Deep Gaussian Processes: Theory and Applications

Petar M. Djurić

Bellairs Workshop, Barbados

February 13, 2019

Stony Brook University

Outline

- Introduction
- Gaussian processes
- Deep Gaussian processes
- Applications
- Conclusions

Introduction

- Probabilistic modeling allows for representing and modifying uncertainty about models and predictions.
- This is done according to well defined rules.
- Probabilistic modeling has a central role in machine learning, cognitive science and artificial intelligence.

The Concept of Uncertainty

- Learning and intelligence depend on the amount of uncertainty in the information extracted from data.
- Probability theory is the main framework for handling uncertainty.
- Interestingly, in the recent progress of deep learning with deep neural networks, which are based on learning from huge amounts of data, the concept of uncertainty is somewhat bypassed.
- In the years to come, we will see further advances in artificial intelligence and machine learning within the probabilistic framework.

The Role of a Model

- To make inference from data, one needs models.
- Models can be simple (like linear models) or highly complex (like large and deep neural networks).
- In most settings, the models must be able to make predictions.
- Uncertainty plays a fundamental role in modeling observed data and in interpreting model parameters, the results of models, and the correctness of models.

The Learning

- Probability distributions are used to represent uncertainty.
- Learning from data occurs by transforming prior distributions (defined before seeing the data) to posterior distributions (after seeing the data).
- The optimal transformation from information-theoretic point of view is the Bayes rule.
- The beauty of the approach is the simplicity of the Bayes mechanism.

Gaussian Processes Regression

- Essentially, a GP can be seen as the distribution of a real-valued function $f(\mathbf{x})$,

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k_f(\mathbf{x}_i, \mathbf{x}_j))$$

- Some assumptions are often made when using GP regression

1. the mean function $m(\mathbf{x}) = 0$ for simplicity, and
2. the observation noise is additive white Gaussian noise for tractability.

Gaussian Processes Regression (contd.)

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and $\mathbf{y}$ denote the collection of all input vectors and all observations, respectively, with the above assumptions, i.e.,

$$\mathbf{y} = \mathbf{f}(\mathbf{X}) + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$. We also have

- Likelihood: $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma_\epsilon^2 \mathbf{I})$, and
- Prior: $p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_{ff})$, where $\mathbf{K}_{ff} = \mathbf{k}_f(\mathbf{X}, \mathbf{X})$ and $\boldsymbol{\theta}$ denote the hyper-parameters in the covariance function.

Gaussian Processes Regression (contd.)

The hyper-parameters $\boldsymbol{\theta}$ can be learned from the training data $\{\mathbf{X}, \mathbf{y}\}$ by maximizing the log-marginal-likelihood

- Log-marginal-likelihood: $\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})$

$$
\begin{aligned}
\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) &= \log \mathcal{N}(\mathbf{y}|\mathbf{0},\ \mathbf{K}_{ff} + \sigma_\epsilon^2 \mathbf{I}) \\
&= \log \mathcal{N}(\mathbf{y}|\mathbf{0},\ \mathbf{K}) \\
&= -\frac{1}{2}\mathbf{y}^T \mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{N}{2}\log 2\pi
\end{aligned}
$$

- The Occam's razor is embedded in the model.

Stony Brook University

Gaussian Processes Regression (contd.)

Let $\mathbf{X}_*$ and $\mathbf{f}_*$ denote the collection of test inputs and the corresponding latent function values, respectively. Then we can express the predictive posterior as

- Predictive posterior: $p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}_*|\mathbb{E}(\mathbf{f}_*), cov(\mathbf{f}_*))$

$$\mathbb{E}(\mathbf{f}_*) = [\mathbf{K}_f(\mathbf{X}_*, \mathbf{X})]\mathbf{K}^{-1}\mathbf{y}$$

$$cov(\mathbf{f}_*) = \mathbf{K}_f(\mathbf{X}_*, \mathbf{X}_*) - [\mathbf{K}_f(\mathbf{X}_*, \mathbf{X})]\mathbf{K}^{-1}[\mathbf{K}_f(\mathbf{X}_*, \mathbf{X})]^T$$

Stony Brook University

Covariance Function

- For example: Radial basis function (RBF) or squared exponential (SE)

  One dimensional form:

$$k_{rbf}(x_i, x_j) = \sigma_f^2 \exp(-\frac{1}{\ell}(x_i - x_j)^2)$$

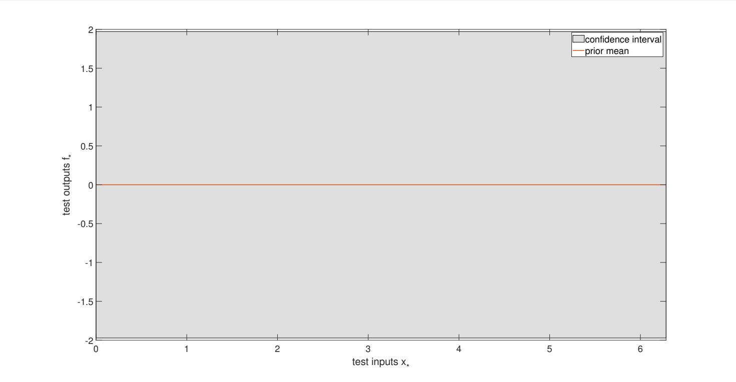
- σ_f^2 measures strength of signal, $\frac{\sigma_f^2}{\sigma_\epsilon^2}$ is equivalent to signal-to-noise ratio (SNR).
- The characteristic length scale ℓ encodes the model complexity in that dimension.
- $r = \frac{1}{\ell}$ measures the relevance of that dimension.
- Automatic relevance determination (ARD)

Toy Example

- Goal: learn $f(x)$ from 5 noisy observations $\{x_i, y_i\}_{i=1}^{5}$.
- Ground truth: $y = \sin(x) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$.
- Test inputs: $\mathbf{x}_* \in \mathbb{R}^{300\times 1}$ equally spaced from $x = 0$ to 2π.
- Test outputs: $\mathbf{f}_* = f(\mathbf{x}_*)$

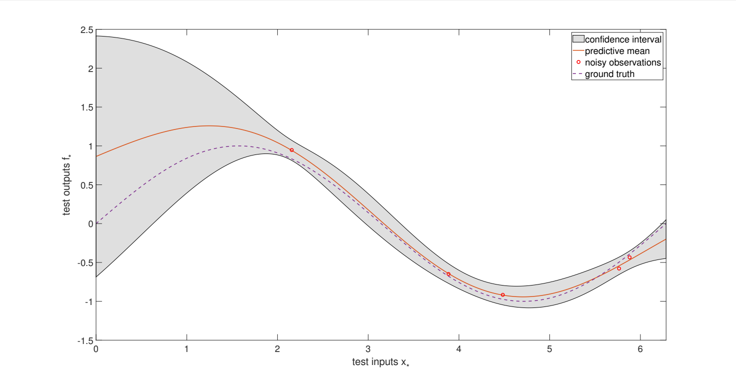
Stony Brook University

Prior Distribution

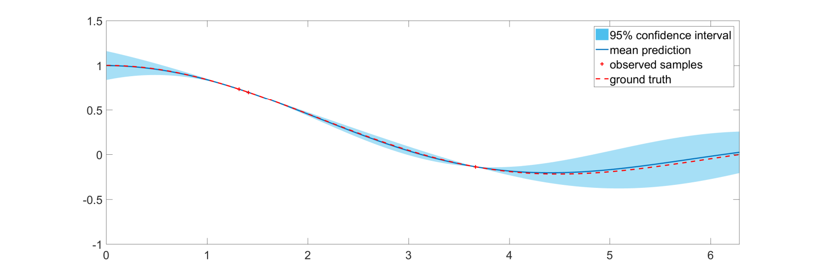


Stony Brook University

Predictive (Posterior) Distribution



Another Toy Example: The Function $\sin x/x$



Stony Brook University

Example: Recovery of Missing Samples in FHR[1]

- Goal: recover missing samples in FHR, using not only observed FHR but also UA samples

- Model:

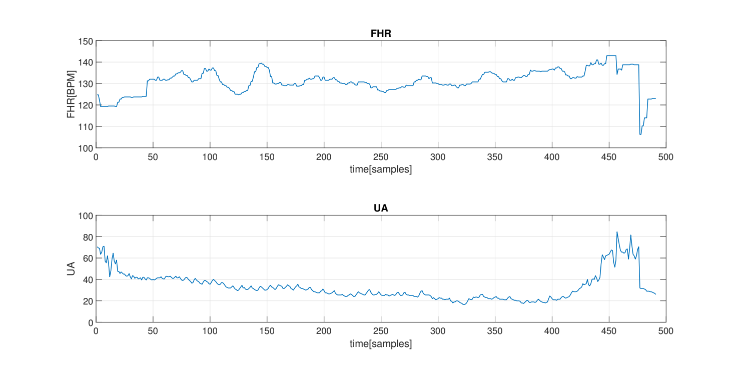
$$y_i = y(\mathbf{x}_i) = f(\mathbf{x}_i) + \epsilon_i$$

  - y_i: i-th sample in an FHR segment
  - $\mathbf{x}_i = [i, u_i]'$ where u_i is the i-th UA sample
  - ϵ_i: Gaussian white noise
  - $f(\mathbf{x}_i)$: i-th latent noise-free FHR sample

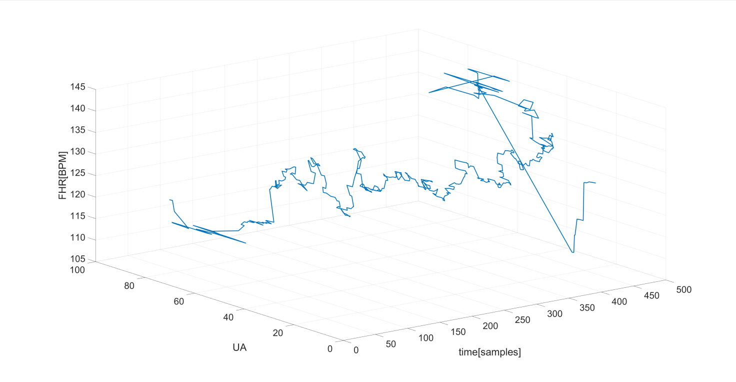
[1] Guanchao Feng, J Gerald Quirk, and Petar M Djurić. "Recovery of missing samples in fetal heart rate recordings with Gaussian processes". In: *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE. 2017, pp. 261–265.

Stony Brook University

CTG Segment for Experiments

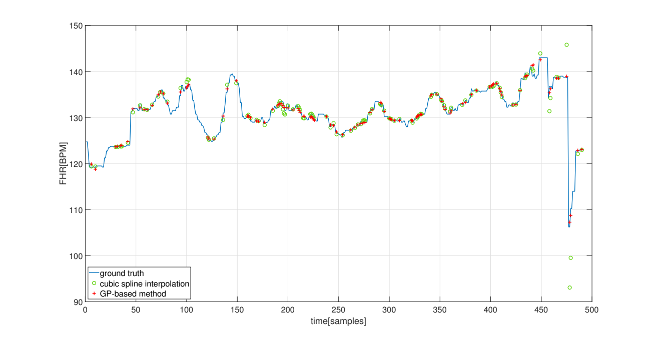


CTG Segment for Experiments



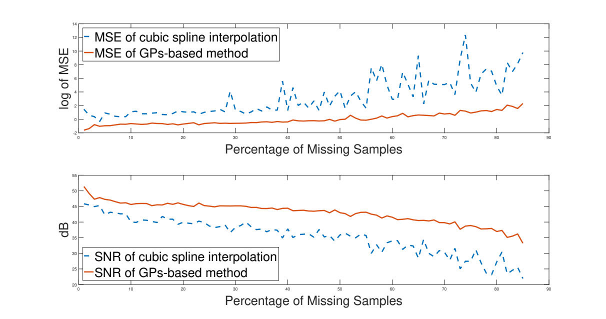
Experiment I

- 120 missing samples were randomly selected, and we tried to recover their true values.



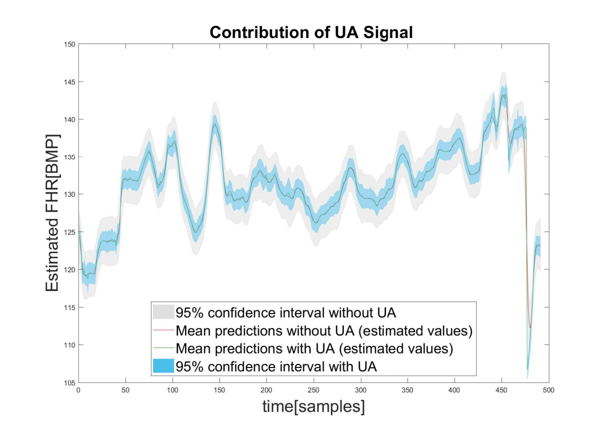
Experiment II

- The percentage of missing samples was increased from 1% to 85% with a step size of 1%.



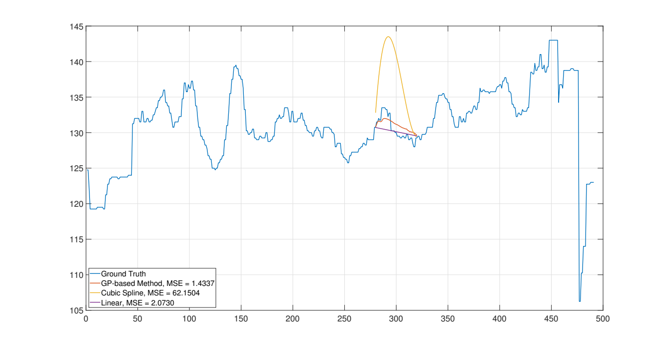
Experiment III

- To demonstrate contribution of UA, we repeated the experiment I, but excluded u_i from the input vector $\mathbf{x}_i$.



Experiment VI (An Extreme Case)

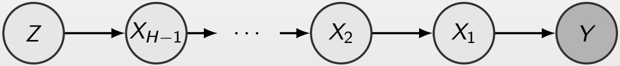
- 10 seconds of consecutive missing samples.



Limitations

- The general framework is computationally expensive, $O(N^3)$, due to the term $\mathbf{K}_{N \times N}^{-1}$.
- Another limitation is the joint Gaussianity that is required by the definition of GPs.

Deep Gaussian Processes



- $\mathbf{Y} \in \mathbb{R}^{N \times d_y}$: observations, output of the network
  - N is the number of observation vectors.
  - d_y is the dimension of the vectors $\mathbf{y}_n$.

- $\{\mathbf{X}_h\}_{h=1}^{H-1}$: intermediate latent states
  - dimensions $\{d_h\}_{h=1}^{H-1}$ are potentially different.

- $\mathbf{Z} \in \mathbb{R}^{N \times d_z}$: the input to the network
  - $\mathbf{Z}$ is observed for supervised learning.
  - $\mathbf{Z}$ is unobserved for unsupervised learning.

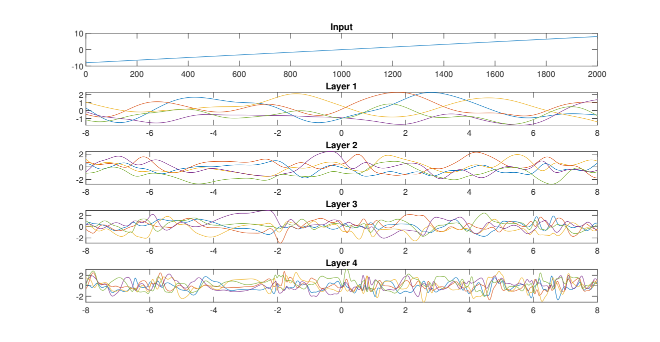
Deep Gaussian Processes (contd.)

- The joint Gaussianity limitation is overcome because nonlinear mappings generally will not preserve Gaussianity.
- DGPs immediately introduce intractabilities.
- One way of handling the difficulties is by introducing a set of inducing points and where within the variational framework, sparsity and a tractable lower bound on the marginal likelihood are obtained.

Stony Brook University

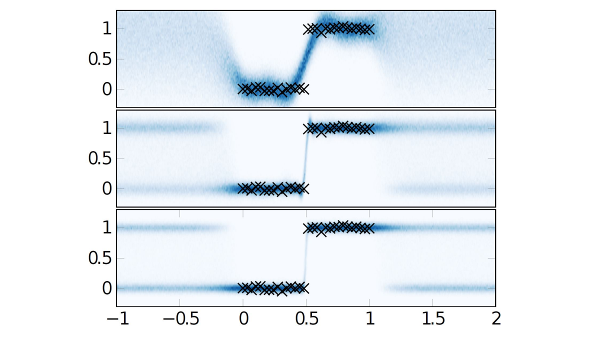
Example: Functions Sampled From DGP

- Gaussianity limitation is overcome by nonlinear function composition.



Example: Learning a Step Function[2]

- Standard GP (top), two- and four-layer DGP (middle, bottom).
- DGPs achieved much better performance.



[2] James Hensman and Neil D Lawrence. "Nested variational compression in deep Gaussian processes". In: *arXiv preprint arXiv:1412.1370* (2014).

Deep GPs and Deep Neural Networks (a comparison)

- A single layer of fully connected neural network with an independent and identically distributed (iid) prior over its parameters and with an infinite width is equivalent to a GP.
- Therefore, deep GPs are equivalent to neural networks with multiple, infinitely wide hidden layers.
- Mappings of a DGP are governed by its GPs instead of activation functions.
- A DGP allows for propagations and quantifications of uncertainties through each layer as a fully Bayesian probabilistic model.
- There is ARD at each layer.

Generative Process

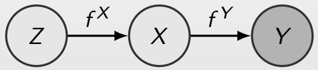


Figure: A two-layer DGP.

- The generative process takes the form:

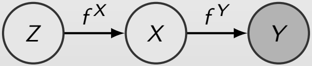
$$\begin{aligned} x_{nl} &= f_l^X(\mathbf{z}_n) + \epsilon_{nl}^X, \quad l = 1, \ldots, d_x, \quad \mathbf{z}_n \in \mathbb{R}^{d_Z} \\ y_{ni} &= f_i^Y(\mathbf{x}_n) + \epsilon_{ni}^Y, \quad i = 1, \ldots, d_y, \quad \mathbf{x}_n \in \mathbb{R}^{d_x} \end{aligned}$$

- ϵ_{nl}^X and ϵ_{ni}^Y are additive white Gaussian processes.

Generative Process (contd.)

- We assume $\mathbf{Z}$ is unobserved with a prior $p(\mathbf{Z}) = \mathcal{N}(\mathbf{Z}|\mathbf{0}, \mathbf{I})$
- If we have specific prior knowledge about $\mathbf{Z}$, we should quantify this knowledge into a prior accordingly.

Inference



- The inference takes the reverse route, i.e., we observe high-dimensional data **Y**, and we learn the low-dimensional manifold **Z** (of dimension d_z, where $d_z < d_x < d_y$) that is responsible for generating **Y**.

Inference Challenges

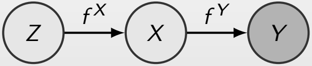
The learning requires maximization of the log-marginal-likelihood,

$$\log p(\mathbf{Y}) = \log \int_{\mathbf{X},\mathbf{Z}} p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})\mathrm{d}\mathbf{X}\mathrm{d}\mathbf{Z}$$

which is intractable.

Stony Brook University

Augmentation of Probability Space



- Original probability space:

$$p(\mathbf{Y}, \mathbf{F}^Y, \mathbf{F}^X, \mathbf{X}, \mathbf{Z}) = p(\mathbf{Y}|\mathbf{F}^Y)p(\mathbf{F}^Y|\mathbf{X})p(\mathbf{X}|\mathbf{F}^X) \times p(\mathbf{F}^X|\mathbf{Z})p(\mathbf{Z})$$

- Augmentation using inducing points:
  - $\mathbf{U}^X = f^X(\tilde{\mathbf{Z}})$, $\tilde{\mathbf{Z}} \in \mathbb{R}^{N_p \times d_Z}$ and $\mathbf{U}^X \in \mathbb{R}^{N_p \times d_x}$
  - $\mathbf{U}^Y = f^Y(\tilde{\mathbf{X}})$, $\tilde{\mathbf{X}} \in \mathbb{R}^{N_p \times d_x}$ and $\tilde{\mathbf{X}} \in \mathbb{R}^{N_p \times d_x}$
  - $N_p \leq N$

Augmentation of Probability Space

- Augmented probability space:

$$\begin{aligned} & p(\mathbf{Y}, \mathbf{F}^Y, \mathbf{F}^X, \mathbf{X}, \mathbf{Z}, \mathbf{U}^Y, \mathbf{U}^X, \tilde{\mathbf{X}}, \tilde{\mathbf{Z}}) \\ & = p(\mathbf{Y}|\mathbf{F}^Y) p(\mathbf{F}^Y|\mathbf{U}^Y, \mathbf{X}) p(\mathbf{U}^Y|\tilde{\mathbf{X}}) \\ & \times p(\mathbf{X}|\mathbf{F}^X) p(\mathbf{F}^X|\mathbf{U}^X, \mathbf{Z}) p(\mathbf{U}^X|\tilde{\mathbf{Z}}) p(\mathbf{Z}) \end{aligned}$$

- Problematic terms:
  - $\mathcal{A} = p(\mathbf{F}^Y|\mathbf{U}^Y, \mathbf{X})$
  - $\mathcal{B} = p(\mathbf{F}^X|\mathbf{U}^X, \mathbf{Z})$

Stony Brook University

Variational Inference

- A variational distribution: $\mathcal{Q} = q(\mathbf{U}^Y)q(\mathbf{X})q(\mathbf{U}^X)q(\mathbf{Z})$
- By Jensen's inequality:

$$\log p(\mathbf{Y}) \geq \mathcal{F}_v = \int \mathcal{Q} \cdot \mathcal{A} \cdot \mathcal{B} \log \mathcal{G} \, \mathrm{d}\mathbf{F}^Y \mathrm{d}\mathbf{X} \mathrm{d}\mathbf{F}^X \mathrm{d}\mathbf{Z} \mathrm{d}\mathbf{U}^X \mathrm{d}\mathbf{U}^Y$$

- The function $\mathcal{G}$ is defined as:

$$\mathcal{G}(\mathbf{Y}, \mathbf{F}^Y, \mathbf{X}, \mathbf{F}^X, \mathbf{Z}, \mathbf{U}^X, \mathbf{U}^Y) = \frac{p(\mathbf{Y}|\mathbf{F}^Y)p(\mathbf{U}^Y)p(\mathbf{X}|\mathbf{F}^X)p(\mathbf{U}^X)p(\mathbf{Z})}{\mathcal{Q}}.$$

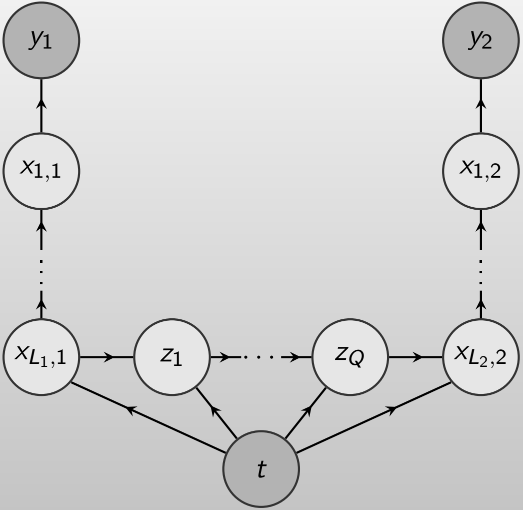
- $\mathcal{F}_v$ is tractable for a collection of covariance functions, since $\mathcal{A}$ and $\mathcal{B}$ are canceled out in $\mathcal{G}$.

Studying Complex Systems

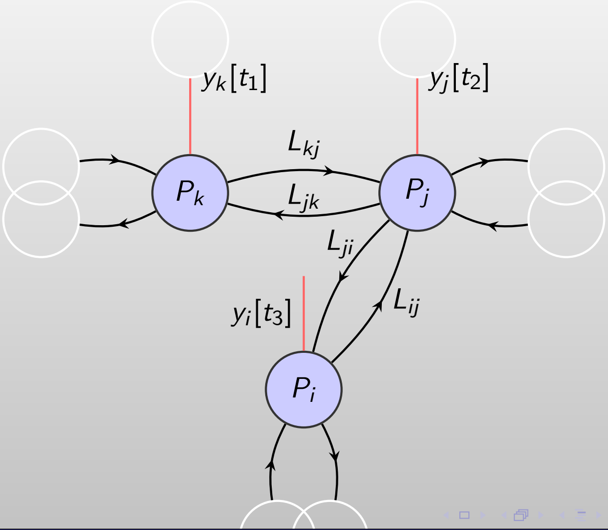
Used principles

- algorithmic compressibility,
- locality, and
- deep probabilistic modeling.

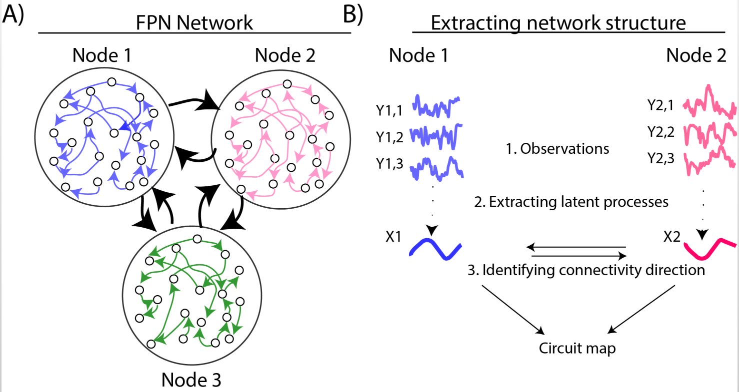
Applications



Applications-contd.



Applications-contd.[3]



[3] Figures obtained by Sima Mofakham and Chuck Mikell.

Example: Binary pH-based Classification[4]

- Goal: to have the DGP classify CTG recordings into health and unhealthy classes.
- Features:
  - 14 FHR features
  - 6 (categorical) UA features
- Labeling:
  - Positive (unhealthy): pH < 7.1
  - Negative (healthy): pH > 7.2

[4] Guanchao Feng, J Gerald Quirk, and Petar M Djurić. "Supervised and Unsupervised Learning of Fetal Heart Rate Tracings with Deep Gaussian Processes". In: *2018 14th Symposium on Neural Networks and Applications (NEUREL)*. IEEE. 2018, pp. 1–6.

- Structure of DGP: our DGP network had two layers, and in each layer, we set the initial latent dimension to five.
- Performance metrics:
  1. Sensitivity (true positive rate)
  2. Specificity (true negative rate)
  3. Geometric mean of specificity and sensitivity

Features

Table: Features for FHR

Category	Feature
Time domain	Mean, Standard deviation, STV, STI, LTV, LTI
Non-linear	Poincaré SD1, Poincaré SD2, CCM
Frequency domain	VLF, LF, MF, HF, ratio

Table: Features for UA

	Normal (0)	Abnormal (1)
Frequency	$\leq$ 8 contractions	$>$ 8 UC (tachysystole)
Duration	$<$ 90s	$>$ 90s
Increased tonus	With toco	Prolonged $>$ 120s
Interval A	Interval – peak to peak	$<$ 2min
Interval B	Interval – offset of UC to onset of next UC	$<$ 1min
Rest time	$>$ 50%	$<$ 50%

Stony Brook University

Classification Results

- Support vector machine (SVM) was used as a benchmarking model.

Table: Classification results

Classifier	Feature	Specificity	Sensitivity	Geometric Mean
SVM	FHR	0.82	0.73	0.77
	FHR+UA	0.82	0.82	0.82
Deep GP	FHR	0.91	0.73	0.82
	FHR+UA	0.82	0.91	0.86

Stony Brook University

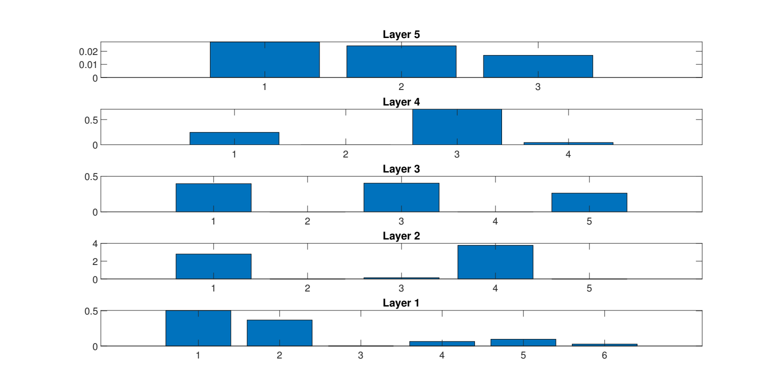
Unsupervised Learning for FHR Recordings

- Goal: to have the DGP learn informative low-dimensional latent spaces that can generate the recordings.
- Labeling:
  - pH-based labeling combined with obstetrician's evaluation.
  - Labels are only used for evaluation of learning results.
- Data:
  - The last 30 minutes of 10 FHR recordings, $\mathbf{Y} \in \mathbb{R}^{10\times 7200}$.
  - Three of them are abnormal and 7 are normal.

Performance Metric and Network Structure

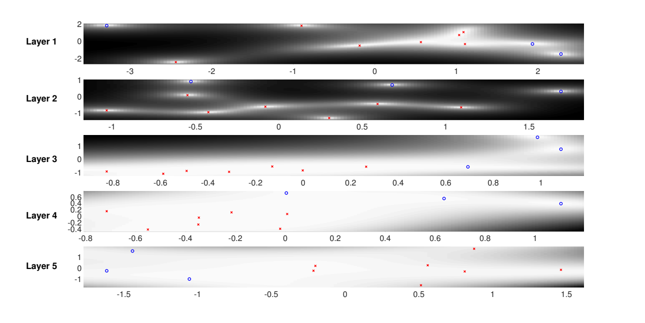
- Performance metric: the number of errors in the latent space for one nearest neighbor.
- Structure of DGP: a five-layer DGP, and the initial dimensions of the latent spaces in the layers were $d_{x_{1:5}} = [6, 5, 5, 4, 3]^T$.

Automatic Structure Learning



Visualization of the Latent Spaces with 2-D Projection.

- Red: the normal recordings
- Blue: the abnormal recordings
- Pixel intensity: proportional to precision
- The total errors in layers 1 to 5 are 2, 2, 1, 1, 0, respectively.



Example: Deep Gaussian Processes with Convolutional Kernels[5]

- Goal: multi-class image classification
- Database: MNIST (handwritten digits)
- Methods:
  1. SGP: Sparse Gaussian processes
  2. DGP: Deep Gaussian processes
  3. CGP: Convolutional Gaussian processes
  4. CDGP: Convolutional deep Gaussian processes

Stony Brook University

[5] Vinayak Kumar et al. "Deep Gaussian Processes with Convolutional Kernels". In: *arXiv preprint arXiv:1806.01655* (2018).

MNIST

Model	Layer 1	Layer 2	Layer 3	Layer 4	Accuracy%	NLPP
SGP	RBF	–	–	–	97.48	–
DGP1	RBF	RBF	–	–	97.94	0.073
DGP2	RBF	RBF	RBF	–	97.99	0.070
CGP1	Conv	–	–	–	95.59	0.170
CGP2	Wconv	–	–	–	97.54	0.103
CDGP1	Wconv	RBF	–	–	**98.66**	**0.046**
CDGP2	Conv	RBF	–	–	98.53	0.536
CDGP3	Conv	RBF	RBF	–	98.40	0.055
CDGP4	Conv	RBF	RBF	RBF	98.41	0.051
CDGP5	Wconv	Wconv	RBF	–	98.44	0.048
CDGP6	Wconv	Wconv	RBF	RBF	98.60	0.046

Stony Brook University

Example: Identification of Atmospheric Variable Using Deep Gaussian Processes[6]

- Goal: modeling temperature using meteorological variables (features).
- Domain of interest: $25Km \times 25Km$ around the nuclear power plant in Krško, Slovenia.
- Features: relative humidity, atmosphere stability, air pressure, global solar radiation, wind speed.

[6] Mitja Jančič, Juš Kocijan, and Boštjan Grašič. "Identification of Atmospheric Variable Using Deep Gaussian Processes". In: *IFAC-PapersOnLine* 51.5 (2018), pp. 43–48.

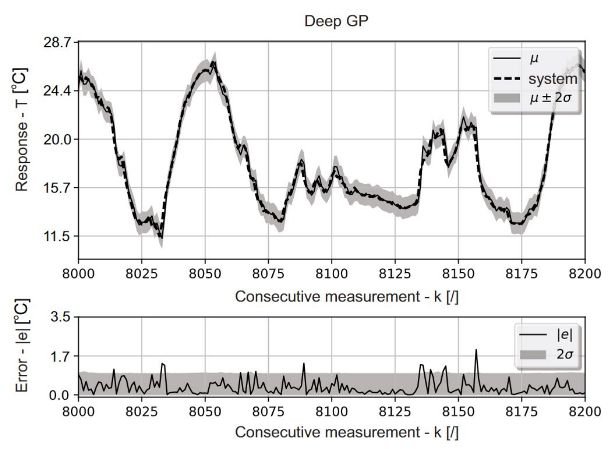
The Geographical Features of the Surrounding Terrain

- The plant and its measurement station (marked as STOLP – Postaja) are situated in the basin surrounded by hills and valleys, which influence micro-climate conditions.

Stony Brook University

One-Step-Ahead Prediction

- Prediction results:



Example: Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data[7]

- Goal: predicting crop yields before harvest
- Model: CNN and LSTM combined with GP

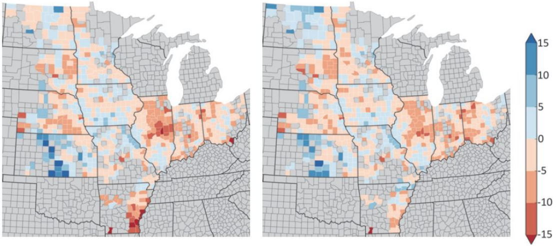
	Baselines			Deep models			
Year	Ridge	Tree	DNN	LSTM	LSTM + GP	CNN	CNN + GP
2011	9.00	7.98	9.97	5.83	5.77	5.76	**5.7**
2012	6.95	7.40	7.58	6.22	6.23	5.91	**5.68**
2013	7.31	8.13	9.20	6.39	5.96	**5.50**	5.83
2014	8.46	7.50	7.66	6.42	5.70	5.27	**4.89**
2015	8.10	7.64	7.19	6.47	**5.49**	6.40	5.67
Avg	7.96	7.73	8.32	6.27	5.83	5.77	**5.55**

[7] Jiaxuan You et al. "Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data.". In: *AAAI*. 2017, pp. 4559–4566.

Stony Brook University

Comparing County-Level Error Maps

- The color represents the prediction error in bushel per acre.



(a) CNN (b) CNN+GP

Stony Brook University

Conclusions

- A case was made for using probability theory in treating uncertainties in inference from data.
- Deep probabilistic modeling based on deep Gaussian processes was addressed.
- The use of DGPs in studying complex interacting systems was described.
- Applications in various fields using DGPs were provided.
- Although the development of DGPs is still in its relatively early stages, DGPs showed great potentials in many challenging machine learning tasks.

Stony Brook University