Bayesian Gaussian Process Latent Variable Model

Michalis K. Titsias and Neil D. Lawrence
School of Computer Science,
University of Manchester

Motivation

- Gaussian processes are used for supervized learning
  - Inputs are fixed/deterministic
- Gaussian process latent variable model (GP-LVM) is trained by optimizing (not marginalizing out) the latent variables

We address the questions:

- How can we train Gaussian process models when inputs are random (e.g. we have uncertain inputs/missing values)?
- How can we marginalize out the latent variables in GP-LVM?

We will introduce a variational Bayes framework that provides approximate Bayesian solutions

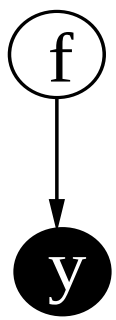
Outline

- Variational inference for GPs with random (uncertain/missing/latent) inputs
  - The role of inducing variables
  - The variational lower bound
- Variational inference for GP-LVM
  - Automatic selection of the latent dimensionality with the squared exponential ARD kernel
- Experiments with GP-LVM
- Summary

Gaussian Processes: Deterministic inputs

- Gaussian process (GP) is used as non-parametric prior over some latent function $f(\mathbf{x})$
- Supervised learning: Estimate regression functions, decision boundaries, intensities etc

- Probability model: Output-input data $(\mathbf{y}, X)$:



$$p(\mathbf{y}, \mathbf{f}|X) = p(\mathbf{y}|\mathbf{f}) \times p(\mathbf{f}|X)$$
$$\text{Joint} = \text{Likelihood} \times \text{marginal GP on X}$$

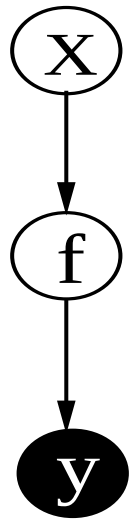
where X is assumed deterministic

But what if the inputs X are random?

Gaussian Processes: Random inputs

- Probability model: As before, but now the inputs X are given a prior (e.g. Gaussian) distribution $p(X)$:

$$p(\mathbf{y}, \mathbf{f}, X) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|X)p(X)$$



- Random inputs can be:
  - Uncertain inputs, i.e. noisy input measurements
  - Missing values in X
  - Latent variables in non-linear probabilistic PCA (GP-LVM)
- The posterior distribution $p(\mathbf{f}, X|\mathbf{y})$ and the marginal likelihood $p(\mathbf{y})$ are intractable

Can we apply the standard mean field approximation?

Variational inference: Difficult to apply

- Standard regression with random inputs:

$$p(\mathbf{y}, \mathbf{f}, X) = \underbrace{\mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I)}_{\text{Gaussian Lik.}} \underbrace{p(\mathbf{f}|X)}_{\mathcal{N}(\mathbf{f}|\mathbf{0}, K_{NN})} \underbrace{p(X)}_{\text{Gaussian}}$$

- A mean field approximation $q(\mathbf{f}, X) = q(\mathbf{f})q(X)$ is difficult to apply:
  - X appears non-linearly inside the kernel matrix inverse K_{NN}^{-1}
  - Seems impossible to compute the variational bound $\int q(\mathbf{f}, X) \log \frac{p(\mathbf{y}, \mathbf{f}, X)}{q(\mathbf{f}, X)} d\mathbf{f} dX$
- But there is a trick:
  - Augment with a finite set of auxiliary parameters
    - These will be extra points of the function $f(\mathbf{x})$ called inducing variables

But why we need auxiliary parameters?

Variational inference: Why we need auxiliary parameters?

- Bayesian linear regression with random inputs

$$\mathbf{y} = X\mathbf{w} + \epsilon, \quad N(\mathbf{w}|\mathbf{0}, \sigma_w^2 I), p(X)$$

  - It is straightforward to apply mean field using $q(\mathbf{w})q(X)$
- Kernelization: We can express the model in a function (non-parametric) view as

$$\mathbf{y} = \mathbf{f} + \epsilon, \quad N(\mathbf{f}|\mathbf{0}, \sigma_w^2 XX^T), p(X)$$

  where the GP prior has a linear kernel
- The kernelization makes variational inference difficult
  - X appears in the inverse of XX^T
  - Not clear how to apply mean field using $q(\mathbf{f})q(X)$

Variational inference: Why we need auxiliary parameters?

- Gaussian processes (kernel methods in general) are somehow marginalized (collapsed)
  - A GP is an exchangeable model:

$$p(f_1, \ldots, f_N) = \int \prod_{n=1}^{N} p(f_n|\mathbf{w}) dP(\mathbf{w})$$

    where the underlying (infinite) parameter $\mathbf{w}$ has been integrated out
  - We are left with the kernel function and inputs that appear inside matrix inverses $\Rightarrow$ intractability
- We need to discover some (approximate) parameters to apply variational inference

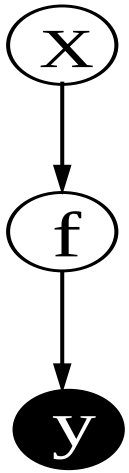
The parameters we use are auxiliary function points used in sparse GPs, called inducing variables

Inducing variables: The general idea

- Initial model:

$$p(\mathbf{y}, \mathbf{f}, X) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I) p(\mathbf{f}|X) p(X)$$

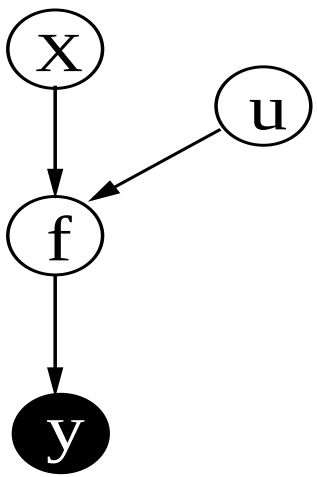
(variational inference in the space of $(\mathbf{f}, X)$ is difficult)



- Augment consistently[a] with inducing variables $\mathbf{u} = (f(\mathbf{z}_1), \ldots, f(\mathbf{z}_M))$:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}, X) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I) p(\mathbf{f}|\mathbf{u}, X) p(\mathbf{u}) p(X)$$

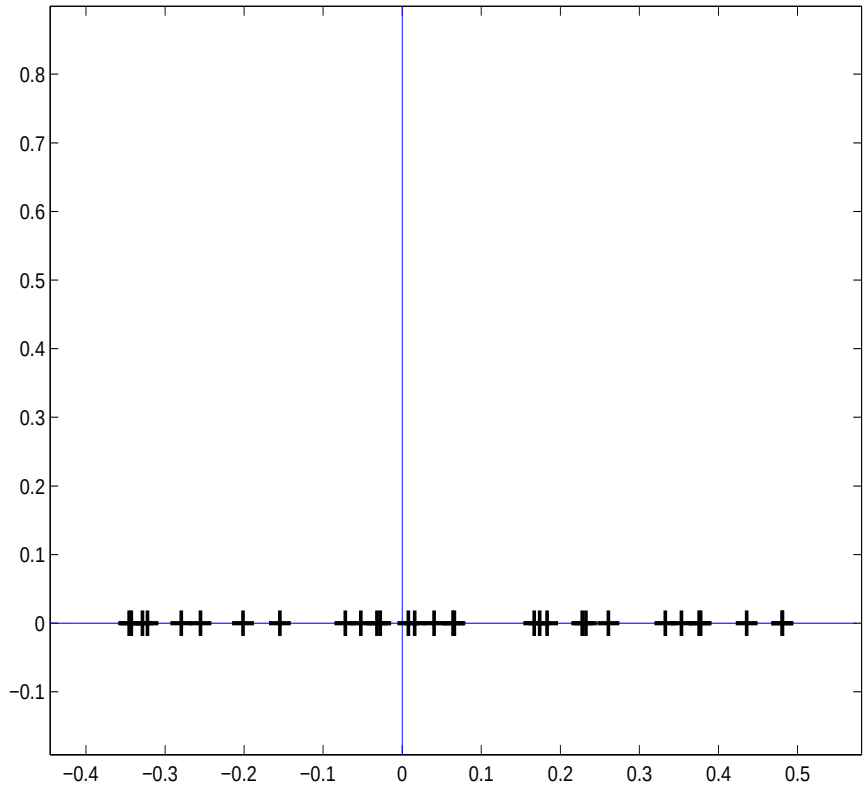
(variational inference in the space of $(\mathbf{f}, \mathbf{u}, X)$ is tractable)



[a] $\int p(\mathbf{f}|\mathbf{u}, X) p(\mathbf{u}) d\mathbf{u} = p(\mathbf{f}|X)$, for any value of inputs Z

Inducing variables: Linear GP

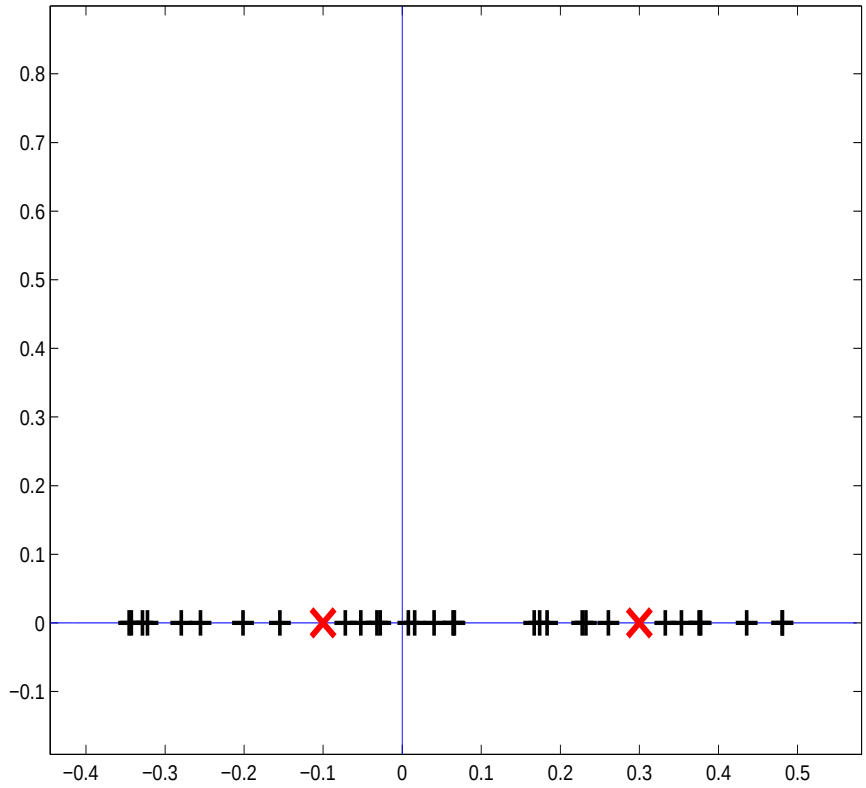
Visualization of the augmented GP model and the inducing variables



- Draw input data X:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}, X) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I) p(\mathbf{f}|\mathbf{u}, X) p(\mathbf{u}) p(X)$$

Inducing variables: Linear GP

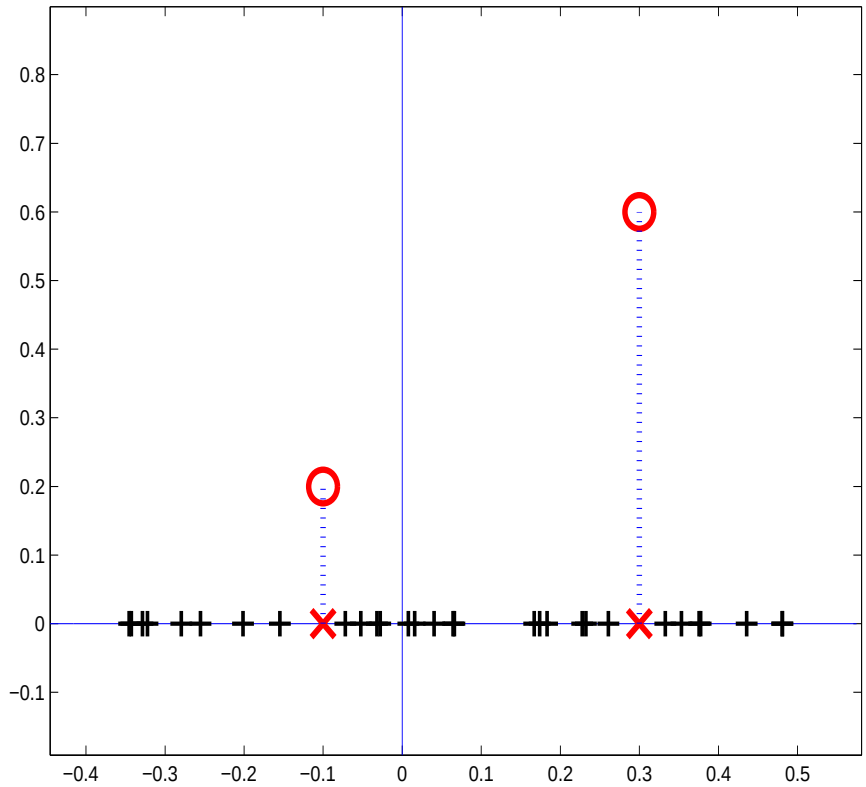


- Choose some pseudo (unrelated to X) inputs Z

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}, X) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I)p(\mathbf{f}|\mathbf{u}, X)p(\mathbf{u})p(X)$$

Crucial: Z are not random variables

Inducing variables: Linear GP

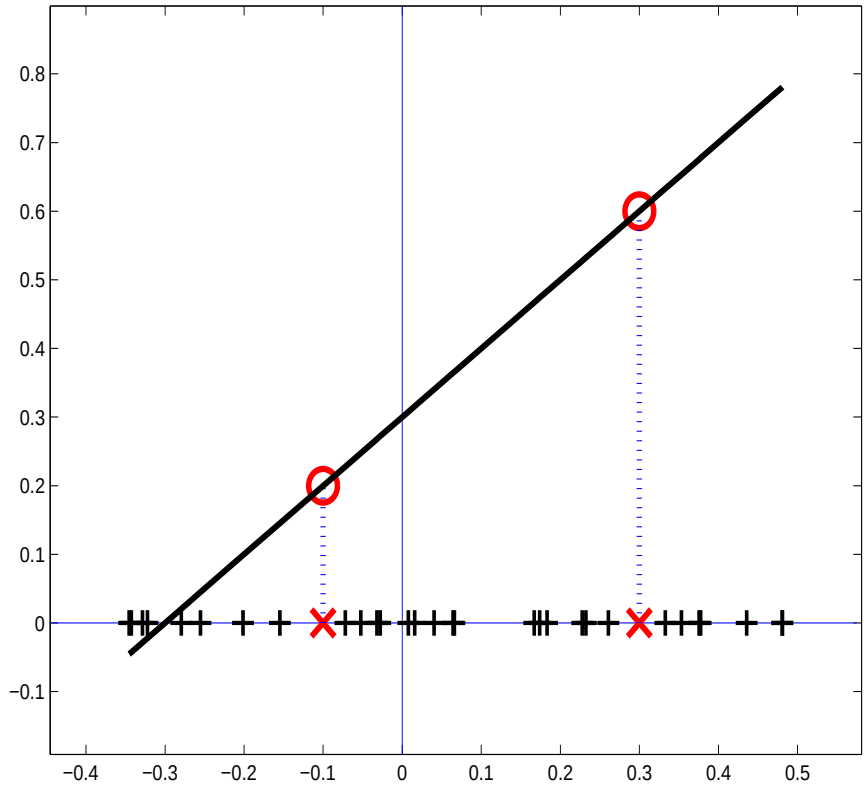


- Sample random function values $\mathbf{u}$ at the pseudo-inputs Z

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}, X) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I) p(\mathbf{f}|\mathbf{u}, X) p(\mathbf{u}) p(X)$$

where $p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, ZZ^T)$

Inducing variables: Linear GP

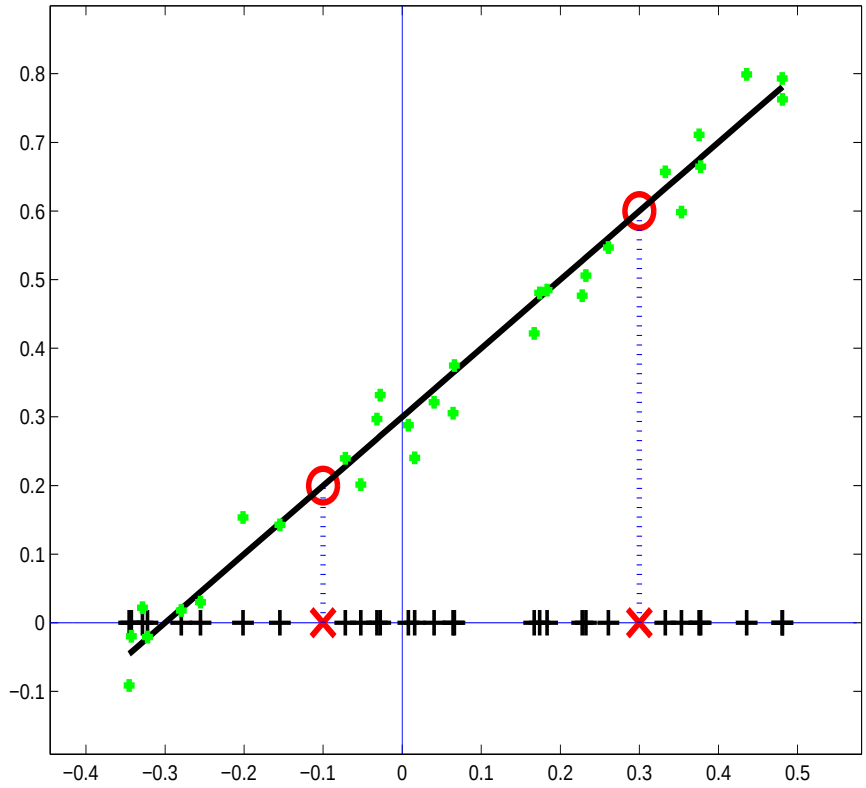


- Sample function values $\mathbf{f}$ on training inputs X so that the function passes from the inducing variables

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}, X) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I)p(\mathbf{f}|\mathbf{u}, X)p(\mathbf{u})p(X)$$

($p(\mathbf{f}|\mathbf{u}, X)$ is the delta function in the example!)

Inducing variables: Linear GP



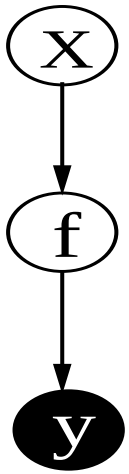
- Generate the observed data $\mathbf{y}$

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}, X) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I) p(\mathbf{f}|\mathbf{u}, X) p(\mathbf{u}) p(X)$$

Variational inference

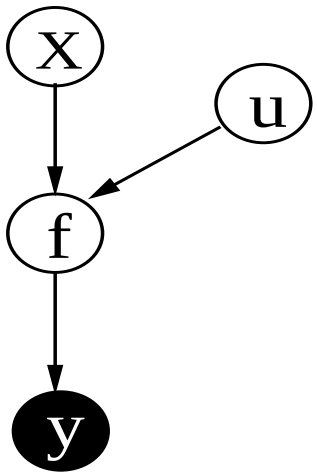
- Initial model:

$$p(\mathbf{y}, \mathbf{f}, X) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I) p(\mathbf{f}|X) p(X)$$



- Augmented model:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}, X) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I) p(\mathbf{f}|\mathbf{u}, X) p(\mathbf{u}) p(X)$$



We apply variational inference in the space of $(\mathbf{f}, \mathbf{u}, X)$

Variational inference

- Variational distribution:

$$q(\mathbf{f}, \mathbf{u}, X) = p(\mathbf{f}|\mathbf{u}, X)\phi(\mathbf{u})q(X)$$

  - $q(X) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$: Gaussian distribution
  - $\phi(\mathbf{u})$: unrestricted (turns out to be Gaussian)
  - $p(\mathbf{f}|\mathbf{u}, X)$: conditional GP prior (**!!trick!!**)

- Maximize the lower bound

$$\log \int \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I)p(\mathbf{f}|\mathbf{u}, X)p(X)d\mathbf{f}d\mathbf{u}\mathbf{X} \geq$$

$$\int p(\mathbf{f}|\mathbf{u}, X)\phi(\mathbf{u})q(X) \log \frac{\mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I)p(\mathbf{f}|\mathbf{u}, X)p(\mathbf{u})p(X)}{p(\mathbf{f}|\mathbf{u}, X)\phi(\mathbf{u})q(X)} d\mathbf{f}d\mathbf{u}\mathbf{X}$$

where $p(\mathbf{f}|\mathbf{u}, X)$s inside the log cancel

This is now tractable. Matrix inverses containing X are gone

Variational inference

$$\int p(\mathbf{f}|\mathbf{u}, X)\phi(\mathbf{u})q(X) \log \frac{\mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I)p(\mathbf{u})p(X)}{\phi(\mathbf{u})q(X)} d\mathbf{f}\, d\mathbf{u}\, d\mathbf{X}$$

- The lower bound is analytically tractable for linear kernels, squared exponential, exponential, polynomial kernels and possibly others
- It is maximized jointly over variational parameters and model hyperparameters

Gaussian process latent variables model (Lawrence, 2005)

- Latent variable model:

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

  - $\mathbf{y} \in \mathbb{R}^D$: observed variable
  - $\mathbf{x} \in \mathbb{R}^Q$ ($Q \ll D$): latent variable
  - $\mathbf{f} : \mathbb{R}^Q \rightarrow \mathbb{R}^D$: latent mapping
  - GP-LVM: GP priors on the latent mapping

GP-LVM is trained by optimizing (not marginalizing out) the latent variables

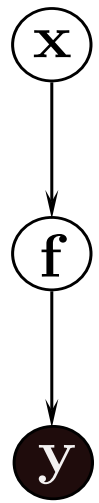
- Not proper density in the latent space
- Cannot select the latent dimensionality Q
- It may overfit since it is not fully Bayesian

Bayesian Gaussian process latent variables model

- Latent variable model:

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

- Bayesian training: Integrate out both the latent mapping and the latent space
  - Exact Bayesian inference is intractable
  - But variational Bayesian inference is tractable



The variational method is applied as before. The only difference is that now we have D latent functions (one for each observed output) and not just one

Bayesian Gaussian process latent variables model

Automatic selection of the latent dimensionality

- Squared exponential ARD kernel

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} \alpha_q (x_q - x_q')^2\right)$$

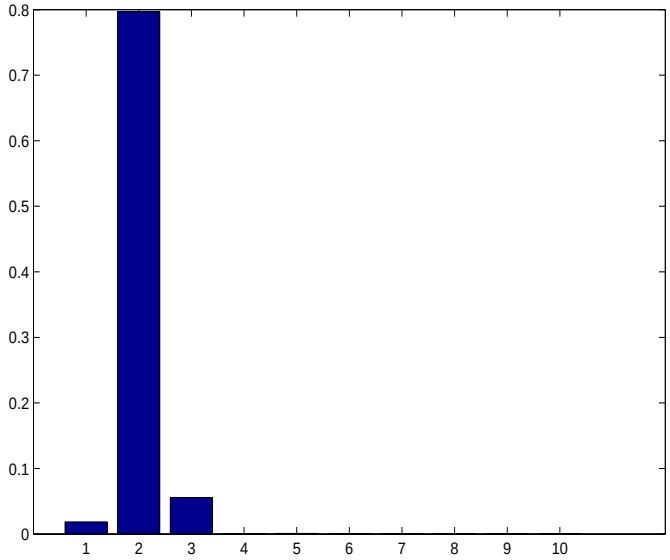
- Maximizing the variational lower bound w.r.t. α_qs allows to remove redundant latent dimensions

Experiments: Visualization

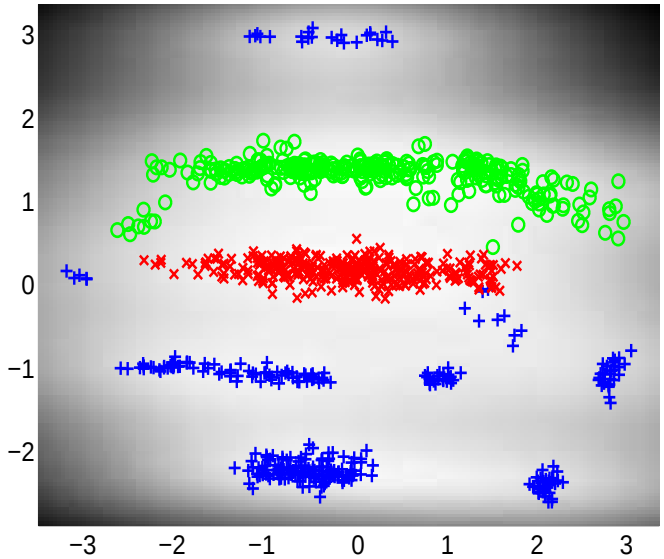
- Oil flow data: 1000 training; 12 dimensions; 3 known classes
- Compare:
  - Bayesian GP-LVM
  - Standard sparse GP-LVM
  - Probabilistic PCA

Experiments: Visualization

Oil flow data



(α_qs)

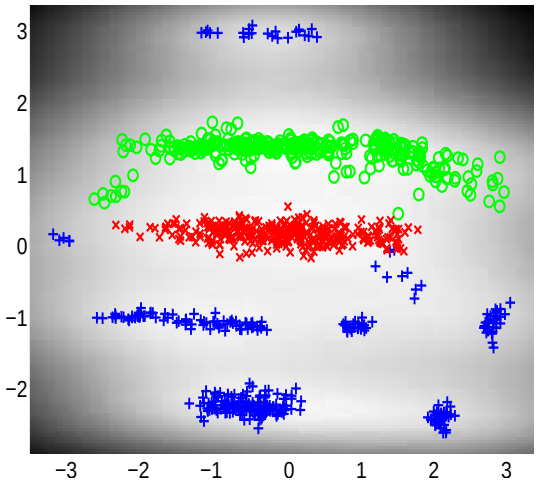


(Bayesian GP-LVM)

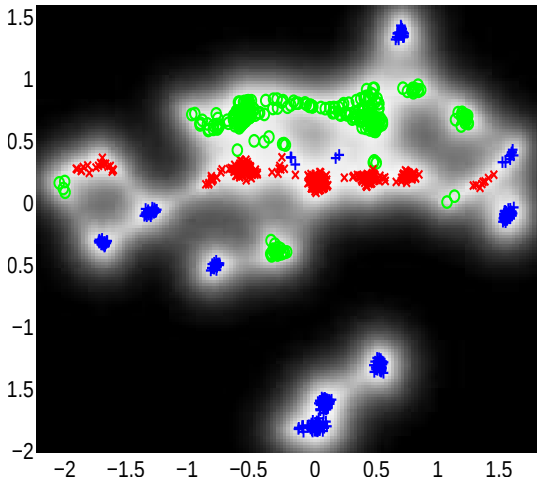
- Bayesian GP-LVM runs with 10 latent dimensions
- The red, green and blue points are the predicted means for the latent variables labeled with the known class
- 7 out 10 latent dimensions are shrunk to zero
- Visualization is shown for the dominant (with the largest inverse lengthscales) latent dimensions

Experiments: Visualization

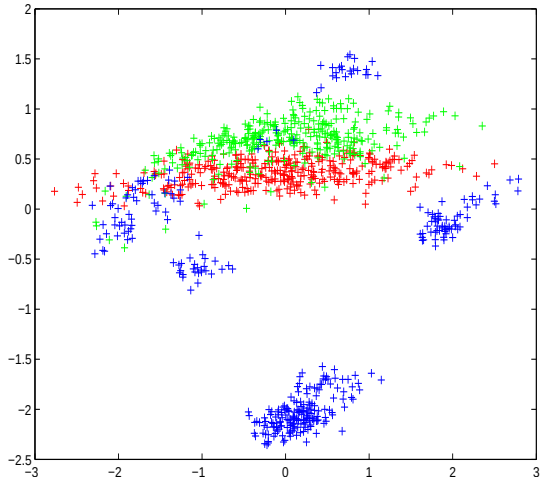
Oil flow data



(Bayesian GP-LVM)



(GP-LVM)

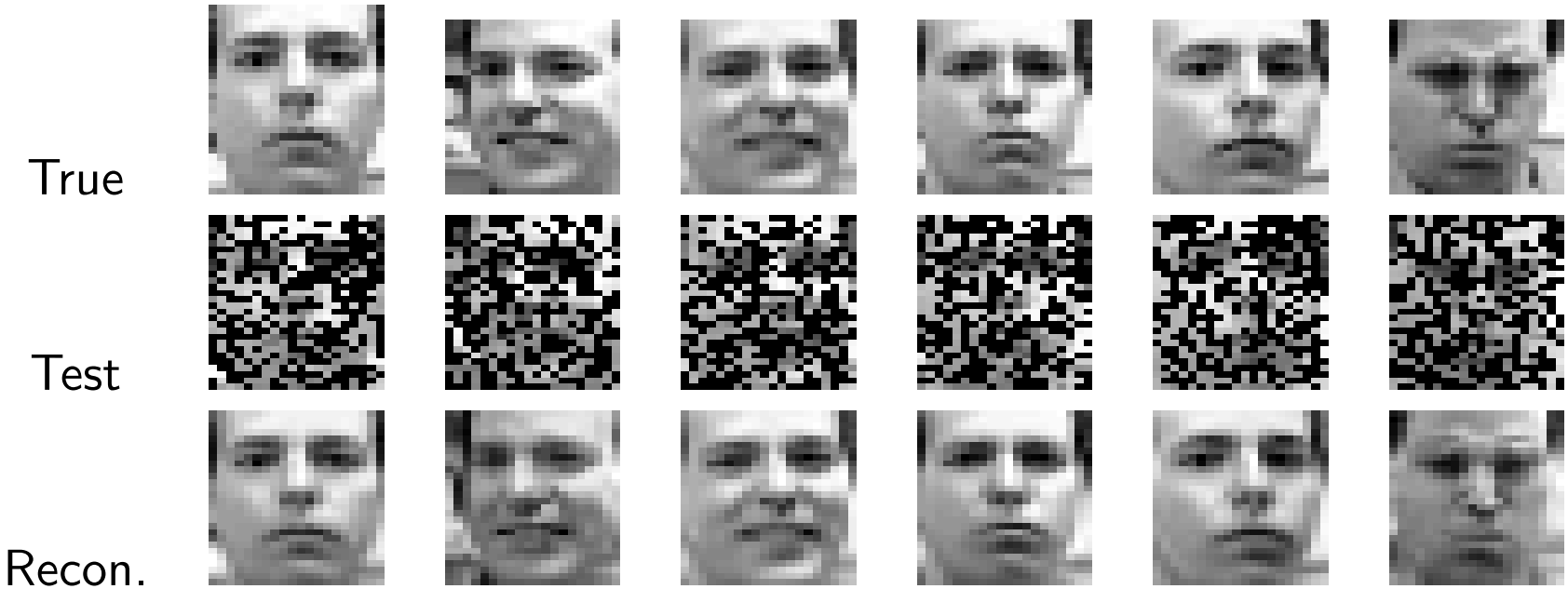


(PPCA)

GP-LVM and Bayesian GP-LVM are both initialized based on PCA

Experiments: Predict missing values

Frey faces: 1965 images; $28 \times 20 = 560$ dimensions; 1000 for training; 965 for testing



- Bayesian GP-LVM is trained with 30 latent dimensions, mean absolute reconstruction error: 7.4003
- Standard sparse GP-LVM is trained with several latent dimensions: $Q = 2, 5, 10, 30$. Errors: $10.5748, 9.7284, 19.6949, 19.6961$

Experiments: Generative classification

- USPS digits dataset: 16×16 images for all 10 digits, 7291 training examples and 2007 test examples
- Run 10 Bayesian GP-LVMs: one for each digit
- Compute Bayesian class conditional densities in the test data of the form $p(\mathbf{y}_* | Y, \text{digit})$

Results: From 2007 test images we have 95 incorrectly classified digits, i.e. 4.73% error

Summary/Future work

Summary:

- Variational framework to approximately integrate out inputs in GPs
- Allows for Bayesian training of GP-LVM

Future work:

- Optimization: Currently we use conjugate gradients to jointly maximize the lower bound over variational and model parameters
  - Improvements: Find fixed point updates, explore the correlation structure of the optimized parameters
- Learn non-parametric/non-linear dynamical systems using GPs and variational Bayes