Gaussian process prediction for time series of structured data

Benjamin Paassen, Christina Göpfert and Barbara Hammer *

CITEC center of excellence
Bielefeld University - Germany

(This is a preprint of the publication [12], as provided by the authors.)

Abstract

Time series prediction constitutes a classic topic in machine learning with wide-ranging applications, but mostly restricted to the domain of vectorial sequence entries. In recent years, time series of structured data (such as sequences, trees or graph structures) have become more and more important, for example in social network analysis or intelligent tutoring systems. In this contribution, we propose an extension of time series models to strucured data based on Gaussian processes and structure kernels. We also provide speedup techniques for predictions in linear time, and we evaluate our approach on real data from the domain of intelligent tutoring systems.

1 Introduction

Time series prediction constitutes a classic topic in machine learning with wide-ranging and successful applications in physics, sociology and medicine [18]. In recent years, time series of structured data (sequences, trees or graphs) have become more and more important, describing for example the development of social networks [17] or learner solutions in intelligent tutoring systems over time [8]. Classic time series prediction models such as ARIMA, NARX, Kalman filters, recurrent networks or reservoir models focus on vectorial data representations, and they are not equipped to handle time series of structured data [18]. In this contribution, we propose an extension of Gaussian process (GP) regression, which is capable of predicting time series of structured data.

GP regression has been successfully applied on time series of vectorial data before [16, 19], but not yet for structured data. To extend GP regression to structured data, we rely on two observations: First, GPs are based on kernel values for the given data as input. Hence we can build upon the vast literature of distance measures and kernels for structured data, such as alignment distances,

*Funding by the DFG under grant number HA 2719/6-2 and the CITEC center of excellence (EXC 277) is gratefully acknowledged.

tree and graph kernels, to access structured data instead of vectors as time series entries [13, 1, 2]. Second, as we will show in this contribution, a special choice of the prior allows us to express the predictions provided by GPs as an affine combination of given data. Hence we can rely on established embeddings of the space of structured objects, which is a discrete data space in itself, in a smooth kernel or pseudo-Euclidean space, and we can access such outputs of a GP for structured data e.g. via efficient distance computations [6, 7]. An additional challenge is posed by the high computational complexity of GPs as regards the number of data points, and the structure kernel computation. For speed-up, we apply state-of-the-art approximation methods for the Gaussian Processes [3] as well as the dissimilarity and kernel data [5].

Now we will first extend time series prediction to structured data via the interface of dissimilarity measures and kernels. Second, we speed up the prediction to a linear time technique by applying state-of-the-art approximation methods. Finally, we evaluate our approach on two datasets from the domain of intelligent tutoring systems.

2 Gaussian processes regression for time series prediction

A Gaussian process (GP) is uniquely characterized by multivariate random variables which follow a Gaussian distribution, where the covariance matrix is given by a kernel matrix. Given examples $\{(x_i, y_i)\}_{i=1,\ldots,N}$, where x_i is element of some space $\mathcal{X}$ and y_i is a real value or vector, and a new point x^* with (unknown) $y^* \in \mathbb{R}$. The conditional density function $p(y_1, \ldots, y_N, y^* | x_1, \ldots, x_N, x^*)$ is the Gaussian

$$\mathcal{N}\Big(\theta_1, \ldots, \theta_N, \theta^*, \begin{pmatrix} \mathrm{K} + \tilde{\sigma}^2 I^N & \vec{\mathrm{k}}^T \\ \vec{\mathrm{k}} & k(x^*, x^*) \end{pmatrix}\Big) \tag{1}$$

where θ_i is the prior mean for y_i, k a kernel on $\mathcal{X}$, K is the matrix $\big(k(x_i, x_{i'})\big)$, $\vec{\mathrm{k}} := \big(k(x^*, x_1), \ldots, k(x^*, x_N)\big)$, I^N is the N-dimensional identity matrix and $\tilde{\sigma}^2$ is the variance of the input noise. Set $\mathrm{Y} = (y_1, \ldots, y_N)^T$. Marginalisation enables the inference of y^* via its density [15, p. 27]:

$$p\Big(y^* \Big| x^*, x_1, \ldots, x_N, y_1, \ldots, y_N\Big) = \mathcal{N}\Big(\mu, \sigma^2\Big) \quad \text{where} \tag{2}$$

$$\mu = \theta^* + \vec{\mathrm{k}} \cdot (\mathrm{K} + \tilde{\sigma}^2 \cdot I^N)^{-1} \cdot (\mathrm{Y} - \Theta) \tag{3}$$

$$\sigma^2 = k(x^*, x^*) - \vec{\mathrm{k}} \cdot (\mathrm{K} + \tilde{\sigma}^2 \cdot I^N)^{-1} \cdot \vec{\mathrm{k}}^T \tag{4}$$

To apply GP regression to time series prediction, we reframe time series prediction as follows: Assume that example time series $\bar{x}^1, \ldots, \bar{x}^M$ are given, where $\bar{x}^j = (x_1^j, \ldots, x_{T_j}^j)$ with $x_t^j \in \mathcal{X}$. Then the task is to infer the successor x_{t+1}^j from its history $(x_1^j, \ldots, x_t^j)$ for all j and t. Following the Markov assumption, all but the last history entry become irrelevant. This leads to the regression problem with input-output pairs $\{(x_t^j, x_{t+1}^j)\}_{t=1,\ldots,T_j-1}^{j=1,\ldots,M}$, which can be modelled by GP regression, provided real vectors x_t^j. For time series models, a natural prior is

to stay where you are, that is $\theta_t^j := x_t^j$. This leads to the predictive mean $\mu = x_T^* + \vec{\mathrm{k}} \cdot (\mathrm{K} + \tilde{\sigma}^2 \cdot I^N)^{-1} \cdot (\mathrm{Y} - \mathrm{X})$ with $\mathrm{X} = (x_1^1, \ldots, x_{T_1-1}^1, \ldots, x_1^M, \ldots, x_{T_M-1}^M)^T$ and $\mathrm{Y} = (x_2^1, \ldots, x_{T_1}^1, \ldots, x_2^M, \ldots, x_{T_M}^M)^T$ and predictive variance (4).

3 Predicting structured data

Classic GP regression has been phrased for vectorial data. It is straightforward to extend GPs to structured input data by means of structure kernels k. Time series prediction for structures, however, deals with structured input and *output* pairs. Here we build upon previous results regarding kernel and dissimilarity spaces. We note that numerous powerful dissimilarity measures and kernels for structured data exist, for example [13, 1, 2].

Any symmetric dissimilarity measure d or kernel k on $\mathcal{X}$ corresponds to a vectorial embedding ϕ of the data in a pseudo-Euclidean or Krein space $\mathcal{X}'$, such that $d(x, x') = \sqrt{\langle \phi(x) - \phi(x'), \phi(x) - \phi(x') \rangle}$ and $k(x, x') = \langle \phi(x), \phi(x') \rangle$ [14]. The core insight is that time series prediction via GP regression can be used in this implicit space $\mathcal{X}'$ with neither having to refer to ϕ nor $\mathcal{X}'$ explicitly. This is obvious for inputs provided a valid kernel is present. For outputs, we represent the predicted point in $\mathcal{X}'$ as an affine combination of known points in $\mathcal{X}'$, whereby only the coefficients of this affine combination are computed but not the embedded point nor the underlying structure. This enables us to further process data by any method which refers to kernels or dissimilarities between structures only, such as kernel- or dissimilarity-based classification, clustering, or visualization [6, 7]. For such postprocessing, it is required that the GP prediction is provided by a linear (for kernels) or affine (for dissimilarities) combination of data [6], which is fulfilled due to our prior:

Theorem 1. *The mean of the GP prediction as defined above is an affine combination of the points $\phi(x_T^*), \phi(x_1^1), \ldots, \phi(x_{T_M}^M)$.*

Proof. We define $\vec{\gamma} = (\gamma_1^1, \ldots, \gamma_{T_1-1}^1, \ldots \gamma_1^M, \ldots, \gamma_{T_M-1}^M) := \vec{\mathrm{k}} \cdot (\mathrm{K} + \tilde{\sigma}^2 \cdot I^N)^{-1}$. The predictive mean is given as $\mu = \phi(x_T^*) + \vec{\gamma} \cdot (\mathrm{Y} - \mathrm{X})$ which yields

$$\mu = \phi(x_T^*) + \sum_{j=1}^{M} \sum_{t=1}^{T_j - 1} \gamma_t^j \cdot \Big(\phi(x_{t+1}^j) - \phi(x_t^j) \Big) \tag{5}$$

$$= \phi(x_T^*) + \sum_{j=1}^{M} -\gamma_1^j \cdot \phi(x_1^j) + \left(\sum_{t=2}^{T_j - 1} (\gamma_{t-1}^j - \gamma_t^j) \cdot \phi(x_t^j) \right) + \gamma_{T_j-1}^j \cdot \phi(x_{T_j}^j) \tag{6}$$

$\phi(x_T^*)$ is weighted with coefficient 1 and all other coefficients add up to zero. Thus, an affine combination results. □

We denote the affine combination $\mu = \vec{\alpha} \cdot \Phi$ with coefficients $\vec{\alpha}$. Kernel values between a point $x \in \mathcal{X}$ and μ can be obtained via

$$k(x, \mu) = \langle \phi(x), \vec{\alpha} \cdot \Phi \rangle = \vec{\alpha} \cdot \vec{\mathrm{k}}^T \tag{7}$$

where $\vec{\mathrm{k}} = \big(k(x, x_1), \ldots, k(x, x_N)\big)$. Dissimilarities are given by

$$d(x, \mu)^2 = \langle \phi(x) - \vec{\alpha} \cdot \Phi, \phi(x) - \vec{\alpha} \cdot \Phi \rangle = \vec{\alpha} \cdot \vec{\mathrm{d}}^T - \frac{1}{2} \vec{\alpha} \cdot \mathrm{D} \cdot \vec{\alpha}^T \tag{8}$$

where $\vec{\mathrm{d}} = \big(d(x, x_1)^2, \ldots, d(x, x_N)^2\big)$ and D is the matrix of pairwise squared distances for all data points [6, pp. 9-10]. Both these equations form the basis for the application of further dissimilarity-based or kernel-based methods.

4 Fast Gaussian process regression

GP regression involves the inversion of the matrix $(\mathrm{K} + \tilde{\sigma}^2 \cdot I^N)$, resulting in $\mathcal{O}(N^3)$ complexity. A variety of efficient approximation schemes exist [15]. Recently, the robust Bayesian Committee Machine (rBCM) has been introduced as particularly fast and accurate approximation [3]. The rBCM approach is to distribute the examples into C disjoint sets, based e.g. on clustering in the input data space. For each of these sets, a separate GP regression is used, yielding the predictive distribution $\mathcal{N}(\mu_c, \sigma_c^2)$. These distributions are combined to the final predictive distribution $\mathcal{N}(\mu_{\mathrm{rBCM}}, \sigma^2_{\mathrm{rBCM}})$ with

$$\sigma^{-2}_{\mathrm{rBCM}} = \sum_{c=1}^{C} \frac{\beta_c}{\sigma_c^{*2}} + \left(1 - \sum_{c=1}^{C} \beta_c\right) \cdot \frac{1}{\sigma^2_{\mathrm{prior}}} \tag{9}$$

$$\mu_{\mathrm{rBCM}} = \sigma^2_{\mathrm{rBCM}} \cdot \left(\sum_{c=1}^{C} \frac{\beta_c}{\sigma_c^{*2}} \cdot \mu_c^* + \left(1 - \sum_{c=1}^{C} \beta_c\right) \cdot \frac{1}{\sigma^2_{\mathrm{prior}}} \cdot \theta^* \right) \tag{10}$$

Here, $\sigma^2_{\mathrm{prior}}$ is the variance of the prior for the prediction, which is a new meta-parameter introduced in the model. The weights β_c are supposed to be a measure for the predictive power of the single GP experts. As suggested by [3], we use the differential entropy, given as $\beta_c = \frac{1}{2} \cdot \Big(\log(\sigma^2_{\mathrm{prior}}) - \log(\sigma_c^2) \Big)$. Also for rBCM, the predictive mean is an affine combination, because each GP expert returns an affine combination and the rBCM combines those predictions in an affine way.

Provided sufficiently many clusters, the size of one cluster can be regarded as a constant, such that an overall linear time prediction results. For the initial clustering, linear time approximations for dissimilarity data can be used [4, 6]. In this contribution, we rely on subsampling of a constant-size subset, which is clustered by relational neural gas and extended to the whole set afterwards [6]. Thereby we avoid the quadratic runtime of relational neural gas on the full dataset.

5 Experiments

We evaluate the predictive performance of our approach on two datasets[1], simulating the behaviour of a student in an intelligent tutoring system. Our mo-

[1] Both datasets are available online at `http://doi.org/10.4119/unibi/2900666` and `http://doi.org/10.4119/unibi/2900684` respectively.

tivation is to guide a fictional student towards a correct solution by predicting a likely extension of her current program based on the development of other students. We simulate such a development by starting with a real, finished program, representing it via its abstract syntax tree and iteratively removing the last semantically important node (class declarations, variable declarations, loops, etc.) and its children until the program is empty. By reversing the order of the resulting programs, we obtain the desired simulation.

MiniPalindrome: This dataset consists of 48 Java programs recognizing palindromic inputs created by a Java expert. The programs come in six different variations described in [9]. Our simulation results in 834 data points.

Sorting: This is a benchmark dataset of 64 Java sorting programs taken from the web, previously described in [10]. Each program implements one of two sorting algorithms (BubbleSort or InsertionSort). Here, our simulation results in 800 data points overall.

Methods: As a dissimilarity measure on computer programs, we applied an affine sequence alignment with parameters obtained via metric learning, as described in [13]. We transformed the dissimilarities to a kernel via a radial basis function (RBF) and clip eigenvalue correction. To prevent an excessive number of costly alignment calculations we applied the Nyström technique to approximate the distance matrix via a subset of 16 time series, considering only every fourth program [5]. We applied relational neural gas to distribute the dataset into 4 clusters as preprocessing for the rBCM. The meta-parameters for the prediction methods were set using a simple data-based heuristic: Let $\bar{d}$ be the average dissimilarity in the data set, then we set $\tilde{\sigma} = \bar{d}$, $\sigma_{\text{prior}} = 2 \cdot \bar{d}$ and the RBF bandwidth to $0.1 \cdot \bar{d}$ for KR and to $0.5 \cdot \bar{d}$ for GP and rBCM.

We evaluated the root mean square error (RMSE) for unseen data points in a crossvalidation (6 folds for the first dataset and 8 for the second). To obtain the squared distances between the correct next step and the predictive mean we used Equation 8. We compared the prediction RMSE of the rBCM and GP with the trivial prediction of just staying in the same place ($x^*_{T+1} = x^*_T$, baseline) and with a simple Nadaraya-Watson estimator [11] (KR), which is defined as:

$$x^*_{T+1} = \frac{\sum_{j=1}^{M} \sum_{t=1}^{T_j-1} k(x^*_T, x^j_t) \cdot x^j_{t+1}}{\sum_{j=1}^{M} \sum_{t=1}^{T_j-1} k(x^*_T, x^j_t)} \tag{11}$$

We also evaluated the runtime for all prediction methods.

Results: The results are shown in Table 1. In both experiments, the rBCM outperforms the baseline measures notably and does not perform worse than a full Gaussian process. As expected, the rBCM is also much faster than full GP regression (including the time needed for clustering).

	MiniPalindrome			Sorting		
method	RMSE	std. dev.	runtime	RMSE	std. dev.	runtime
rBCM	0.089	0.0129	0.12s	0.069	0.0058	0.11s
GP	0.090	0.0129	3.08s	0.069	0.0063	2.36s
KR	0.300	0.0165	0.14s	0.299	0.0062	0.14s
baseline	0.287	0.0452	0.00s	0.337	0.0243	0.00s

Table 1: The resulting prediction error (RMSE) and its standard deviation (std. dev.) for all three experimental datasets. For the first experiment the overall runtime in seconds is provided as well.

6 Conclusion

Gaussian processes seem promising to predict time series of structured data, and relational data in general. By returning an affine combination, they enable further processing, such as classification and clustering. Using state-of-the-art approximation methods it is possible to obtain good-quality predictions in linear time. However, there is work that remains to be done: First, usual hyperparameter optimization techniques depend on a vectorial data representation [3] and one has to adapt them for a relational case. Second, an affine combination might not be a sufficient data representation of the predicted point for some applications, for example for feedback provision in intelligent tutoring systems. For such cases, an inverse problem has to be solved: Finding the original point that maps to the affine combination in the pseudo-Euclidean space. Both problems pose interesting challenges for further research in the field.

References

[1] F. Aiolli, G. D. S. Martino, and A. Sperduti. An efficient topological distance-based tree kernel. *IEEE Trans. Neural Netw. Learning Syst.*, 26(5):1115–1120, 2015.

[2] G. Da San Martino and A. Sperduti. Mining structured data. *Computational Intelligence Magazine, IEEE*, 5(1):42–49, Feb 2010.

[3] M. P. Deisenroth and J. W. Ng. Distributed gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1481–1490, 2015.

[4] A. Gisbrecht, B. Mokbel, F. Schleif, X. Zhu, and B. Hammer. Linear time relational prototype based learning. *Int. J. Neural Syst.*, 22(5), 2012.

[5] A. Gisbrecht and F.-M. Schleif. Metric and non-metric proximity transformations at linear costs. *Neurocomputing*, 167:643–657, 2015.

[6] B. Hammer and A. Hasenfuss. Topographic mapping of large dissimilarity data sets. *Neural Computation*, 22(9):2229–2284, 2010.

[7] D. Hofmann, F.-M. Schleif, B. Paaßen, and B. Hammer. Learning interpretable kernelized prototype-based models. *Neurocomputing*, 141:84–96, 2014.

[8] K. R. Koedinger, E. Brunskill, R. S. Baker, E. A. McLaughlin, and J. Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.

[9] B. Mokbel, S. Gross, B. Paaßen, N. Pinkwart, and B. Hammer. Domain-independent proximity measures in intelligent tutoring systems. *Proceedings of the 6th International Conference on Educational Data Mining (EDM)*, pages 334–335, 2013.

[10] B. Mokbel, B. Paaßen, F.-M. Schleif, and B. Hammer. Metric learning for sequences in relational lvq. *Neurocomputing*, 169:306–322, 2015.

[11] E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[12] B. Paaßen, C. Göpfert, and B. Hammer. Gaussian process prediction for time series of structured data. In M. Verleysen, editor, *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 41–46. i6doc.com, 2016.

[13] B. Paaßen, B. Mokbel, and B. Hammer. Adaptive structure metrics for automated feedback provision in intelligent tutoring systems. *Neurocomputing*, 2016.

[14] E. Pękalska. *The dissimilarity representation for pattern recognition: foundations and applications*. PhD thesis, Delft University of Technology, 2005.

[15] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[16] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2012.

[17] N. Santoro, W. Quattrociocchi, P. Flocchini, A. Casteigts, and F. Amblard. Time-varying graphs and social network analysis: Temporal indicators and metrics. *arXiv preprint arXiv:1102.0629*, 2011.

[18] R. H. Shumway and D. S. Stoffer. *Time series analysis and its applications*. Springer Science & Business Media, 2013.

[19] J. Wang, A. Hertzmann, and D. M. Blei. Gaussian process dynamical models. In *Advances in neural information processing systems*, pages 1441–1448, 2005.