Non-parametric Bayesian Methods in Machine Learning

Dr. Simon Rogers
School of Computing Science
University of Glasgow
simon.rogers@glasgow.ac.uk
@sdrogers

April 28, 2014

Outline

- (My) Bayesian philosophy
- Gaussian Processes for Regression and Classification
  - GP preliminaries
  - Classification (including semi-supervised)
  - Regression application 1: clinical (dis)-agreement
  - Regressopn application 2: typing on touch-screens
- Dirichlet Process flavoured Cluster Models
  - DP preliminaries
  - Idenfitying metabolites
  - (if time) Cluster models for multiple data views

About me

- I'm not a statistican by training (don't ask me to prove anything!).
- Education:
  - Undergraduate Degree: Electrical and Electronic Engineering (Bristol)
  - PhD: Machine Learning Techniques for Microarray Analysis (Bristol)
- Currently:
  - Lecturer: Computing Science
  - Research Interests: Machine Learning and Applied Statistics in Computational Biology and Human-Computer Interaction (HCI)

Bayesian Inference

Standard setup:

- We have some data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
- We have a model $p(\mathbf{X}|\mathbf{\Theta})$
- We define a prior $p(\mathbf{\Theta})$

Bayesian Inference

Standard setup:

- We have some data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
- We have a model $p(\mathbf{X}|\mathbf{\Theta})$
- We define a prior $p(\mathbf{\Theta})$
- We use Bayes rule (and typically lots of computation) to compute (or estimate) the posterior:

$$p(\mathbf{\Theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{\Theta})p(\mathbf{\Theta})}{p(\mathbf{X})}$$

Why be Bayesian?

Why be Bayesian?

- Within ML we are often interested in making predictions (predicing y_* from $\mathbf{x}_*$).
- Being Bayesian allows us to *average* over uncertainity in parameters when making predictions:

$$p(y_*|\mathbf{x}_*, \mathbf{X}) = \int p(y_*|\mathbf{x}_*, \mathbf{\Theta}) p(\mathbf{\Theta}|\mathbf{X}) \, d\mathbf{\Theta}$$

Why be Bayesian?

- Within ML we are often interested in making predictions (predicing y_* from $\mathbf{x}_*$).
- Being Bayesian allows us to *average* over uncertainity in parameters when making predictions:

$$p(y_*|\mathbf{x}_*, \mathbf{X}) = \int p(y_*|\mathbf{x}_*, \mathbf{\Theta}) p(\mathbf{\Theta}|\mathbf{X}) \, d\mathbf{\Theta}$$

- Bayes rule tells us how this uncertainty should change as data appear.

Gaussian Processes

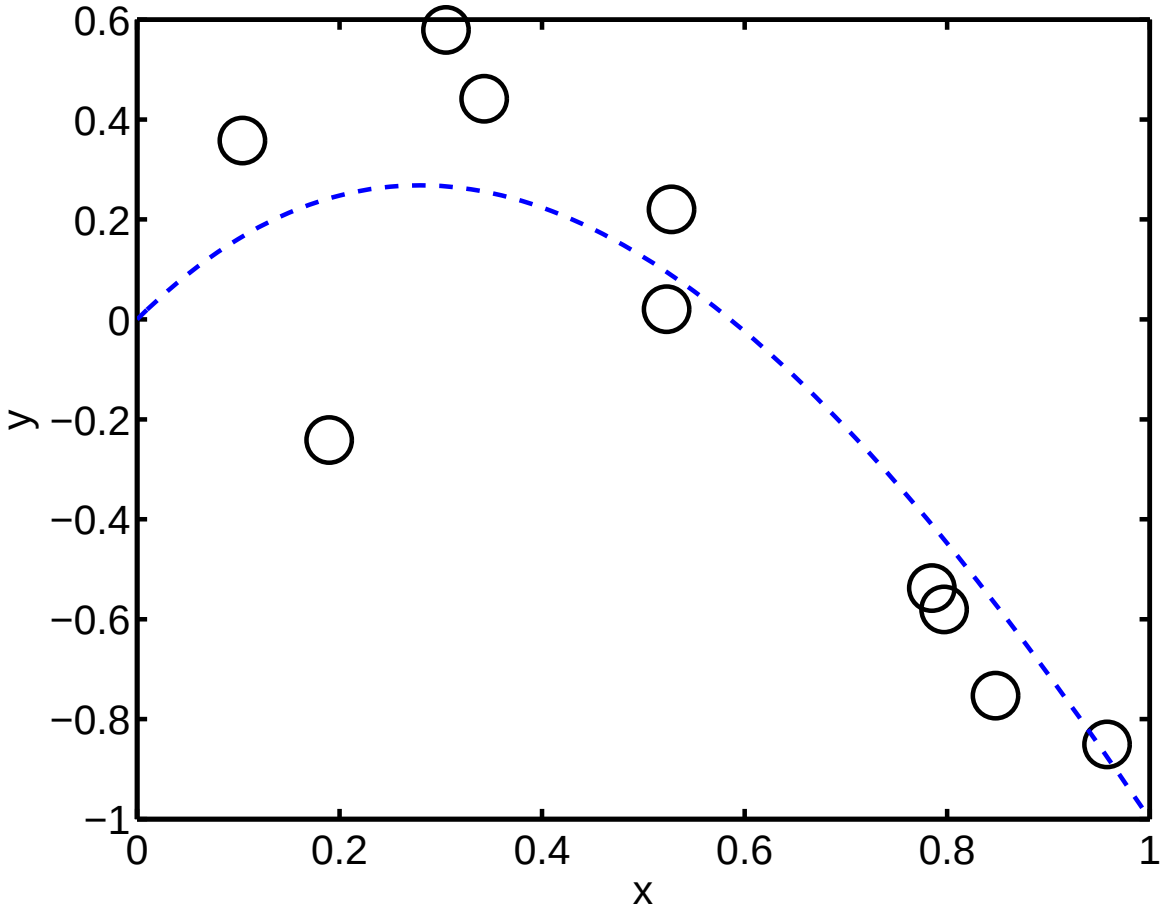

Figure 1 : A familiar problem: learn the underlying function (blue) from the observed data (crosses).

A parametric approach?

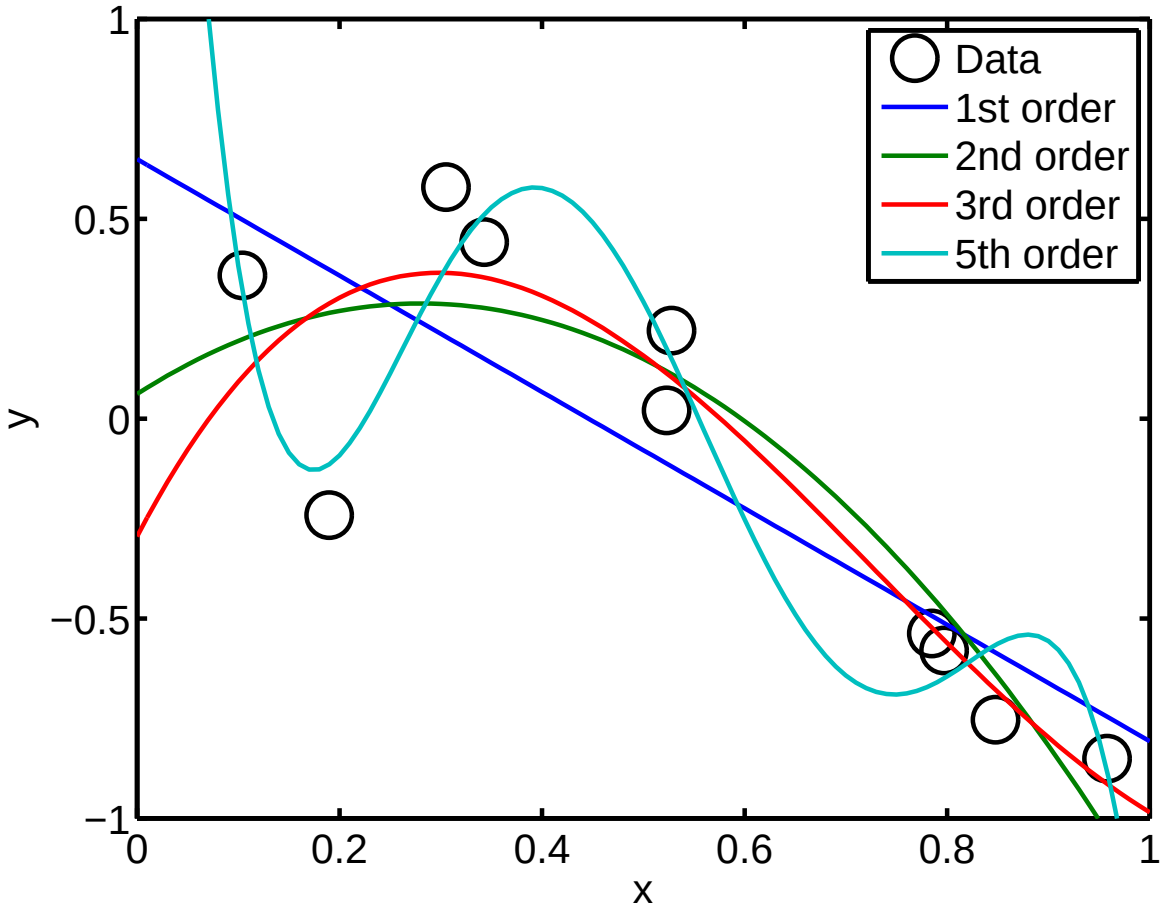

Figure 2 : Polynomials fitted by least squares.

It's easy to under and over-fit. What if we have no idea of the parametric form of the function?

A non-parametric approach - Gaussian Processes

- Rather than forcing us to choose a particular parametric form, a Gaussian Process (GP) allows us to place a prior distribution directly on *functions*
- With a GP prior we can:
  - Sample functions from the prior
  - Incorporate data to get a *posterior* distribution over functions
  - Make predictions

Visual example – prior

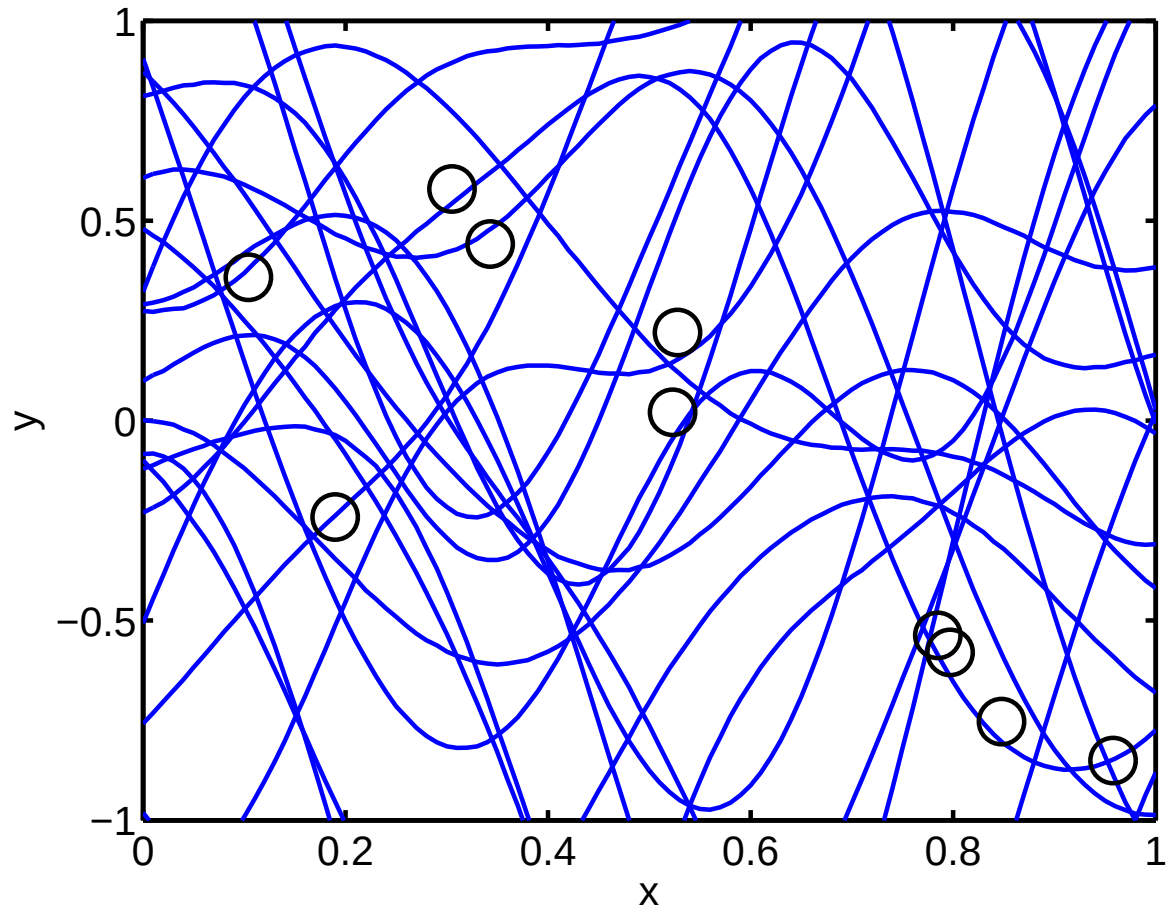

Figure 3 : Some functions drawn from a GP prior.

Visual exmample – posterior

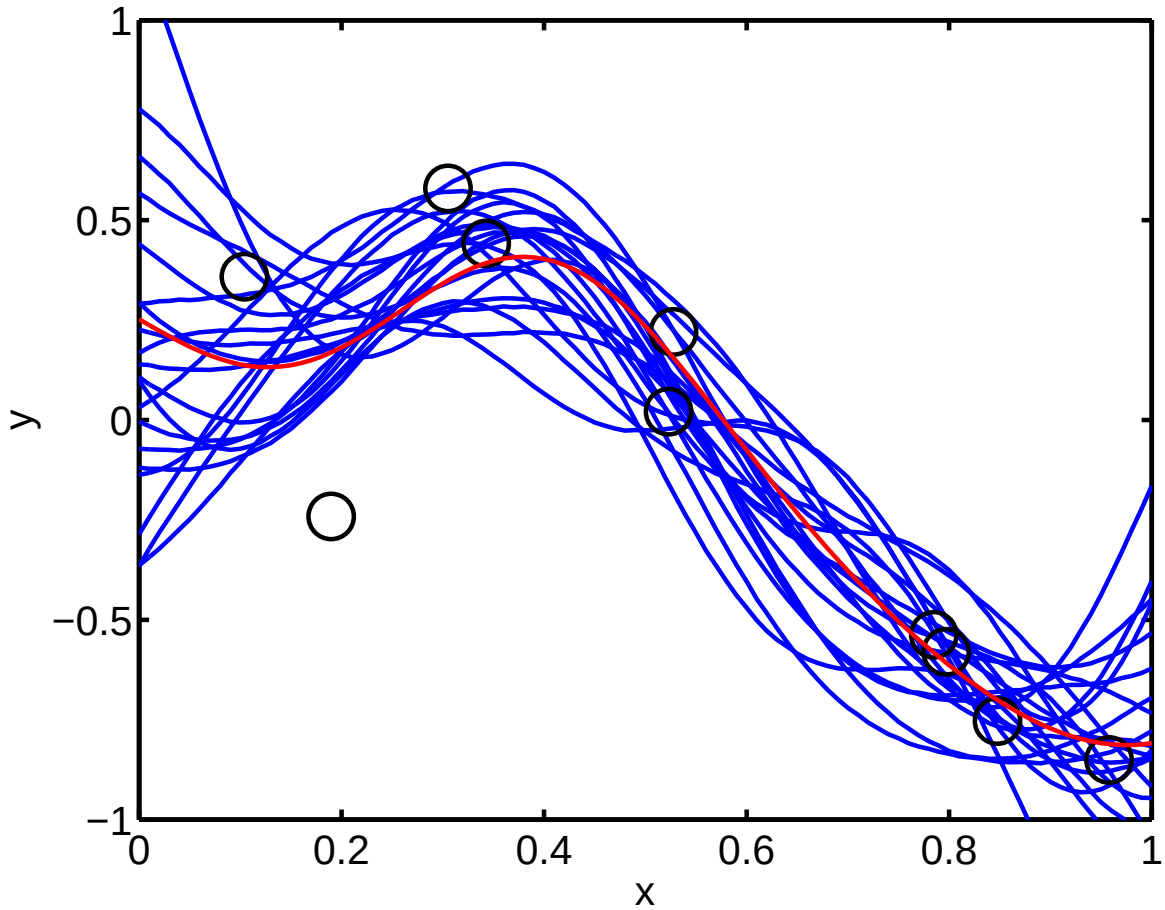

Figure 4 : Some functions drawn from the GP posterior. Posterior mean is shown in red.

Visual example – predictions

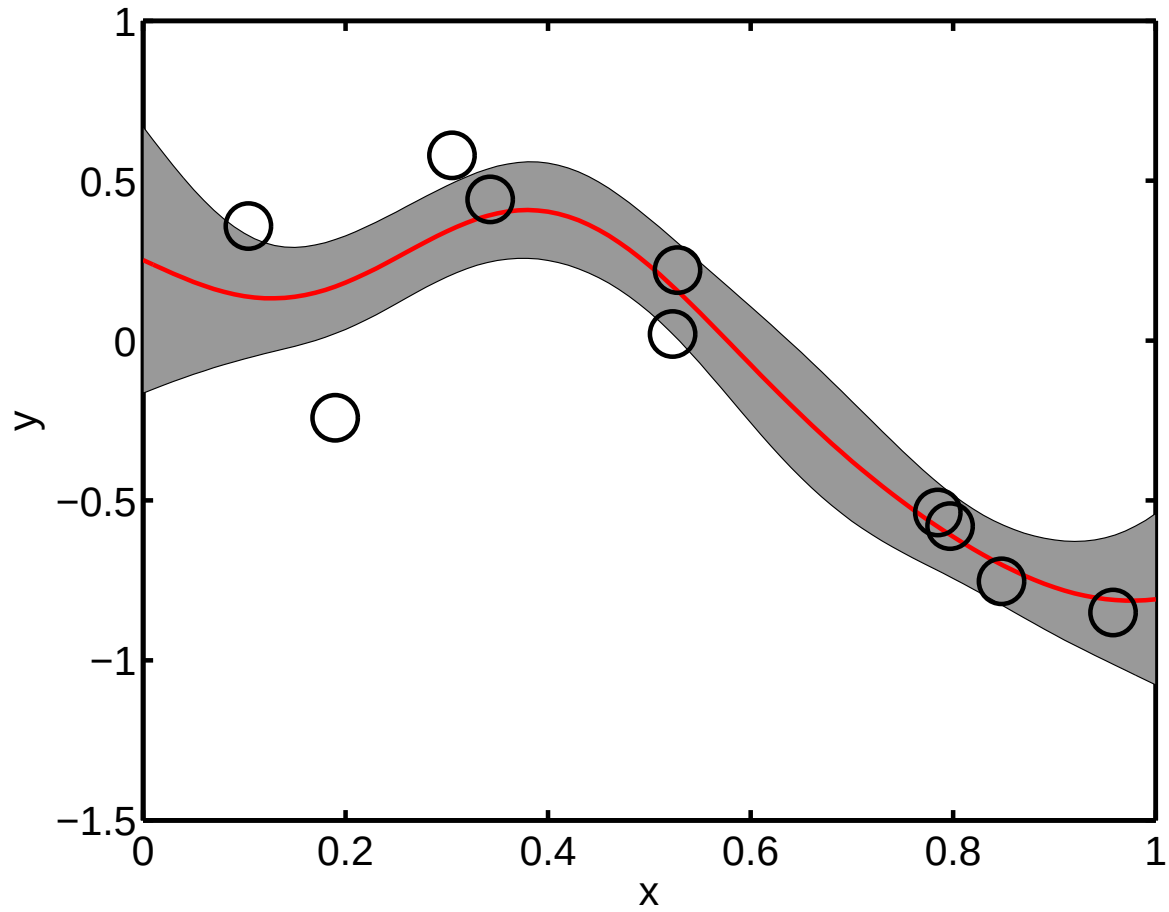

Figure 5 : Predictive mean and standard deviations.

Some formalities

- We observe N training points, each of which consists of a set of features $\mathbf{x}_n$ and a target y_n.
- We can stack all of the y_n into a vector and $\mathbf{x}_n$ into a matrix:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}$$

GP definition

- The GP assumes that the vector of *all possible* y_n is a draw from a Multi-Variate Gaussian (MVG).
- We don't observe *all possible* values (if we did, we wouldn't need to make predictions!)

GP definition

- The GP assumes that the vector of *all possible* y_n is a draw from a MVG.
- We don't observe *all possible* values (if we did, we wouldn't need to make predictions!)
- But the marginal densities of a MVG are also MVGs so the subset we observe are assumed to come from a MVG.

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$$

GP definition

- The GP assumes that the vector of *all possible* y_n is a draw from a MVG.
- We don't observe *all possible* values (if we did, we wouldn't need to make predictions!)
- But the marginal densities of a MVG are also MVGs so the subset we observe are assumed to come from a MVG.

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$$

- With mean $\boldsymbol{\mu}$ (normally 0) and covariance $\mathbf{C}$

GP definition

- The GP assumes that the vector of *all possible* y_n is a draw from a MVG.
- We don't observe *all possible* values (if we did, we wouldn't need to make predictions!)
- But the marginal densities of a MVG are also MVGs so the subset we observe are assumed to come from a MVG.

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$$

- With mean $\boldsymbol{\mu}$ (normally 0) and covariance $\mathbf{C}$
- $\mathbf{x}_n$ looks to have disappeared – we find it inside $\mathbf{C}$.