How to Encode Dynamic Gaussian Bayesian Networks as Gaussian Processes?

Mattis Hartwig and Ralf Möller

University of Lübeck, Institute of Information Systems
{hartwig,moeller}@ifis.uni-luebeck.de

Abstract. One dimensional versions of the Markov chain and the hidden Markov model have been generalized as Gaussian processes. Currently these approaches support only a single dimension which is limiting their usability. In this paper we encode the more general dynamic Gaussian Bayesian network as a Gaussian process and thus allow arbitrary number of dimensions and arbitrary connections between time steps. Our developed Gaussian process based formalism has the advantage of supporting a direct inference from any time point to the other without propagation of evidence throughout the whole network, flexibility to combine the covariance function with others if needed and keeping all properties of the dynamic Gaussian Bayesian network.

Keywords: Gaussian process · kernel · Bayesian network

1 Introduction

Understanding the fundamental relationships between different probabilistic models is vital to guide further research and to exploit the benefits of different approaches. Two specific types of one-dimensional Gaussian distributed probabilistic graphical models (PGMS), the Markov chain (MC) and the hidden Markov model (HMM), have already been encoded as Gaussian Processes (GPs), showing the generalizing power of GPs [6]. As Murphy [11] has elaborated, dynamic Bayesian networks, are a more general type of a PGM compared to the MC and the HMM. Consequently, it is an improvement and thus a contribution to encode the dynamic Gaussian Bayesian network (DGBN) as a Gaussian Process, which is focus of this paper. By encoding we mean a generalization of the DGBN into the GP framework while maintaining all characteristics of the original DGBN.

DGBNs in general offer a sparse and interpretable representation for probabilistic distributions and allow to model (in)dependencies between its random variables [7, 9]. The interpretability of the modeling language also makes it possible to construct DGBNs based on expert knowledge instead of or as an addition to learning them from data [3, 4]. Komurlu and Bilgic[8] explicitly favor the usage of a DGBN over a GP in their application because in classic GPs, dependencies between output random variables are not easily taken into account. There are also downsides of DGBNs. First, the time dimension is still discrete which brings up the problem of finding the right sampling rate. Second, evidence is usually

propagated through the graphical structure which can be computational expensive. Third, they are based on linear relationships between random variables which makes it difficult to model certain real-world phenomena, e.g. periodic behaviors.

Gaussian Processes (GPs) are another approach applied for modeling time-series [16, 5] and have been rather recently brought into focus of the machine learning community [13]. Both DGBNs and GPs have Gaussian distributions over their random variables at any point in time. In contrast to DGBNs, GPs are continuous on the time dimension and allow direct inference without propagation of evidence through a network. Additionally, an existing GP that models a certain behavior can be easily extended or adapted by making changes to its covariance function. Drawbacks of GPs are that modeling multiple outputs at once is challenging [1] and that modeling a detailed interpretable (in)dependence structure as it is done in a DGBNs is currently not possible. Encoding the multidimensional and Markovian aspects of a DGBN into a Gaussian process could combine the benefits of two models.

The remainder of the paper has following structure. We start by explaining the preliminaries about PGMs and GPs. After discussing related work, we construct GPs Dynamic Gaussian Bayesian Networks with arbitrary connections between time steps. We conclude with a discussion of benefits and downsides of the created GPs and with an agenda for further research in that area.

2 Preliminaries

In this section we introduce PGMs, GPs and kernel functions for GPs. Afterwards, we briefly review the advantages of the two models, which also motivates combining them.

2.1 Probabilistic Graphical Models

This section gives a brief overview about the different types of PGMs used in this paper. For further details we refer to the work by Koller et al. [7], Pearl [12] and Murphy [11].

In general, a PGM is a network with nodes for the random variables and edges to describe relations between them. When looking at random variables over time dynamic variants of PGMs are used and when looking at continuous random variables often Gaussian PGMs are used. Dynamic Gaussian Bayesian networks (DGBNs) are a general representation for the development of continuous random variables over time. A Gaussian Markov chain, which describes the development of a single Gaussian distributed random variable over time, and a Gaussian hidden Markov model, which contains two random variables over time, are special cases of the DGBN [11]. A DGBN allows arbitrary links between the random variables [11]. Figure 1 contains illustrations of three different types of DGBNs. Since Hartwig et al.[6] have already worked on Gaussian Markov chains and Gaussian hidden Markov models, this paper focuses on generalizing

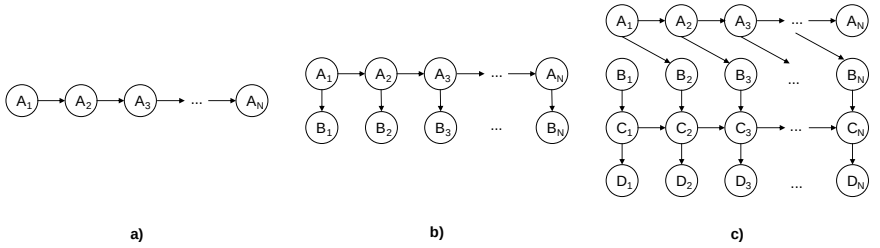

Fig. 1. Three different types of PGMs: a) Markov chain b) hidden Markov model c) dynamic Bayesian network

the approach to DGBNs. The only restriction is that we do not allow connections between random variables within an individual time step.

In general, we denote the set of random variables as $\mathbb{X}$ and the set random variables that are influencing a specific random variable $X \in \mathbb{X}$ as its parents $Pa(X)$. Each random variable follows a conditional Gaussian probability distribution that is linearly dependent on its parent and is given by

$$P(X|Pa(X)) \sim N\left(\mu_X + \sum_{\Pi \in Pa(X)} \beta_{X,\Pi}(\pi - \mu_\Pi); \sigma_X^2\right), \tag{1}$$

where μ_A and μ_Π are the unconditional means of X and Π respectively, π is the realization of Π, σ_X^2 is the variance of X and $\beta_{X,\Pi}$ represents the influence of the parent Π on its child X.

A DGBN can be represented by a pair of BNs. The first BN defined the prior distribution $P(\mathbb{X}_1)$ at time $t = 1$. The second BN is a two-slice temporal BN (2TBN) which defines $P(\mathbb{X}_t|\mathbb{X}_{t-1})$. This representation is parameterized by a mean vector μ and covariance matrix $\mathbf{\Sigma}$ for the first BN and a transition matrix $\mathbf{M}$ containing the linear relationships over time. Figure 2 contains a visualization of a three dimensional 2TBN.

2.2 Gaussian Processes

A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution [13]. A GP can be interpreted as a distribution over functions on a spatial dimension, which is in our case the time dimension t. It is completely specified by its mean $\mu = m(t)$ and its covariance function $k(t, t')$ and can be written as

$$f(t) \sim GP(m(t), k(t, t')). \tag{2}$$

The covariance function (also known as kernel function) describes the similarity of function values at different points in time (t and t') and influences the shape of the function space [13].

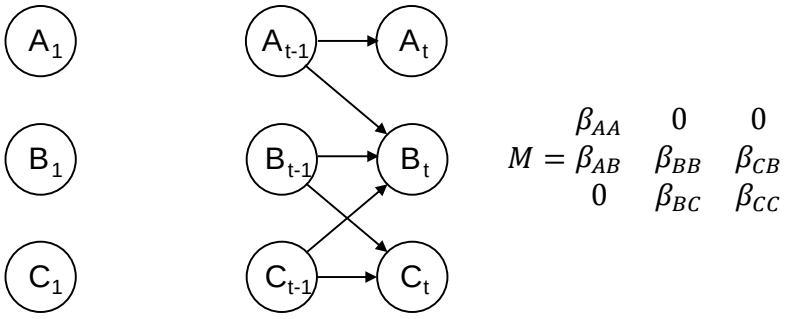

Fig. 2. A DGBN represented by a prior and a 2TBN

If we have a dataset that consists of an input vector $\mathbf{t}$ and an output vector $\mathbf{y}$, we can define any vector of time points $\mathbf{t}^*$ for which we would like to calculate the posterior distribution. The joint distribution over the observed and the unknown time points is given by

$$p\left(\begin{bmatrix} y \\ y* \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu(t) \\ \mu(t*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{t},\mathbf{t}) & K(\mathbf{t},\mathbf{t}^*) \\ K(\mathbf{t}^*,\mathbf{t}) & K(\mathbf{t}^*,\mathbf{t}^*) \end{bmatrix}\right), \tag{3}$$

where $K(\mathbf{t},\mathbf{t}^*)$ is a covariance matrix produced by plugging all values from $(\mathbf{t},\mathbf{t}^*)$ into the covariance function $k(t,t')$. By applying the conditional probability rules for multivariate Gaussians [16] we obtain the posterior $P(\mathbf{y}^*)$ with mean $\mathbf{m}^*$ and covariance matrix C^*

$$P(\mathbf{y}^*) = N(\mathbf{m}^*, C^*), \tag{4}$$

where

$$\mathbf{m}^* = \mu(\mathbf{t}^*) + K(\mathbf{t}^*,\mathbf{t})K(\mathbf{t},\mathbf{t})^{-1}(\mathbf{y} - \mu(\mathbf{t})) \tag{5}$$

and

$$C^* = K(\mathbf{t}^*,\mathbf{t}^*) - K(\mathbf{t}^*,\mathbf{t})K(\mathbf{t},\mathbf{t})^{-1}K(\mathbf{t}^*,\mathbf{t})^T. \tag{6}$$

2.3 Kernel Functions

Rasmussen [13] provides an overview of different possible kernels with the squared exponential kernel

$$k_{SE}(t,t') = \sigma^2 \exp\left(-\frac{(t-t')^2}{2l^2}\right), \tag{7}$$

where σ^2 and l are hyperparameters for the signal noise and the length scale respectively, being a commonly used one.

A valid kernel $k : \mathbb{T} \times \mathbb{T} \to \mathbb{R}$ for a GP needs to fulfill two characteristics [13]:

- symmetry, i.e., $k(t,t') = k(t',t)$ for all t and t',

– being positive semidefinite, i.e., symmetry and $\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(t_i, t_j)$ for $n \in \mathbb{N}$, $t_1, ..., t_n \in \mathbb{T}$, $c_1, ..., c_n \in \mathbb{R}$.

Valid Kernels can be constructed of other kernels. Bishop [2] lists valid kernel operations from which we use the following subset in later sections. Given valid kernels $k_1(t, t')$, $k_2(t, t')$ and a constant c, the following kernels will also be valid:

$$k(t, t') = ck_1(t, t'), \tag{8}$$

$$k(t, t') = k_1(t, t') + k_2(t, t'), \tag{9}$$

$$k(t, t') = \exp(k_1(t, t')), \tag{10}$$

$$k(t, t') = k_1(t, t')k_2(t, t'). \tag{11}$$

2.4 Benefits of the Models

In their work on one-dimensional markov Chains and hidden Markov models Hartwig et al. [6] have listed benefits of the GPs and PGMs, which we will review here briefly. PGMs can capture (conditional) dependencies and independencies of the random variables very intuitively [7] and be thus also constructed by incorporating expert knowledge (either entirely or as a prior). PGMs can be naturally multidimensional, which allows representing the probability distribution over multiple random variables simultaneously. Last but not least, PGMs have already been used in many applications and therefore a wide range of inference and learning tactics have been developed [7].

The usage of GPs has also benefits. In general GPs have a continuous spatial dimension which allows to model continuous changes directly and without the need of discretization. GPs are nonparametric and directly incorporate a quantification of uncertainty. Because of their joint Gaussian characteristics, calculating posterior distributions is straightforward and relatively efficient [16].

Consequently, converting the PGMs to GPs while retaining the PGM benefits is a promising research direction.

3 Related Work

There have been three different streams to bring graphical or relational models together with GPs. One research stream known as relation learning uses multiple GPs to identify probabilistic relations or links within sets of entities [19, 20]. A second research stream uses GPs for transition functions in state space models. Frigola-Alcalde [5] has researched different techniques for learning state space models that have GP priors over their transition functions and Turner [18] has explored change point detection in state space models using GPs. A third research stream focuses on constructing covariance functions for GPs to mimic certain behaviors from other models. Reece and Roberts [15, 14] have shown that they can convert a specific Kalman filter model for the near constant acceleration model into a kernel function for a GP and then combine that kernel function

with other known kernels to get better results temporal-spatial predictions. Rasmussen [13] has introduced GP kernels for, e.g., a wiener process. As mentioned above Hartwig et al. [6] have constructed a kernel for a one-dimensional DGBN also referred to a scalar version of a markov chain. The kernel is defined by

$$k(t,t') = \sigma_X^2 \beta_X^{|t-t'|} \frac{1-\beta_X^{2\min(t,t')}}{1-\beta_X^2}. \tag{12}$$

This paper will build upon the scalar case and generalize it for the multidimensional DGBN.

4 Gaussian Processes for Dynamic Gaussian Bayesian Networks

We have $X^{(1)},...,X^{(N)}$ random variables in the DGBN evolving over time $t = 1,...,\tilde{T}$ (we use $\tilde{T}$ to avoid confusion with the matrix transpose), where N is the number of dimensions and $\tilde{T}$ the number of time steps in a DGBN. As Alvarez et al. [1] described, multidimensional kernels follow the form $K(D_t, D'_{t'})$, where D and D' are dimensions of the underlying model. We will develop a kernel function that has an $N \times X$ dimensional matrix as an output containing all covariances between random variables in time steps t and t'

$$K(t,t') = \begin{bmatrix} K(X_t^{(1)}, X_{t'}^{(1)}) & \dots & K(X_t^{(1)}, X_{t'}^{(N)}) \\ \vdots & \ddots & \vdots \\ K(X_t^{(N)}, X_{t'}^{(1)}) & \dots & K(X_t^{(N)}, X_{t'}^{(N)}) \end{bmatrix}. \tag{13}$$

4.1 Constructing the GP

Shachter and Kenley [17] have developed an algorithm to convert a Gaussian Bayesian network into a multivariate Gaussian distribution. To prove correctness, they formulated the following Lemma that we will reuse.

Lemma 1. *For $G \in \mathbb{N}$ topological ordered random variables $X^{(i)} \in \mathbb{X}$, $i = 1,\dots,G$ in a Gaussian Bayesian network let σ_i^2 be the variance of the conditional distribution of X^i given its parents. Let $\mathbf{B} \in \mathbb{R}^{G\times G}$ be a matrix, where the entries $\beta_{i,l}$, $l = 1,...,G$ describe the linear relationship between a child $X^{(i)}$ and its parent $X^{(l)}$. If $X^{(l)}$ is no parent of $X^{(i)}$ the entry is zero. For a fixed $j \in \{1,\dots,G\}$ let $\mathbf{\Sigma}_{qq}$ be the covariance matrix between all random variables $X^{(q)}$, $q = 1,...,j$ and $\mathbf{B}_{sj} \in \mathbb{R}^{j-1\times 1}$, $s = 1,\dots,j-1$ the corresponding part of $\mathbf{B}$. We denote the matrices*

$$\mathbf{S}_j := \begin{bmatrix} \Sigma_{tt} & 0 & \dots & 0 \\ 0 & \sigma_{j+1}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \sigma_G^2 \end{bmatrix}, \tag{14}$$

$$\mathbf{U}_j := \begin{bmatrix} \mathbf{I}_{j-1} & \mathbf{B}_{sj} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{G-j} \end{bmatrix}. \tag{15}$$

Then it is

$$\mathbf{S}_j = \mathbf{U}_{j-1}^T \mathbf{S}_{j-1} \mathbf{U}_{j-1} \tag{16}$$

and

$$\mathbf{\Sigma} = \mathbf{S}_G = \mathbf{U}_G^T ... \mathbf{U}_1^T \mathbf{S}_0 \mathbf{U}_1 ... \mathbf{U}_G, \tag{17}$$

where $\mathbf{\Sigma} \in \mathbb{R}^{G \times G}$ *is the covariance matrix of the equivalent multivariate Gaussian distribution for the above defined Gaussian Bayesian network.*

The $N \times N$ covariance matrix from Equation 17 is calculated by recursively multiplying the $\mathbf{U}$-matrices. To define a GP we do not want to calculate a full covariance matrix as it is done in Lemma 1, but we need a kernel function mapping arbitrary time points t and t' to a covariance value or in our case covariance matrix as defined in Equation 13. Therefore we convert the recursive multiplication of the matrices in Equation 17 into closed form kernel function.

If we look at a certain number of time steps $\tilde{T}$, we have a number of total nodes in our network of $G = \tilde{T}N$. The matrix S_0 is diagonal with the individual variances for each of the G individual nodes. In the respective network. To ensure a topological ordering for our DGBN, we position all variables belonging to a time step t before all variables of $t + 1$. The order within a time step is irrelevant because there are no relations within a time step. For the sake of simplicity we order the variables within a time step based on their indexing $X^{(1)}, ..., X^{(N)}$. Figure 3 contains a visualization for the structure of the resulting covariance matrix. As shown, the kernel function can flexibly generate parts of a full covariance matrix.

Since the σ^2-values for all random variables stay constant over time, the matrix $\mathbf{S}_0$ has repeating diagonal entries every N entries. We denote $\mathbf{A}$ for the $N \times N$ block that is on the diagonal of $\mathbf{S}_0$, which itself is constructed by

$$\mathbf{A} = diag(\sigma^2_{X^{(1)}}, ..., \sigma^2_{X^{(N)}}). \tag{18}$$

The $G \times G$-dimensional matrix B from Lemma 1 containing all linear relationships in the DGBN has the $\tilde{T} \times \tilde{T}$-dimensional block structure, where $\mathbf{M}$ is the $N \times N$ transition matrix, resulting in

$$B = \begin{bmatrix} \mathbf{0} & \mathbf{M} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{M} \end{bmatrix}, \tag{19}$$

having the transition matrix M at all block positions $(t, t+1)$. Given this structure, we can reformulate Equation 17 from Lemma 1. With N dimensions we can multiply N consecutive matrices from U_t to U_{t+N} that would belong to the

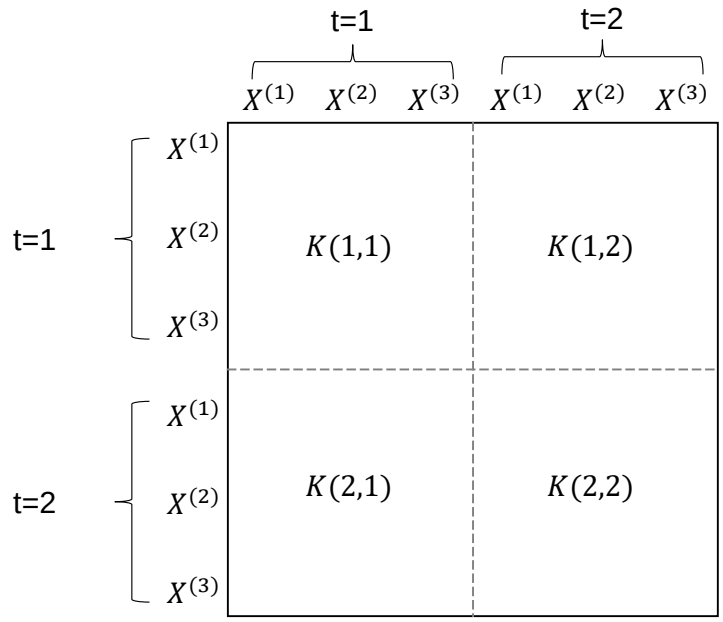

Fig. 3. Visual structure of the resulting covariance matrix

random variables within one time step.

$$\mathbf{O}_t = \prod_{i=t}^{t+N} \mathbf{U}_i = \begin{bmatrix} \mathbf{I}_{(t-2)N} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_N & \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{(T-t)N} \end{bmatrix}. \tag{20}$$

With block matrix multiplication, and the construction of matrices $\mathbf{O}_t$ we can reformulate the multiplication from Lemma 1 into

$$\prod_{i=1}^{G} U_i = \prod_{t=1}^{T} O_t = \begin{bmatrix} \mathbf{I} & \mathbf{M} & \mathbf{M}^2 & \dots & \mathbf{M}^{\tilde{T}} \\ \mathbf{0} & \mathbf{I} & \mathbf{M} & \dots & \mathbf{M}^{\tilde{T}-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \dots & \mathbf{M}^{\tilde{T}-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} \end{bmatrix}. \tag{21}$$

The full $G \times G$ covariance matrix would be calculated by using Equation 17. In our kernel function we only want to calculate the $N \times N$ matrix containing the covariances between two time steps t and t'. We would get this matrix by multiplying the t-th row of blocks from $\mathbf{U}_G^T ... \mathbf{U}_1^T$, with the t'-th column of $\mathbf{S}_0 \mathbf{U}_1 ... \mathbf{U}_G$. If $t = t'$ we have

$$\begin{bmatrix} \mathbf{M}^{t-1} \\ \mathbf{M}^{t-2} \\ \mathbf{M}^{t-3} \\ \vdots \\ \mathbf{M}^{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}^{T} \cdot \begin{bmatrix} \mathbf{A}\mathbf{M}^{t-1} \\ \mathbf{A}\mathbf{M}^{t-2} \\ \mathbf{A}\mathbf{M}^{t-3} \\ \vdots \\ \mathbf{A}\mathbf{M}^{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{A0} \end{bmatrix} = \sum_{k=0}^{t} \left(\mathbf{M}^{T}\right)^{k} \mathbf{A}\mathbf{M}^{k}. \tag{22}$$

In the case $t \neq t'$ we look at $t < t'$, because of symmetry same conclusions can be made when setting $t < t'$. In our case, the t-th row of blocks contains more blocks and also blocks with higher exponential. Because of the all-zero matrices, only the first t blocks are relevant, resulting in

$$\begin{bmatrix} \mathbf{M}^{t-1} \\ \mathbf{M}^{t-2} \\ \vdots \\ \mathbf{M}^{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{A}\mathbf{M}^{t'-1} \\ \mathbf{A}\mathbf{M}^{t'-2} \\ \vdots \\ \mathbf{A}\mathbf{M}^{t'-t} \end{bmatrix}^{T} = \sum_{k=0}^{t} \left(\mathbf{M}^{T}\right)^{k} \mathbf{A}\mathbf{M}^{k+(t'-t)} = \left(\sum_{k=0}^{t} \left(\mathbf{M}^{T}\right)^{k} \mathbf{A}\mathbf{M}^{k}\right) \mathbf{M}^{t'-t} \tag{23}$$

For the case $t > t'$, the resulting matrix needs to be the transposed version of the previous case, which can be proven by

$$\begin{bmatrix} \mathbf{M}^{t-1} \\ \mathbf{M}^{t-2} \\ \vdots \\ \mathbf{M}^{t-t'} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{A}\mathbf{M}^{t'-1} \\ \mathbf{A}\mathbf{M}^{t'-2} \\ \vdots \\ \mathbf{A}\mathbf{M}^{0} \end{bmatrix}^{T} = \sum_{k=0}^{t'} \left(\mathbf{M}^{T}\right)^{k+(t-t')} \mathbf{A}\mathbf{M}^{k} = \mathbf{M}^{T^{t'-t}} \sum_{k=0}^{t'} \left(\mathbf{M}^{T}\right)^{k} \mathbf{A}\mathbf{M}^{k}. \tag{24}$$

Resulting in a kernel function of

$$K(t,t') = \begin{cases} \left(\sum_{i=0}^{min(t,t')} \mathbf{M}^{T}\mathbf{A}\mathbf{M}\right) \mathbf{M}^{|t-t'|}, t \leq t', \\ \left(\left(\sum_{i=0}^{min(t,t')} \mathbf{M}^{T}\mathbf{A}\mathbf{M}\right) \mathbf{M}^{|t-t'|}\right)^{T}, t > t'. \end{cases} \tag{25}$$

As mentioned in Section 2.2, the GP is defined by its kernel or covariance function and the mean function. We have defined the covariance function in Equation 25. The mean function is time independent and is simply a constant mean vector defined by the DGBN, where each random variable $X^{(i)}$ has a mean value $\mu_{X^{(i)}}$, resulting in

$$m(t) = \mu \tag{26}$$

4.2 Continuity Discussion

In general, a GP is defined over a continuous scale. Having a continuous scale would be a benefit compared to the discrete DGBN. For a GP to be continuous the kernel need to be defined for $t, t' \in \mathbb{R}$. The two key issues are that neither the summation term nor the exponential of the matrix $\mathbf{M}$ are necessarily uniquely defined defined for $t \in \mathbb{R}$. The one-dimensional kernel function for the scalar case from Equation 12 solves this issue by converting the summation in a continuous defined partial sum of a geometric series [6, 10]

In this paper we will keep the time-scale discrete but we discuss a few ideas how to generalize the kernel for a continuous case. The summation is dependent on the min value of t and t', meaning that if the smaller of the two values is a natural number, the summation is defined. In realtiy for working with the GP this means that if we would like to forecast one moment in the future t_f, that moment could be a real number. All other evidence points in the past would need to be discrete. Müller and Schleicher [10] have discussed specific fractional sums but a full mathematical consideration is not in the scope of this paper. The exponent $|t - t|'$ is real value if t or t' are real numbers. For rational exponents the result is defined by the n-th root

$$\mathbf{M}^{(\frac{q}{d})} = \sqrt[d]{\mathbf{M}^q}. \tag{27}$$

The n-th root can have none, exactly one or multiple solutions, depending on the structure of $\mathbf{M}$. A full continuous definition of the GP would need to handle the cases where there is no exact solution or put further restrictions on the transition matrix $\mathbf{M}$.

4.3 Kernel Properties

To be a valid kernel, the kernel needs to be symmetric which directly follows from Equation 13 and Equation 24. Additionally, the kernel needs to result in a positive semidefinite covariance matrix. In Section 2.3 we introduced that kernels can be created of other valid kernels. The $min(t, t')$ and the $a^{|t-t'|}$ terms are valid kernels. Also a summation of valid kernels is a valid kernel. Since the matrix $\mathbf{M}$ only contains constant values, using the resulting kernel function from Equation 25 results in a symmetric and positive semidefinite covariance matrix and is therefore valid for a GP.

5 Discussion and Outlook

In this paper we encoded a multidimensional DGBN with arbitrary connections between time steps into a Gaussian Process. We demonstrate the generalizing power of GPs by converting a already very general PGM into a GP. All that is needed, is the correct kernel function to describe relationships between random variables along the time dimension. The contribution of the paper has impact on the theoretical research in the fields. Bringing together different research streams

and the underlying concepts can benefit both research areas. Existing methods from one area can be possibly transferred to the other research stream and enhance existing applications and vice versa. Also further research can be better directed because scholars in different research groups can work closer together. The results of the paper also bring practical benefits:

Efficient Query Answering: In a DGBN the evidence is usually propagated through the model along the time dimension. In the constructed GP the kernel allows us to explicitly define the effect of an observation to any other queried point in time which can speeds up the querying answering process

Markov Property: The defined kernels keep the Markov property and the transition behavior of the underlying model.

Kernel Combination: The created kernel can be combined with any other existing kernels. If the real-world phenomenon is relatively well described by the DGBN-kernel but also contains a slight periodic behavior, both kernels can easily combined by different operations, e.g. addition and multiplication [13].

There are three streams for further research. First, even further generalize the DGBN and allow also intra-time slice connections. Second, conduct mathematical deep-dives to understand the circumstances under which the kernel can be used in a continuous time setting. Third, transfer real-world applications previously using DGBN into GPs and evaluate query answering time and potential model enhancements by combining kernels.

References

1. Alvarez, M.A., Rosasco, L., Lawrence, N.D., Others: Kernels for vector-valued functions: A review. Foundations and Trends®in Machine Learning **4**(3), 195–266 (2012)
2. Bishop, C.M.: Pattern recognition and machine learning. Springer (2006)
3. Constantinou, A.C., Fenton, N., Neil, M.: Integrating expert knowledge with data in Bayesian networks: Preserving data-driven expectations when the expert variables remain unobserved. Expert Systems with Applications **56**, 197–208 (sep 2016). https://doi.org/10.1016/j.eswa.2016.02.050
4. Flores, M.J., Nicholson, A.E., Brunskill, A., Korb, K.B., Mascaro, S.: Incorporating expert knowledge when learning Bayesian network structure: a medical case study. Artificial intelligence in medicine **53**(3), 181–204 (2011)
5. Frigola-Alcalde, R.: Bayesian time series learning with Gaussian processes. Ph.D. thesis, University of Cambridge (2016)
6. Hartwig, M., Mohr, M., Möller, R.: Constructing Gaussian Processes for Probabilistic Graphical Models. In: The Thirty-Third International Flairs Conference (2020)
7. Koller, D., Friedman, N., Bach, F.: Probabilistic graphical models: principles and techniques. MIT press (2009)
8. Komurlu, C., Bilgic, M.: Active inference and dynamic gaussian bayesian networks for battery optimization in wireless sensor networks. In: Workshops at the Thirtieth AAAI Conference on Artificial Intelligence. Citeseer (2016)
9. McCann, R.K., Marcot, B.G., Ellis, R.: Bayesian belief networks: applications in ecology and natural resource management. Canadian Journal of Forest Research **36**(12), 3053–3062 (2006)

10. Müller, M., Schleicher, D.: How to add a non-integer number of terms, and how to produce unusual infinite summations. Journal of computational and applied mathematics **178**(1-2), 347–360 (2005)
11. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. thesis (2002)
12. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1988)
13. Rasmussen, C.E.: Gaussian processes for machine learning. MIT Press (2006)
14. Reece, S., Roberts, S.: An introduction to Gaussian processes for the Kalman filter expert. In: 2010 13th International Conference on Information Fusion. pp. 1–9. IEEE (2010)
15. Reece, S., Roberts, S.: The near constant acceleration Gaussian process kernel for tracking. IEEE Signal Processing Letters **17**(8), 707–710 (2010)
16. Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., Aigrain, S.: Gaussian processes for time-series modelling. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **371**(1984), 20110550 (2013)
17. Shachter, R.D., Kenley, C.R.: Gaussian Influence Diagrams. Management Science **35**(5), 527–550 (1989). https://doi.org/10.1287/mnsc.35.5.527
18. Turner, R.D.: Gaussian processes for state space models and change point detection. Ph.D. thesis, University of Cambridge (2012)
19. Xu, Z., Kersting, K., Tresp, V.: Multi-relational learning with gaussian processes. In: Twenty-First International Joint Conference on Artificial Intelligence (2009)
20. Yu, K., Chu, W., Yu, S., Tresp, V., Xu, Z.: Stochastic relational models for discriminative link prediction. In: Advances in neural information processing systems. pp. 1553–1560 (2007)