# Heterogeneous Multi-output Gaussian Process Prediction

Pablo Moreno-Muñoz
Antonio Artés-Rodríguez
Mauricio A. Álvarez

uc3m | Universidad Carlos III de Madrid

**Poster #14**

# Motivation

uc3m | Universidad Carlos III de Madrid

**Problem**

- Output observations are a mix of continuous, binary, categorical or discrete variables
- Multi-output Gaussian process models usually focus on all-regression or all-classification tasks

**Goal**

- Provide an extension of multi-output Gaussian processes for prediction in arbitrary heterogeneous datasets

# Motivation

## Applications

### Medical

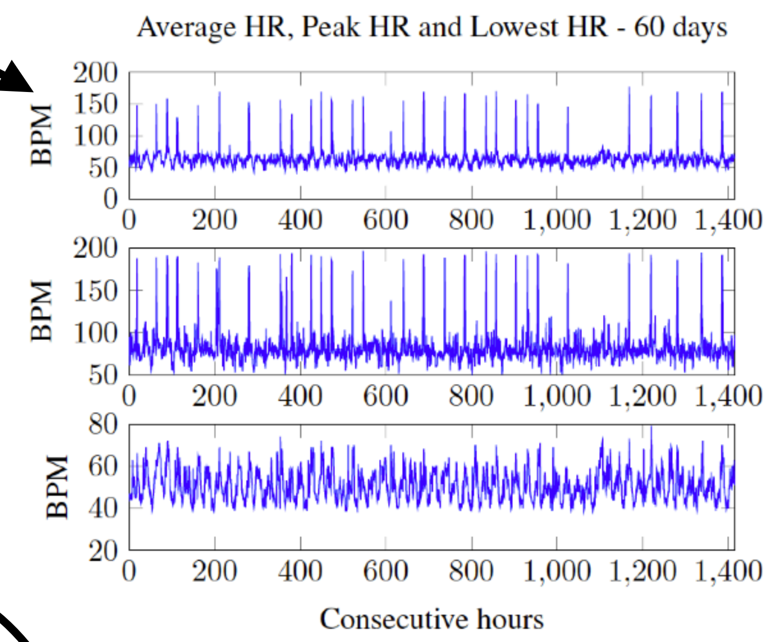
- Intensive Care Units (ICU) Electronic Health Records

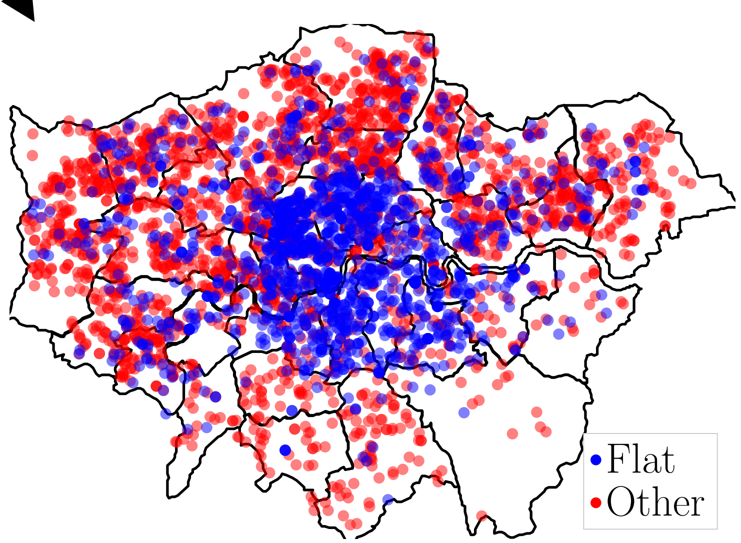
### Spatio-temporal

- Demographic, sociological or economic analysis of cities

### Bayesian Optimization

- Functions with multiple outputs from different nature



# Heterogeneous Likelihoods

$$p(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x})) = \prod_{d=1}^{D} p(\mathbf{y}_d|\boldsymbol{\theta}_d(\mathbf{x}))$$

# Heterogeneous Likelihoods

uc3m | Universidad Carlos III de Madrid

$$p(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x})) = \prod_{d=1}^{D} p(\mathbf{y}_d|\boldsymbol{\theta}_d(\mathbf{x}))$$

$D$ different statistical data types

# Heterogeneous Likelihoods

$$p(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x})) = \prod_{d=1}^{D} p(\mathbf{y}_d|\boldsymbol{\theta}_d(\mathbf{x}))$$

Example with $D = 3$

$\mathrm{Ber}(\mathbf{y}_1|\boldsymbol{\theta}_1(\mathbf{x}))$

$\mathrm{Poisson}(\mathbf{y}_2|\boldsymbol{\theta}_2(\mathbf{x}))$

$\mathcal{N}(\mathbf{y}_3|\boldsymbol{\theta}_3(\mathbf{x}))$

$\boldsymbol{\theta}(\mathbf{x}) = \phi(f(\mathbf{x}))$

$f \sim \mathcal{GP}(0, k(\cdot,\cdot))$

# Heterogeneous Likelihoods

$$p(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x})) = \prod_{d=1}^{D} p(\mathbf{y}_d|\boldsymbol{\theta}_d(\mathbf{x}))$$

Example with $D = 3$

Link functions

$\text{Ber}(\mathbf{y}_1|\boldsymbol{\theta}_1(\mathbf{x}))$ → $\boldsymbol{\theta}_1(\mathbf{x}) = 1/(1 + \exp(-f_1(\mathbf{x})))$

$\text{Poisson}(\mathbf{y}_2|\boldsymbol{\theta}_2(\mathbf{x}))$ → $\boldsymbol{\theta}_2(\mathbf{x}) = \exp(f_2(\mathbf{x}))$

$\mathcal{N}(\mathbf{y}_3|\boldsymbol{\theta}_3(\mathbf{x}))$ → $\boldsymbol{\theta}_3(\mathbf{x}) = f_3(\mathbf{x})$

# Heterogeneous Likelihoods

$$p(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x})) = \prod_{d=1}^{D} p(\mathbf{y}_d|\boldsymbol{\theta}_d(\mathbf{x}))$$

Example with $D = 3$

$$\boldsymbol{\theta}_1(\mathbf{x}) = 1/(1+\exp(-f_1(\mathbf{x})))$$

$$\boldsymbol{\theta}_2(\mathbf{x}) = \exp(f_2(\mathbf{x}))$$

$$\boldsymbol{\theta}_3(\mathbf{x}) = f_3(\mathbf{x})$$

$$f(\mathbf{x}) = \sum_{q=1}^{Q} a_q u_q(\mathbf{x})$$

Linear combination

# Heterogeneous Likelihoods

uc3m | Universidad Carlos III de Madrid

$$p(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x})) = \prod_{d=1}^{D} p(\mathbf{y}_d|\boldsymbol{\theta}_d(\mathbf{x}))$$

Example with $D = 3$

$$\boldsymbol{\theta}_1(\mathbf{x}) = 1/(1 + \exp(-f_1(\mathbf{x})))$$

$$\boldsymbol{\theta}_2(\mathbf{x}) = \exp(f_2(\mathbf{x}))$$

$$\boldsymbol{\theta}_3(\mathbf{x}) = f_3(\mathbf{x})$$

$$f(\mathbf{x}) = \sum_{q=1}^{Q} a_q u_q(\mathbf{x})$$

Linear combination

# Multi-parameter priors

Independent latent functions

$$u_q(\mathbf{x}) \sim \mathcal{GP}(0, k_q(\cdot,\cdot))$$

Multi-output GP prior

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \sim \mathcal{MOGP}\left(\mathbf{0}, \sum_{q=1}^{Q} \mathbf{B}_q \otimes k_q(\cdot,\cdot)\right)$$

$$f_1(\mathbf{x}) = \sum_{q=1}^{Q} a_{1,q} u_q(\mathbf{x})$$

$$f_2(\mathbf{x}) = \sum_{q=1}^{Q} a_{2,q} u_q(\mathbf{x})$$

$$f_3(\mathbf{x}) = \sum_{q=1}^{Q} a_{3,q} u_q(\mathbf{x})$$

# Scalable inference



Sparse MOGP approximation

$$\mathbf{u}_q = [u_q(\mathbf{z}_1), \cdots, u_q(\mathbf{z}_M)]^\top$$

with **inducing** inputs $\mathbf{Z} = \{\mathbf{z}_m\}_{m=1}^{M}$

and variational inference

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

# Scalable inference

Sparse MOGP approximation

$$\mathbf{u}_q = [u_q(\mathbf{z}_1), \cdots, u_q(\mathbf{z}_M)]^\top$$

with **inducing** inputs $\mathbf{Z} = \{\mathbf{z}_m\}_{m=1}^{M}$

and variational inference

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

# Scalable inference

Sparse MOGP approximation

$$\mathbf{u}_q = [u_q(\mathbf{z}_1), \cdots, u_q(\mathbf{z}_M)]^\top$$

with **inducing** inputs $\mathbf{Z} = \{\mathbf{z}_m\}_{m=1}^{M}$

and variational inference

$$q(\mathbf{f}, \mathbf{u}) = \prod_{d=1}^{D} \prod_{j=1}^{J_d} p(\mathbf{f}_{d,j}|\mathbf{u}) \prod_{q=1}^{Q} q(\mathbf{u}_q)$$

where $q(\mathbf{u}_q) = \mathcal{N}(\mathbf{u}_q|\boldsymbol{\mu}_{\mathbf{u}_q}, \mathbf{S}_{\mathbf{u}_q})$

# Scalable inference

Variational Lower Bound

$$\mathcal{L} = \iint p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} d\mathbf{u} - \sum_{q=1}^{Q} \mathrm{KL}(q(\mathbf{u}_q)||p(\mathbf{u}_q))$$

# Scalable inference

Variational Lower Bound

$$\mathcal{L} = \iint p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} d\mathbf{u} - \sum_{q=1}^{Q} \mathrm{KL}(q(\mathbf{u}_q)||p(\mathbf{u}_q))$$

double expectation

factorised divergences

# Scalable inference

Variational Lower Bound

$$\sum_{n=1}^{N}\sum_{d=1}^{D}\mathbb{E}_{q(\mathbf{f}_d(\mathbf{x}_n))}[\log p(\mathbf{y}_d|\mathbf{f}_d(\mathbf{x}_n))] - \sum_{q=1}^{Q}\mathrm{KL}(q(\mathbf{u}_q||p(\mathbf{u}_q)))$$

amenable for stochastic optimisation

# Scalable inference

Variational Lower Bound

$$\sum_{n=1}^{N}\sum_{d=1}^{D}\mathbb{E}_{q(\mathbf{f}_d(\mathbf{x}_n))}[\log p(\mathbf{y}_d|\mathbf{f}_d(\mathbf{x}_n))] - \sum_{q=1}^{Q}\mathrm{KL}(q(\mathbf{u}_q||p(\mathbf{u}_q)))$$

Variational parameters

$\{\mu_{\mathbf{u}_q}\}_{q=1}^{Q}$

$\{\mathbf{L}_{\mathbf{u}_q}\}_{q=1}^{Q}$ where $\mathbf{S}_{\mathbf{u}_q} = \mathbf{L}_{\mathbf{u}_q}\mathbf{L}_{\mathbf{u}_q}^{\top}$

Hyperparameter learning

$\mathbf{Z}$ inducing points

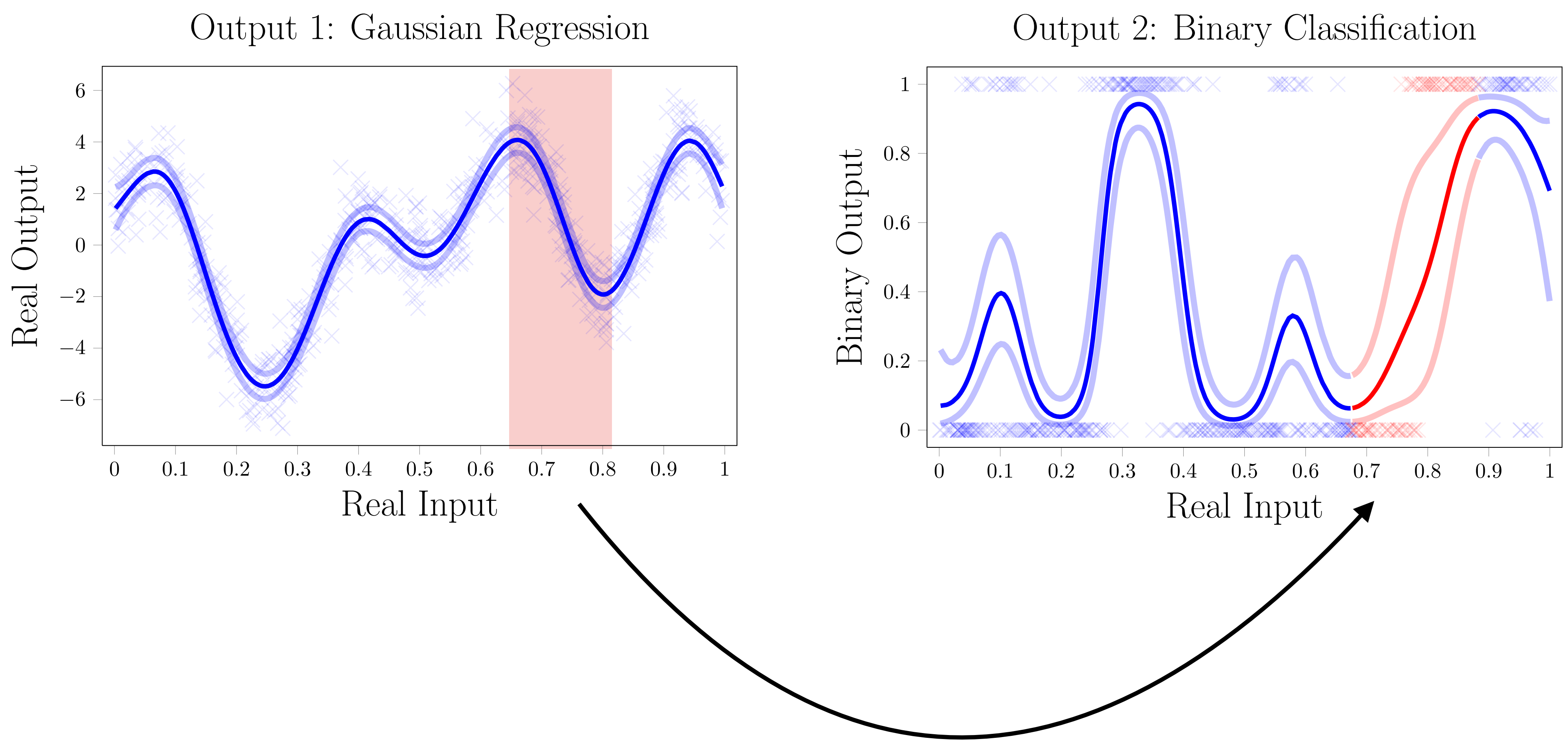
$\{\mathbf{B}_q\}_{q=1}^{Q}$ linear combination

$\{\gamma_q\}_{q=1}^{Q}$ kernel hyperparameters

**VEM algorithm**

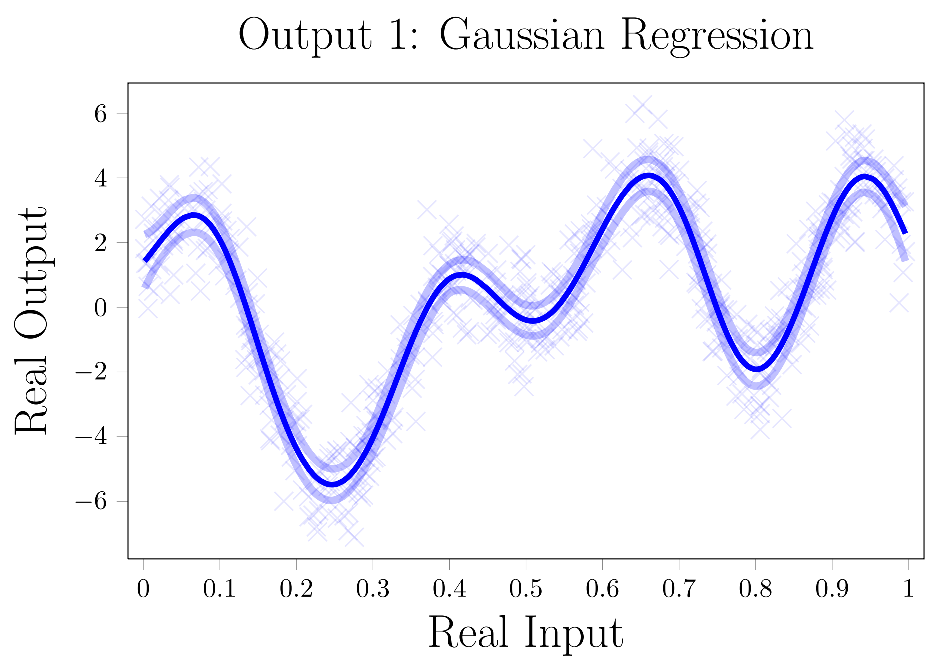
# Results

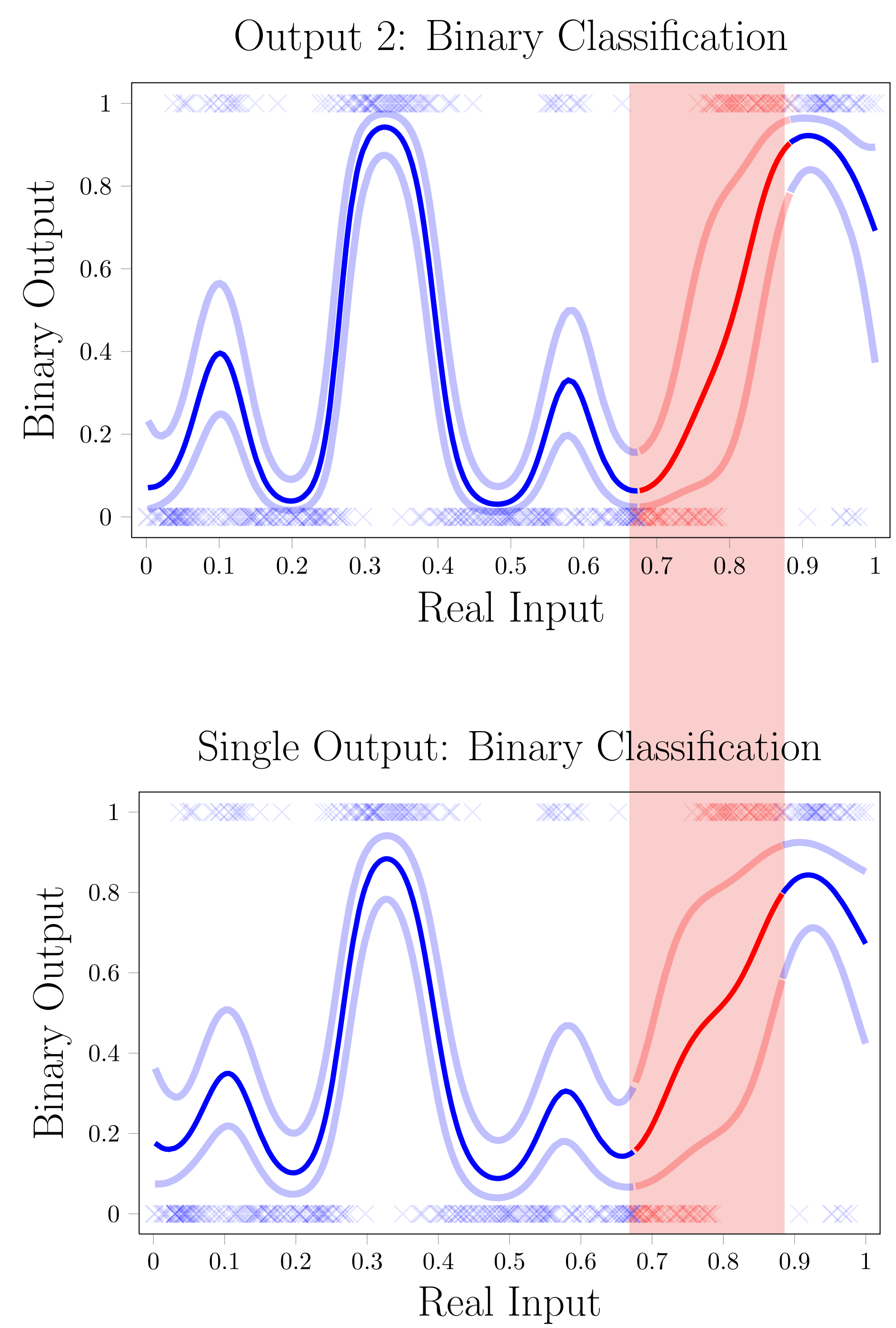
Multi-task toy experiment




**Test prediction** improves in the binary **classification** using training information from the **regression** task

## Multi-task toy experiment

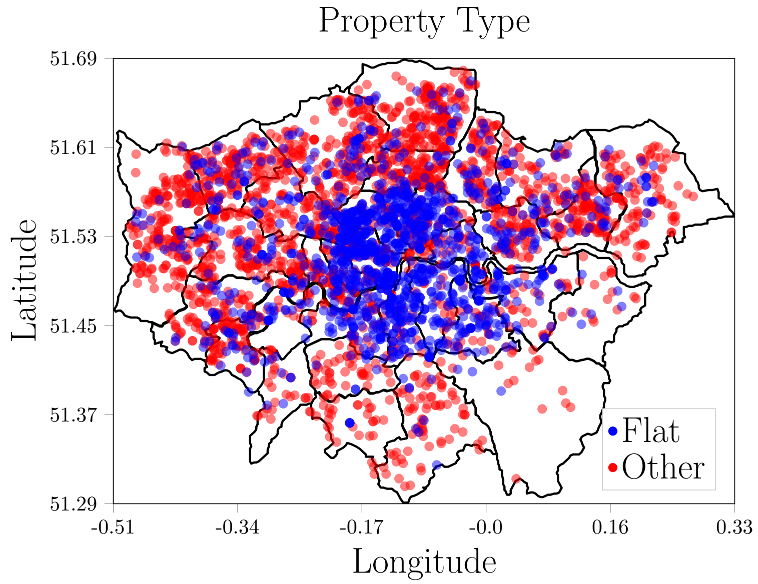


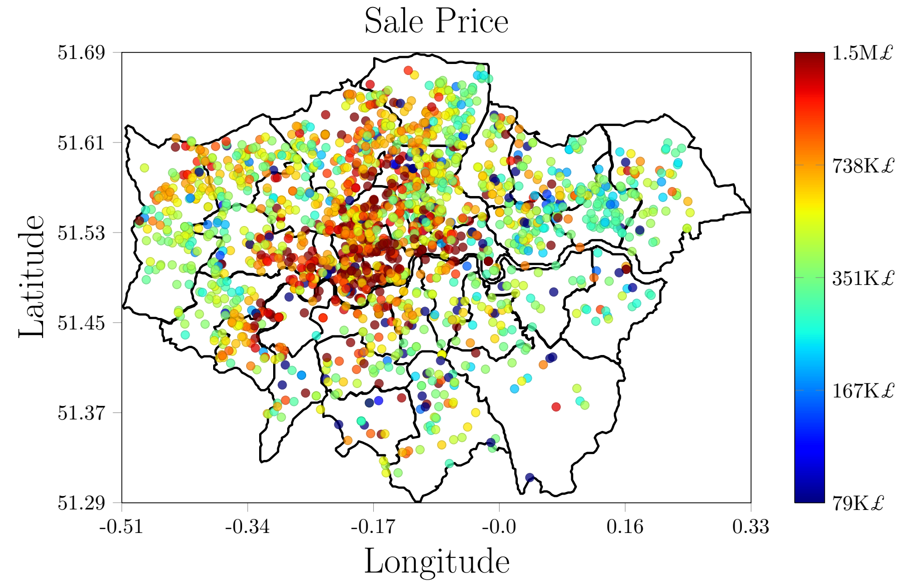




Uncertainty is reduced in multi-output learning with respect to an independent training

# Results

## London House Price Data




Binary classification

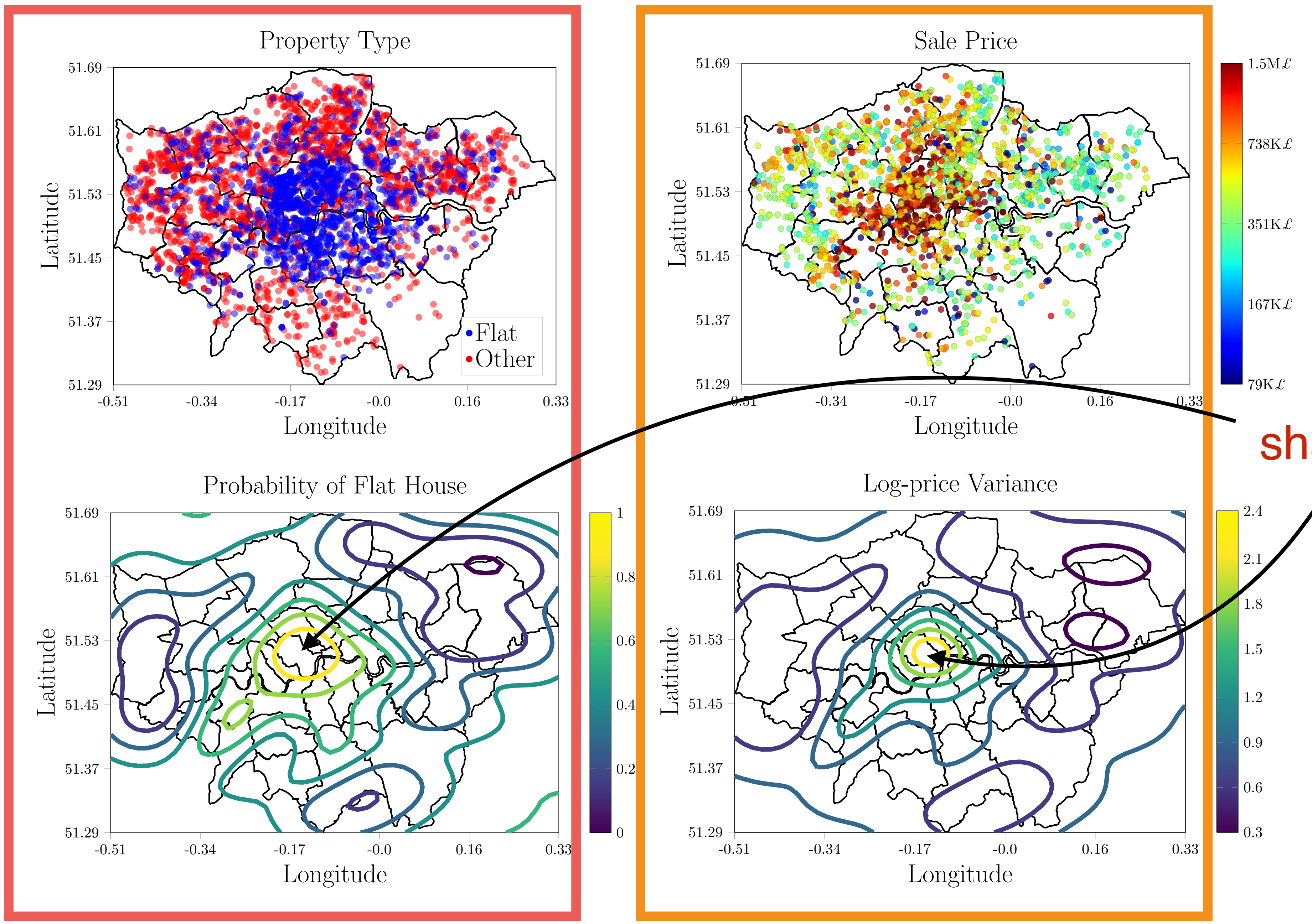
Heteroscedastic Gaussian Regression

Heterogeneous multi-output GP modeling

# Results

Universidad Carlos III de Madrid

## London House Price Data




shared modes

test prediction

# Summary

uc3m | Universidad Carlos III de Madrid

**Thanks for listening!**

Table 2: London Dataset Test-NLPD ($\times 10^{-2}$)

| | **Bernoulli** | **Heteroscedastic** | **Global** |
|---|---|---|---|
| HetMOGP | $\mathbf{6.38 \pm 0.46}$ | $\mathbf{10.05 \pm 0.64}$ | $\mathbf{16.44 \pm 0.01}$ |
| ChainedGP | $6.75 \pm 0.25$ | $10.56 \pm 1.03$ | $17.31 \pm 1.06$ |

See you at **Poster #14** (today)

Code at `github.com/pmorenoz/HetMOGP`

easy syntax for arbitrary heterogeneous likelihoods

```
likelihood_list = [Bernoulli(), Poisson(), Gaussian()]
likelihood_list = [Categorical(), Gamma(), HetGaussian(), Gaussian()]
likelihood_list = [Bernoulli(), Bernoulli(), Beta()]
```

**Python**