*, 2006, pp. 1257–1264.

[29] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," in *Uncertainty in Artificial Intelligence*, 2013.