Additive Kernel GPPVAE

Santeri Mentu

Aalto University CSB group

Summer 2019

Contents

- Motivation for the work
- Mathematical description
- Software implementation
- Experimental results

Aalto University
School of Science

Motivation for the work

- Variational Autoencoders (VAE) are a powerful method for unsupervised learning
- However the i.i.d. assumption for latent representations is too strong
- We would like to include label covariates, such as time, in the model
- Multiple extensions and variations on the VAE have been published
- ...

Aalto University
School of Science

- Gaussian Process Variational Autoencoder (GPPVAE) is one such extension
- Gaussian Processes (GPs) act as prior for the latent space
- The GPs are indexed with label information
- Once the model is trained, we can perform label inference on new data

Bayesian inference

- In a generative model the probability of observations is defined by a likelihood $p(x|z)$ and prior on the latent space $p(z)$
- Ideally we could simply maximize the marginal probability of the posterior

$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

- However, the evidence term in the denominator is intractable
- Therefore we use a variational approximation where the latent distribution and the likelihoods are parametrized by NNs

Aalto University
School of Science

Structure of a VAE

- We optimize the variational approximation of the latent distribution by minimizing the KL divergence between it, and the prior
- This is equivalent to maximizing the evidence lower bound, or ELBO

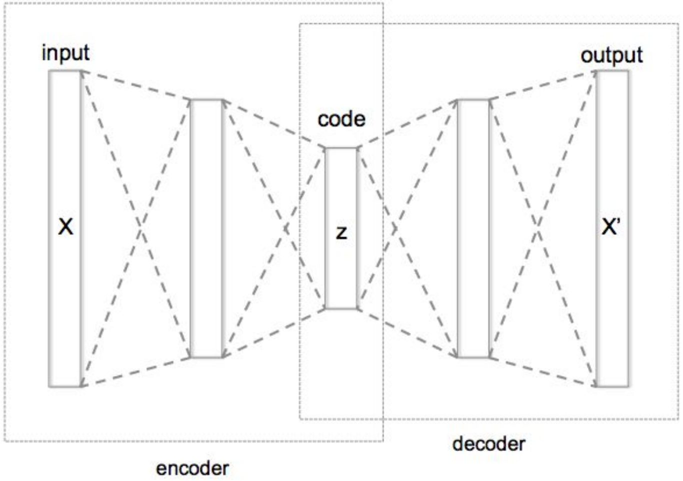


Figure from Wikipedia

Aalto University
School of Science

(Very) short intro to Gaussian Processes

- A Gaussian process is a stochastic process where each finite collection of variables has a multivariate normal distribution
- A zero mean GP is completely defined by its covariance function, or kernel
- The choice of kernel defines the function space in GP regression

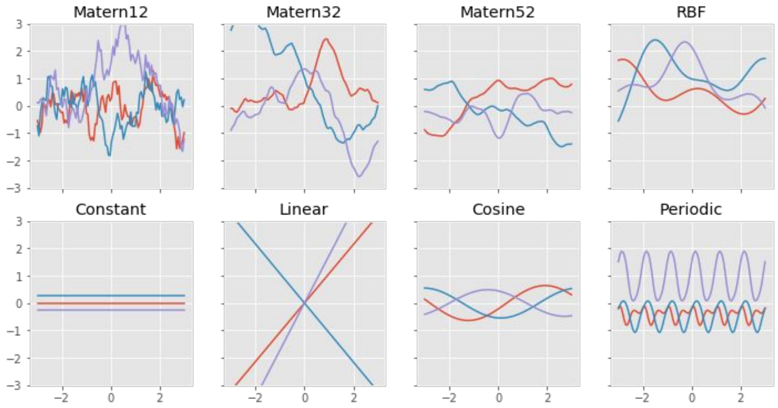


Figure from GPflow by Matthews et al.

Aalto University
School of Science

Gaussian prior

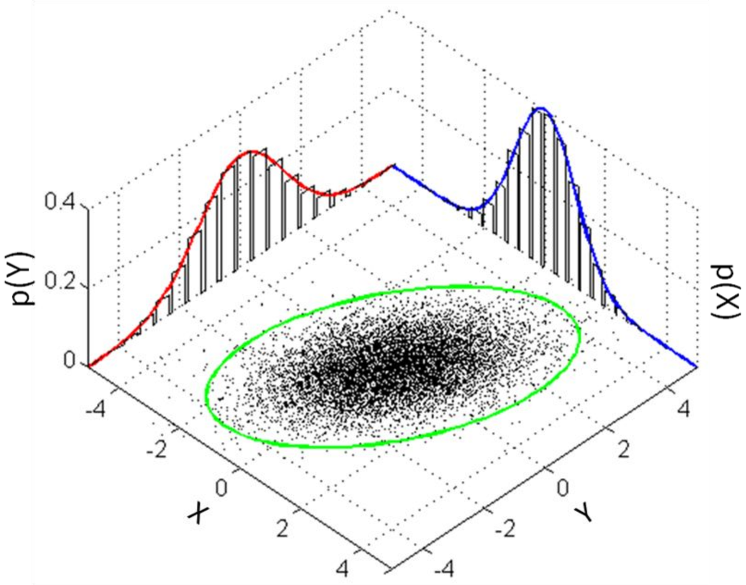


Figure from Wikipedia

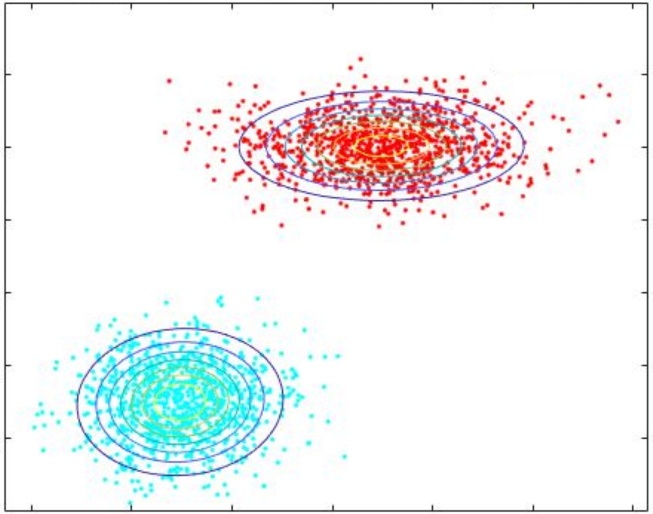


Figure by MathWorks

Aalto University
School of Science

ELBO for GPPVAE

- ELBO derived by Casale et al.

$$\log p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{W}, \phi, \sigma_y^2, \boldsymbol{\theta}) \geq \mathbb{E}_{\boldsymbol{Z} \sim q_{\psi}} \left[\sum_n \log \mathcal{N}(\boldsymbol{y}_n \mid g_{\boldsymbol{\phi}}(\boldsymbol{z}_n), \sigma_y^2 \boldsymbol{I}_K) + \log p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\theta}, \alpha) \right] + \\ + \frac{1}{2} \sum_{nl} \log(\boldsymbol{\sigma}^{z}{}_{\boldsymbol{\psi}}^{2}(\boldsymbol{y}_n)_l) + \text{const.}$$

- Which gives the loss for SGD

$$l(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}, \alpha, \sigma_y^2) = \\ = NK \log \sigma_y^2 + \underbrace{\sum_n \frac{\left\| \boldsymbol{y}_n - g_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}_n}) \right\|^2}{2\sigma_y^2}}_{\text{reconstruction term}} - \underbrace{\log p(\boldsymbol{Z}_{\boldsymbol{\psi}} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\theta}, \alpha)}_{\text{latent-space GP term}} + \underbrace{\frac{1}{2} \sum_{nl} \log(\boldsymbol{\sigma}^{z}{}_{\boldsymbol{\psi}}^{2}(\boldsymbol{y}_n)_l)}_{\text{regularization term}},$$

GP likelihoods

The marginal likelihood for observations

$$p(\boldsymbol{y}, f_*) = \int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}, f_*)\mathrm{d}\boldsymbol{f}$$
$$= \mathcal{N}\left(\begin{bmatrix} \boldsymbol{y} \\ f_* \end{bmatrix} \middle| \boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{ff} + \sigma^2\boldsymbol{I} & \boldsymbol{K}_{f_* f} \\ \boldsymbol{K}_{f_* f} & K_{f_* f_*} \end{bmatrix}\right)$$

The marginal likelihood of kernel parameters

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{\theta})\mathrm{d}\boldsymbol{f}$$
$$= \int \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{f}, \sigma^2\boldsymbol{I}\right)\mathcal{N}\left(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}\right)\mathrm{d}\boldsymbol{f}$$
$$= \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{0}, \sigma^2\boldsymbol{I} + \boldsymbol{K}\right)$$

Aalto University
School of Science

Additive and multiplicative kernels

- It would be most beneficial if kernels could be configured in a modular and flexible manner
- My implementations allows for kernel addition and multiplication thanks to the GPyTorch library

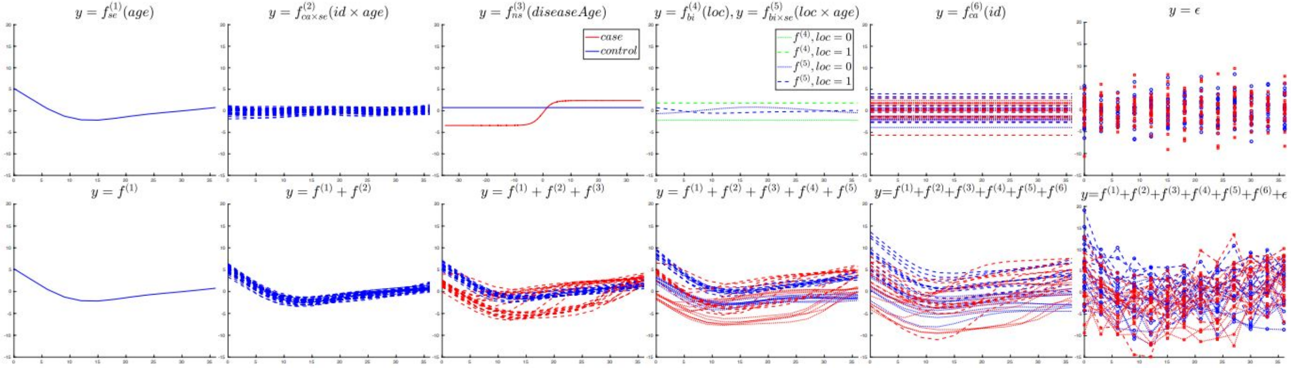


Figure by Cheng et al.

Aalto University
School of Science

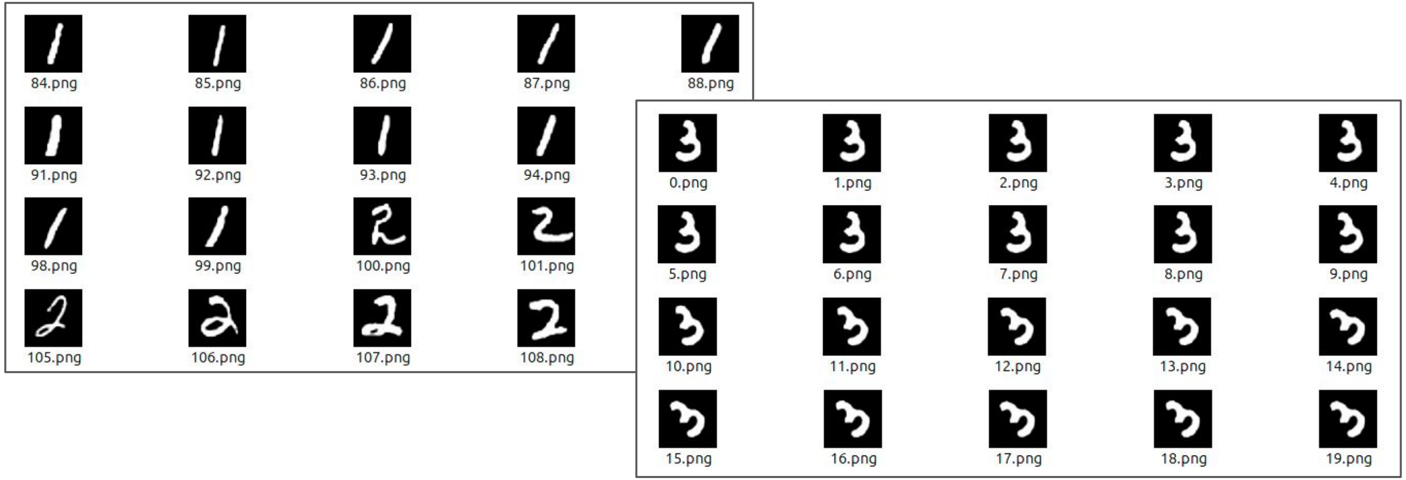
Software Implementation

- The implementation was built using PyTorch and the GPyTorch GP library
- GPyTorch provides a highly modular and flexible framework for building GP models, which can be modified to suit specific requirements
- The implementation is available at Aalto Version

Aalto University
School of Science

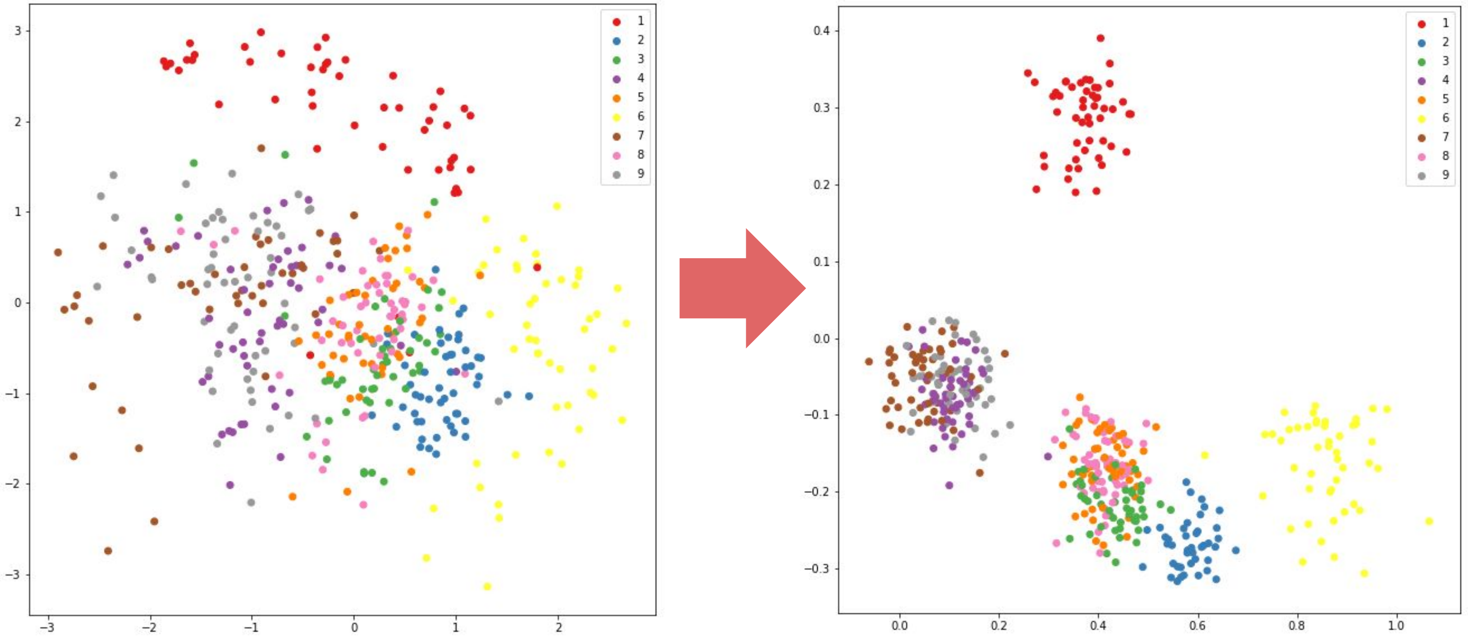
Test datasets

- The implementation was tested using two toy datasets

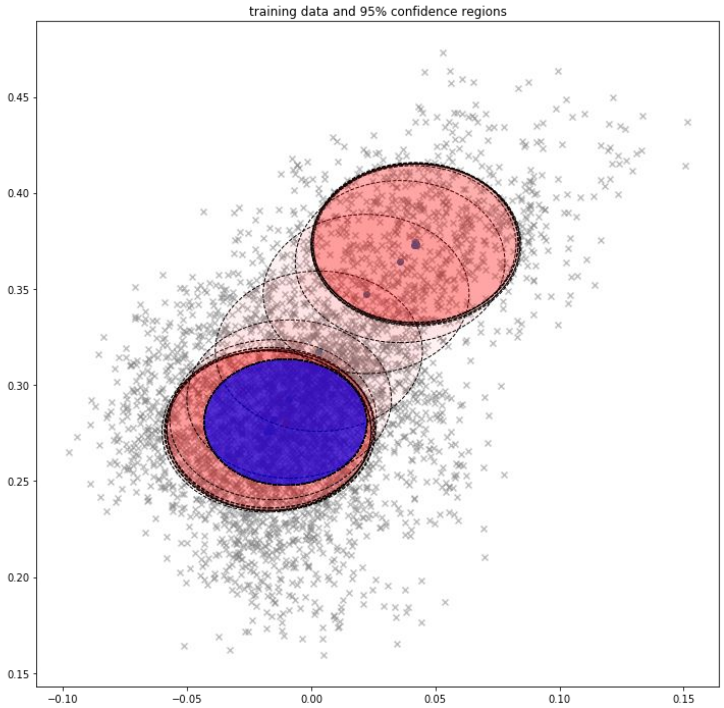


Aalto University
School of Science

MNIST with categorical kernel

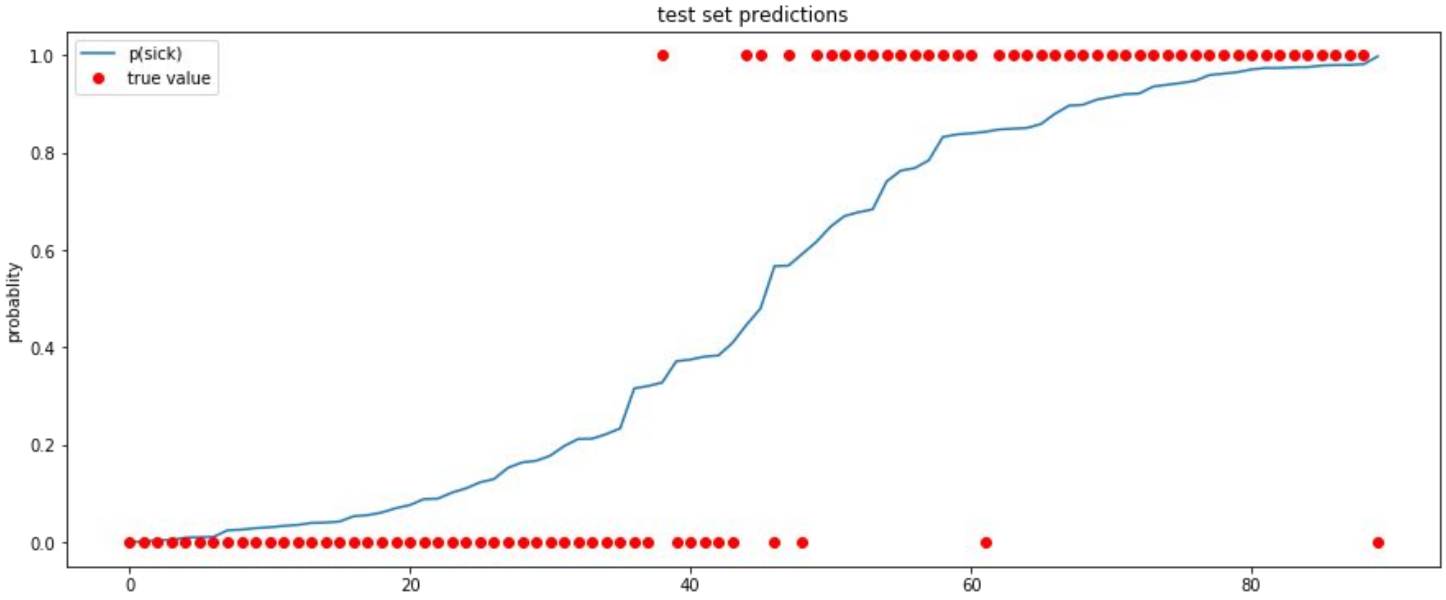


Rotating MNIST with binary + warping kernel



Aalto University
School of Science

Rotating MNIST label inference



Aalto University
School of Science

All code and notebooks available at:

https://version.aalto.fi/gitlab/mentus1/HIT_VAE

Aalto University
School of Science

Special thanks to Gleb Tikhonov

Aalto University
School of Science