Non-parametric Bayesian Methods in Machine Learning

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

May 11, 2014

Outline

- (My) Bayesian philosophy
- Gaussian Processes for Regression and Classification (Monday)
  - GP preliminaries
  - *Application 1*: typing on touch-screens
  - Classification (including semi-supervised)
  - *Application 2*: clinical (dis)-agreement
- Dirichlet Process flavoured Cluster Models (Tuesday)
  - DP preliminaries
  - *Application 3*:Idenfitying metabolites
  - *Application 4*:Cluster models for multiple data views
- Summary

Relevant publications

- The four applications are described in the following papers:
  - Uncertain Text Entry on Mobile Devices Weir et. al, CHI 2014
  - Investigating the Disagreement Between Clinicians' Ratings of Patients in ICUs Rogers et. al 2013, IEEE Trans Biomed Health Inform
  - MetAssign: Probabilistic annotation of metabolites from LC–MS data using a Bayesian clustering approach Daly et. al, Bioinformatics, under review
  - Infinite factorization of multiple non-parametric views Rogers et. al, Machine Learning 2009

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)

Lecture 1: Bayesian Inference

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

May 11, 2014

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?

- Ability to incoroporate prior information?

Why Be Bayesian?

- Ability to incoroporate prior information?
- Ability to compute posterior densities (combine prior with likelihood)?

Why Be Bayesian?

- Ability to incoroporate prior information?
- Ability to compute posterior densities (combine prior with likelihood)?
- Ability to compare models via marginal likelihood?

Why Be Bayesian?

- Ability to incoroporate prior information?
- Ability to compute posterior densities (combine prior with likelihood)?
- Ability to compare models via marginal likelihood?
- For me: the ability to integrate out model parameters completely...

Why be Bayesian?

- We're often not interested in parameter values
- We're normally interested in something that is a function of the parameter values e.g.:
  - Within Machine Learning (ML) we are often interested in making predictions (predicing y_* from $\mathbf{x}_*$).
  - This will often require values of some parameters $\mathbf{\Theta}$
  - Being Bayesian allows us to *average* over uncertainity in parameters when making predictions:

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

- This for me, is the biggest Bayesian selling point!

Lecture 3: Application: Touchscreen typing

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

May 11, 2014

Typing on touchscreens

- Most people have smartphones
- Most smartphones have touchscreens
- Touchscreens are small
- Keyboards on touchscreens are small
- Typing on them is hard!
  - . . . but people type on them a lot

Background 1: Why is it hard?

- Occlusion of target by finger
- 'fat finger' problem
- Small targets
- Demo: http://bit.ly/1nBws97

Background 1: Why is it hard?

- Occlusion of target by finger
- 'fat finger' problem
- Small targets
- Demo: http://bit.ly/1nBws97
- Quite a bit of work in this area:
  - Holz and Baudisch
  - Henze (100,000,000 taps)
- Collecting data is fairly easy

Background 2: All users are different

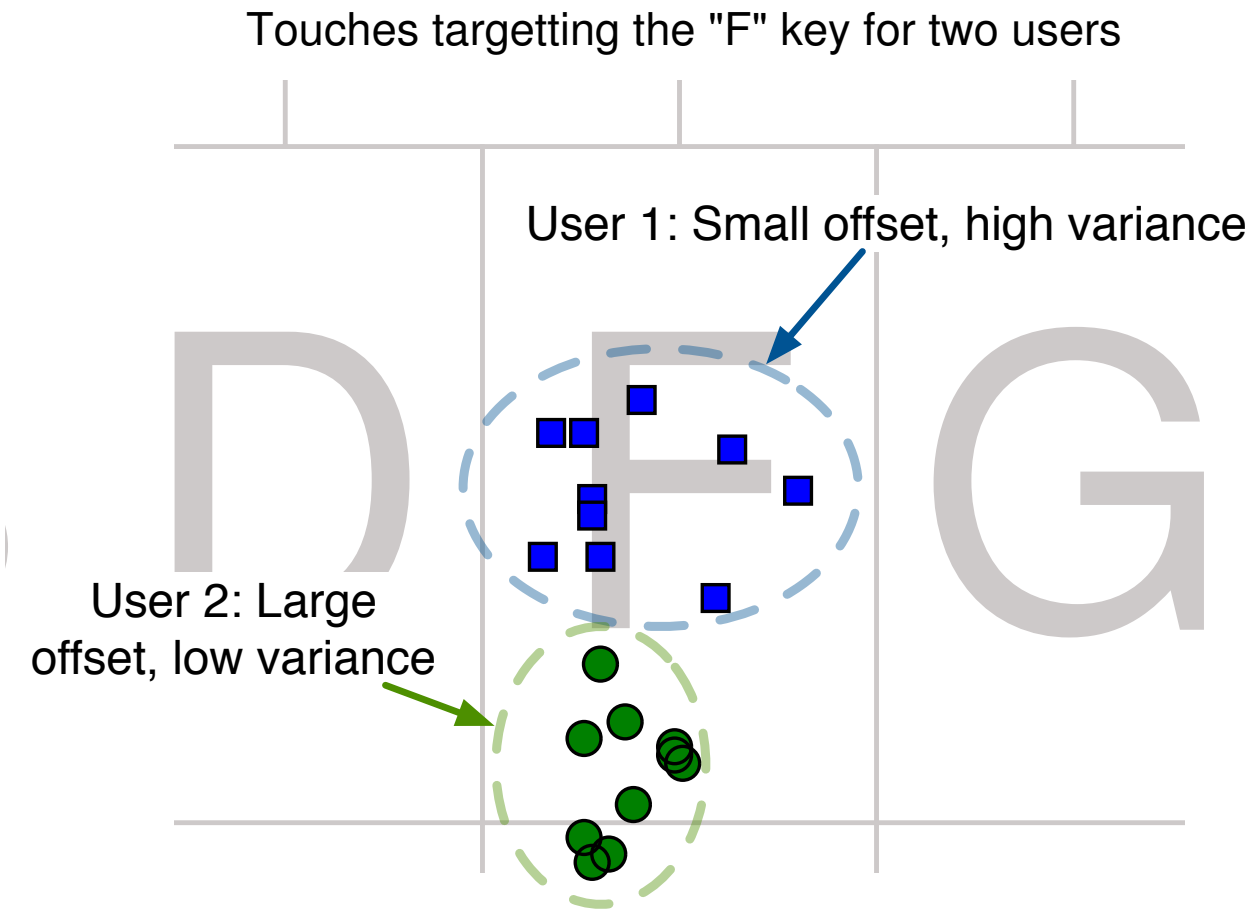


Figure 5 : Touches recorded by two users aiming for the 'F' key. User 2 has high bias and low variance, user 1 has low bias and high variance.

Background 3: Current systems (maybe?)

- Touch is boxed into nearest key.
- Key ID is passed to a Statistical Language Model (SLM).
- SLM is made up of probabilities of observing certain character strings (from large text corpora).
- SLM can swap characters to make the character string more likely.
  - e.g. 'HELLP $\rightarrow$ HELLO'

Our idea

- There is a lot of uncertainty present in touch (bias and variance)
- Boxing a touch into a key is probably bad
- Why can't we pass a *distribution* to the SLM?
  - Pass the uncertainty onwards
  - Being Bayesian!

Our idea

- There is a lot of uncertainty present in touch (bias and variance)
- Boxing a touch into a key is probably bad
- Why can't we pass a *distribution* to the SLM?
  - Pass the uncertainty onwards
  - Being Bayesian!
- Can use a user specific GP regression model to predict target from input touch.

The model

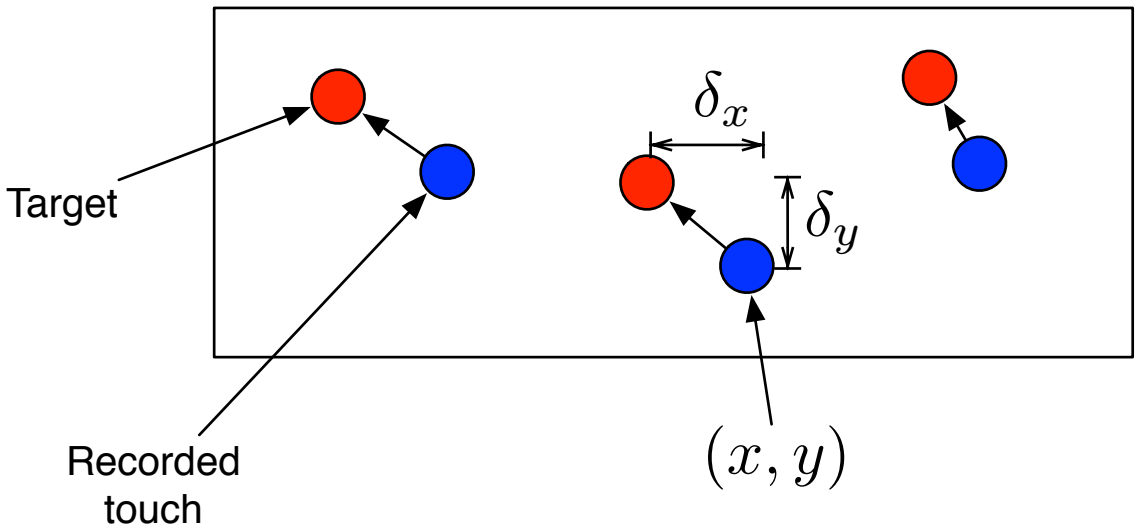
- We use independent GP regressions for predicting x and y offsets.
- Training data:
  - Each user typed phrases provided to them.
  - Data: the x, y location of the recorded touch (i.e. $\mathbf{x}_n = [x_n, y_n]^T$). Target: the center of the intended key minus the touch (i.e. the offset).

The model

- We use independent GP regressions for predicting x and y offsets.
- Training data:
  - Each user typed phrases provided to them.
  - Data: the x, y location of the recorded touch (i.e. $\mathbf{x}_n = [x_n, y_n]^T$). Target: the center of the intended key minus the touch (i.e. the offset).
- Used a GP with zero mean and a composite covariance:

$$C(\mathbf{x}_1, \mathbf{x}_2) = a\mathbf{x}_1^T\mathbf{x}_2 + (1-a)\exp\{-\gamma||\mathbf{x}_1 - \mathbf{x}_2||^2\}$$

The model



$$\boldsymbol{\delta}_x \sim GP(\mathbf{0}, \mathbf{C})$$
$$\boldsymbol{\delta}_y \sim GP(\mathbf{0}, \mathbf{C})$$
$$\mathbf{C}_{nm} = f((x_n, y_n), (x_m, y_m))$$

- x and y offsets both depend on x and y position of touch
- Have also used raw capacitive sensor data as input
  - Potentially better (more info) but only accessible on some devices

System cartoon

Figure 6 : Train GPs to predict the intended touch from an input touch. The flexibility of GPs means that the mean and covariance of the offset can vary across the keyboard.

System cartoon

Figure 6 : Train GPs to predict the intended touch from an input touch. The flexibility of GPs means that the mean and covariance of the offset can vary across the keyboard.

System cartoon

Figure 6 : Train GPs to predict the intended touch from an input touch. The flexibility of GPs means that the mean and covariance of the offset can vary across the keyboard.

System cartoon

Figure 6 : Train GPs to predict the intended touch from an input touch. The flexibility of GPs means that the mean and covariance of the offset can vary across the keyboard.

System cartoon

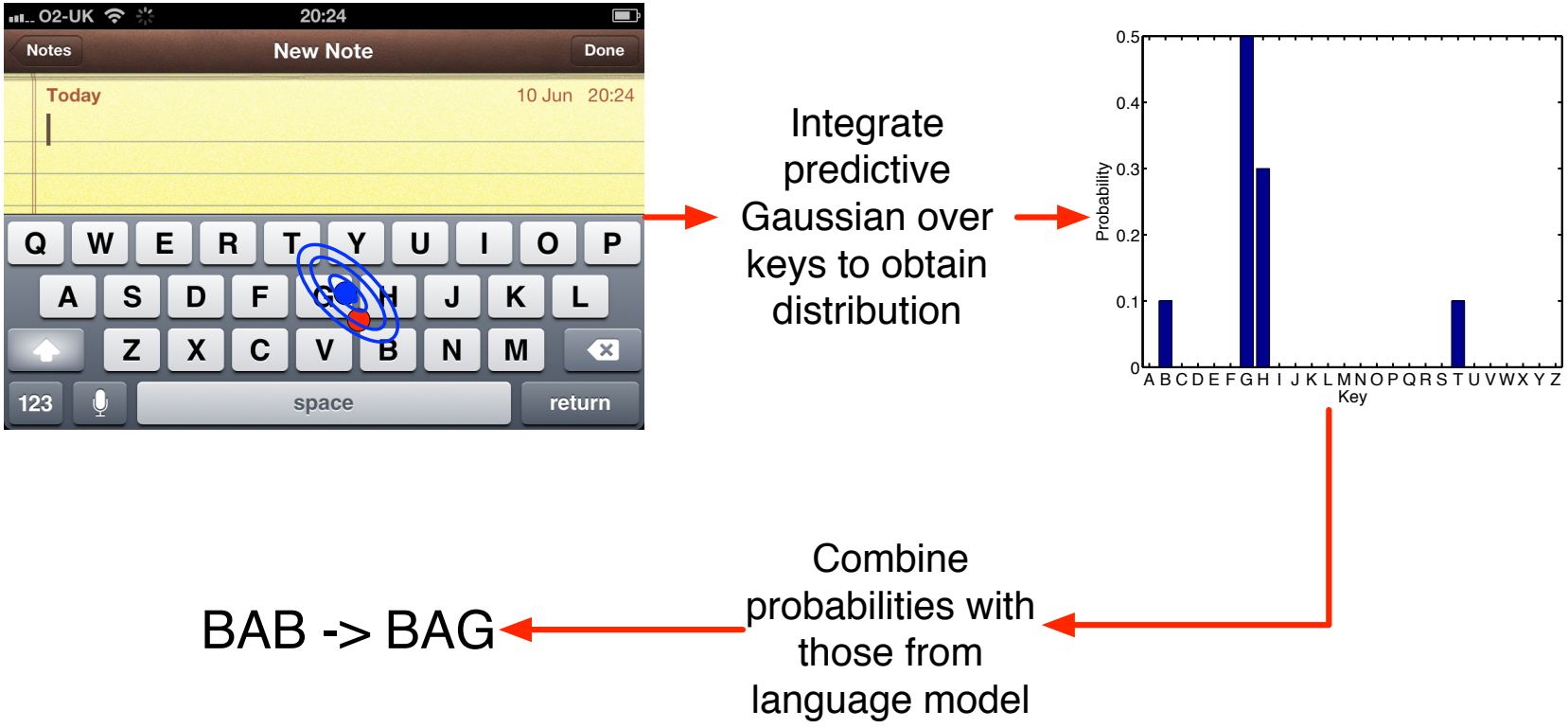


Figure 7 : The complete system

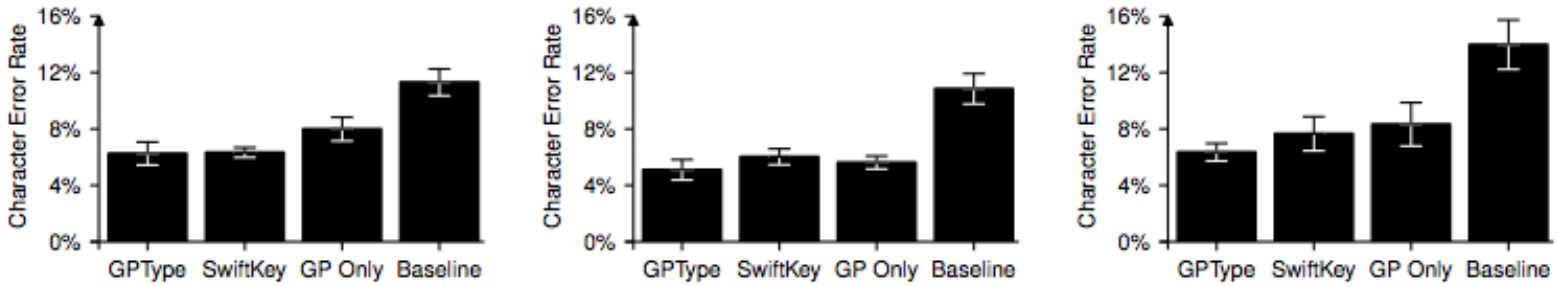
Video

- http://www.youtube.com/watch?v=llQI5gV5l74

The experiment

- 10 participants
- Calibration data collected for each
  - Note: calibration task matters
- each did 3$\times$ 45 minute sessions, typing whilst sitting, standing and walking. [more details in paper]
- Compared:
  - GPtype (our system), Swiftkey (commercial Android keyboard), GP only (just offset, no SLM), baseline (boxing, no SLM).

Results



(a) Sitting (b) Standing (c) Walking

Figure 4. Character error rates for the two keyboards we evaluated, separated by mobility condition (Study 2). Plots show mean and standard error across all participants. The baseline method represents the literal keys touched, while GP Only shows the keys hit after the mean GP offset is applied.

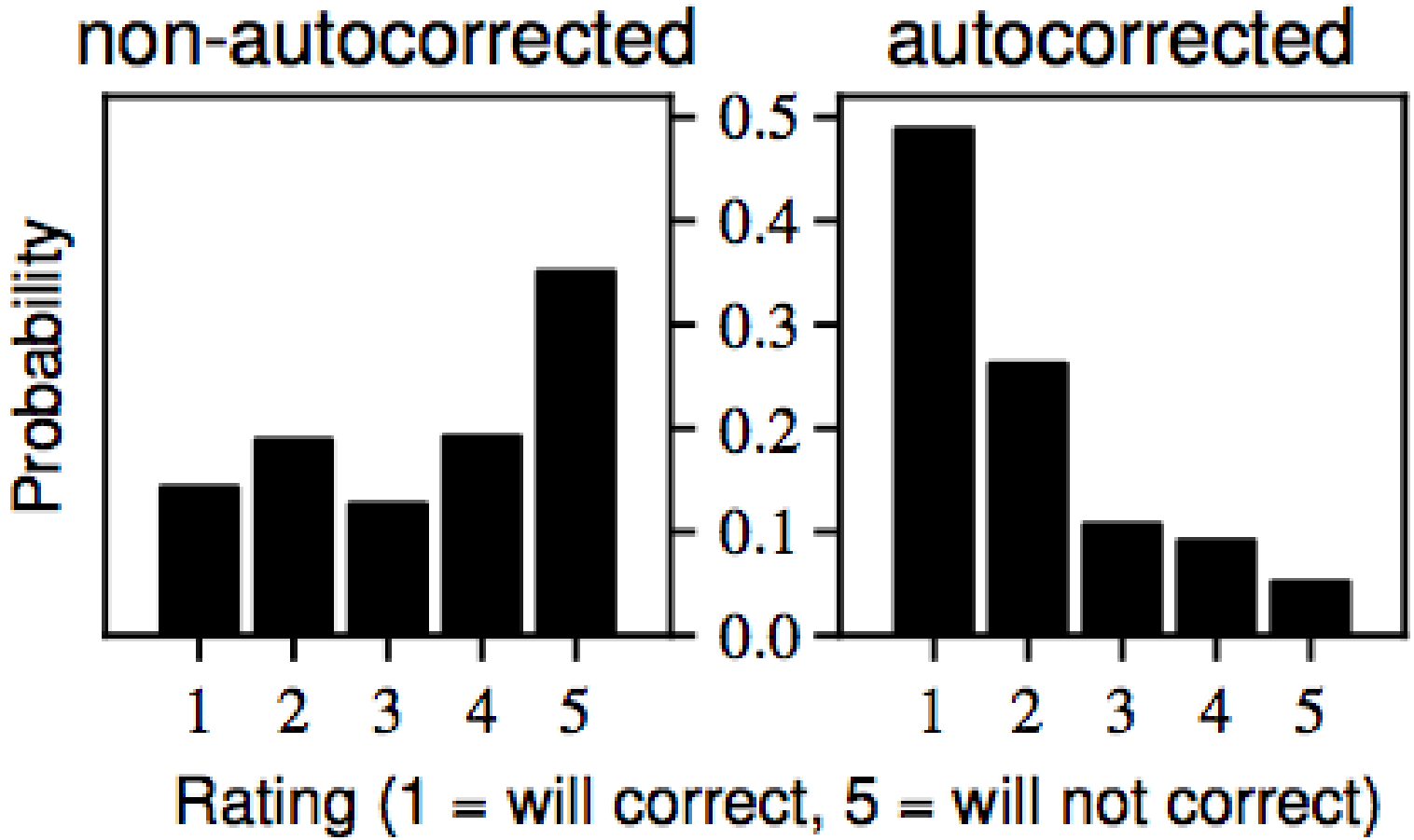
Figure 8 : Results of GPType experiment

- GPType marginally (stat sig) better than Swiftkey.
  - A **lot** of people work on SwiftKey
- Baseline awful!

Explicit uncertainity control

- In GPType, uncertainity is handled implicitly
- As user typing becomes more uncertain, more power given to language model
- Could users *explicitly* control this?
  - Certain inputs: no SLM control (slang, names, etc)
  - Uncertain inputs: high SLM control
- Use pressure to control certainty:
  - High pressure: high certainty
  - Low pressure: low certainty

Do users know when SLM will fail?



- Users given phrases and asked whether they thought autocorrect would change them incorrectly
- Users quite good at understanding SLM failings

ForceType

- Modified Synaptics Forcepad
- Pressure mapped to Gaussian variance (no GP)
- System explained to users
- Users type phrases with and without forcetype

ForceType: Results

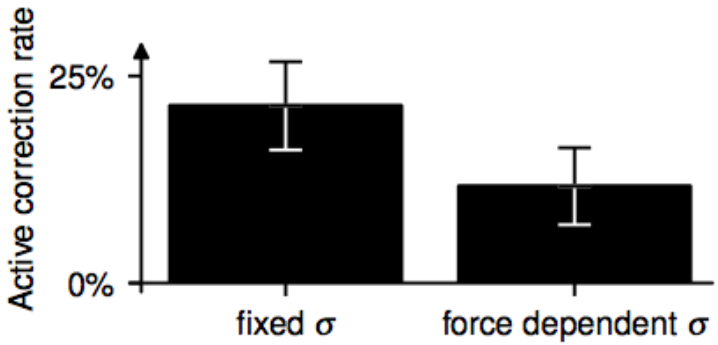


Figure 8. ForceType requires significantly fewer active corrections from users when entering text. Required corrections dropped by ≈ 10 percentage points. Errors bars are 1 sd.

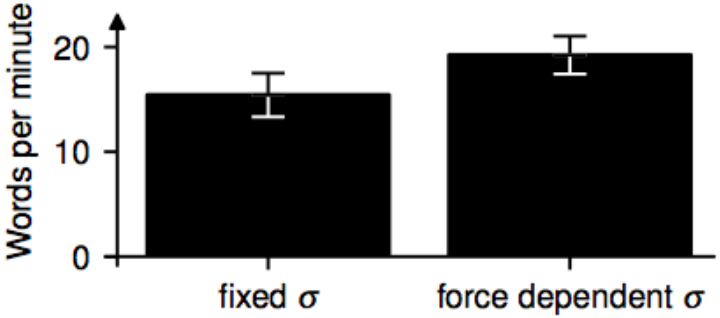


Figure 9. ForceType enabled users to enter phrases > 20% faster. A significant increase over the baseline. Errors bars are 1 sd.

- Forcetype reduced number of corrections performed by users (top)
- Forcetype improved overall text entry rate

Conclusions

- GP regression is key to the approach: we make no parametric assumptions (what would they be?)
- . . . and get probabilistic predictions
- . . . that can be fed to the SLM – (un)certainity is passed to the SLM
- Performance is promising

Conclusions

- GP regression is key to the approach: we make no parametric assumptions (what would they be?)
- . . . and get probabilistic predictions
- . . . that can be fed to the SLM – (un)certainity is passed to the SLM
- Performance is promising
- Can also use pressure to provide explicit uncertainty control

Conclusions

- GP regression is key to the approach: we make no parametric assumptions (what would they be?)
- . . . and get probabilistic predictions
- . . . that can be fed to the SLM – (un)certainity is passed to the SLM
- Performance is promising
- Can also use pressure to provide explicit uncertainty control
- More info:
  - http://www.youtube.com/watch?v=llQI5gV5l74
  - http://pokristensson.com/pubs/WeirEtAlCHI2014.pdf
  - Acknowledgements: Daryl Weir, Per Ola Kristensson, Keith Vertanen, Henning Pohl

Lecture 4: GPs for classification and ordinal regression via the auxiliary variable trick

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

May 11, 2014

GPs for Classification and ordinal regression

- What if our observation model is non-Gaussian?
  - Classification:

$$P(y_n = 1|f_n) = \int_{-\infty}^{f_n} \mathcal{N}(z|0,1)\ dz = \phi(f_n)$$

  - Logistic Regression:

$$P(y_n = k|f_n) = \phi(b_{k+1}) - \phi(b_k)$$

  - etc
- Analytical inference is no longer possible
- I'll cover how to do inference in these models and extensions with the *auxiliary variable trick*

Binary classification

- Problem setup: we observe N data / target pairs $(\mathbf{x}_n, y_n)$ where $y_n \in \{0, 1\}$
- Place a GP prior on a set of latent variables f_n

$$\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$$

- Use the probit likelihood:

$$P(y_n = 1|f_n) = \phi(f_n) = \int_{-\infty}^{f_n} \mathcal{N}(z|0, 1)\ dz$$

- Inference in this form is hard

Auxiliary Variable Trick

- Re-write the probit function:

$$
\begin{aligned}
P(y_n = 1|f_n) &= \int_{-\infty}^{f_n} N(z|0,1)\ dz \\
&= \int_{-\infty}^{0} N(z|-f_n,1)\ dz \\
&= \int_{0}^{\infty} N(z|f_n,1)\ dz \\
&= \int_{-\infty}^{\infty} \delta(z>0)\mathcal{N}(z|f_n,1)\ dz
\end{aligned}
$$

where $\delta(expr)$ is 1 if $expr$ is true, and 0 otherwise.

Auxiliary Variable Trick

- If we define $P(y_n = 1|z_n) = \delta(z_n > 0)$ then we have:

$$P(y_n = 1|f_n) = \int_{-\infty}^{\infty} P(y_n = 1|z_n)p(z_n|f_n)\ dz_n$$

- and could therefore remove the integral to obtain a model including z_n:

$$p(y_n = 1, z_n|f_n) = P(y_n = 1|z_n)p(z_n|f_n)$$

- Doing inference in this model (i.e. with additional variables z_n) is much easier (but still not analytically tractable)
- Note: $P(y_n = 0|z_n) = \delta(z_n < 0)$

Example - Gibbs sampling for binary classification

- An easy way to perform inference in the augmented model is via Gibbs sampling
- Sample $z_n|f_n, y_n$:

$$\begin{aligned} p(z_n|f_n, y_n = 0) &\propto \delta(z_n < 0)\mathcal{N}(z_n|f_n, 1) \\ p(z_n|f_n, y_n = 1) &\propto \delta(z_n < 1)\mathcal{N}(z_n|f_n, 1) \end{aligned}$$

Example - Gibbs sampling for binary classification

- An easy way to perform inference in the augmented model is via Gibbs sampling
- Sample $z_n|f_n, y_n$:

$$\begin{aligned} p(z_n|f_n, y_n = 0) &\propto \delta(z_n < 0)\mathcal{N}(z_n|f_n, 1) \\ p(z_n|f_n, y_n = 1) &\propto \delta(z_n < 1)\mathcal{N}(z_n|f_n, 1) \end{aligned}$$

- Sample $\mathbf{f}|\mathbf{z}, \mathbf{C}$

$$p(\mathbf{f}|\mathbf{z}, \mathbf{C}) = \mathcal{N}(\boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f)$$

where

$$\boldsymbol{\Sigma}_f = \left(\mathbf{I} + \mathbf{C}^{-1}\right)^{-1}, \quad \boldsymbol{\mu}_f = \boldsymbol{\Sigma}_f^{-1}\mathbf{z}$$

- Repeat ad infinitum

Example - Gibbs sampling for binary classification

- To make predictions:
  - At each sampling step, do a (noise-free) GP regression using the current sample of $\mathbf{f}$ to get a density over f_* (Details in a previous slide).
  - Sample a specific realisation of f_* from this density.
  - Compute $\phi(f_*)$ (or sample a z_* and then record whether it's > 0 or not)
  - Average this value over all Gibbs sampling iterations!

Example - binary classification

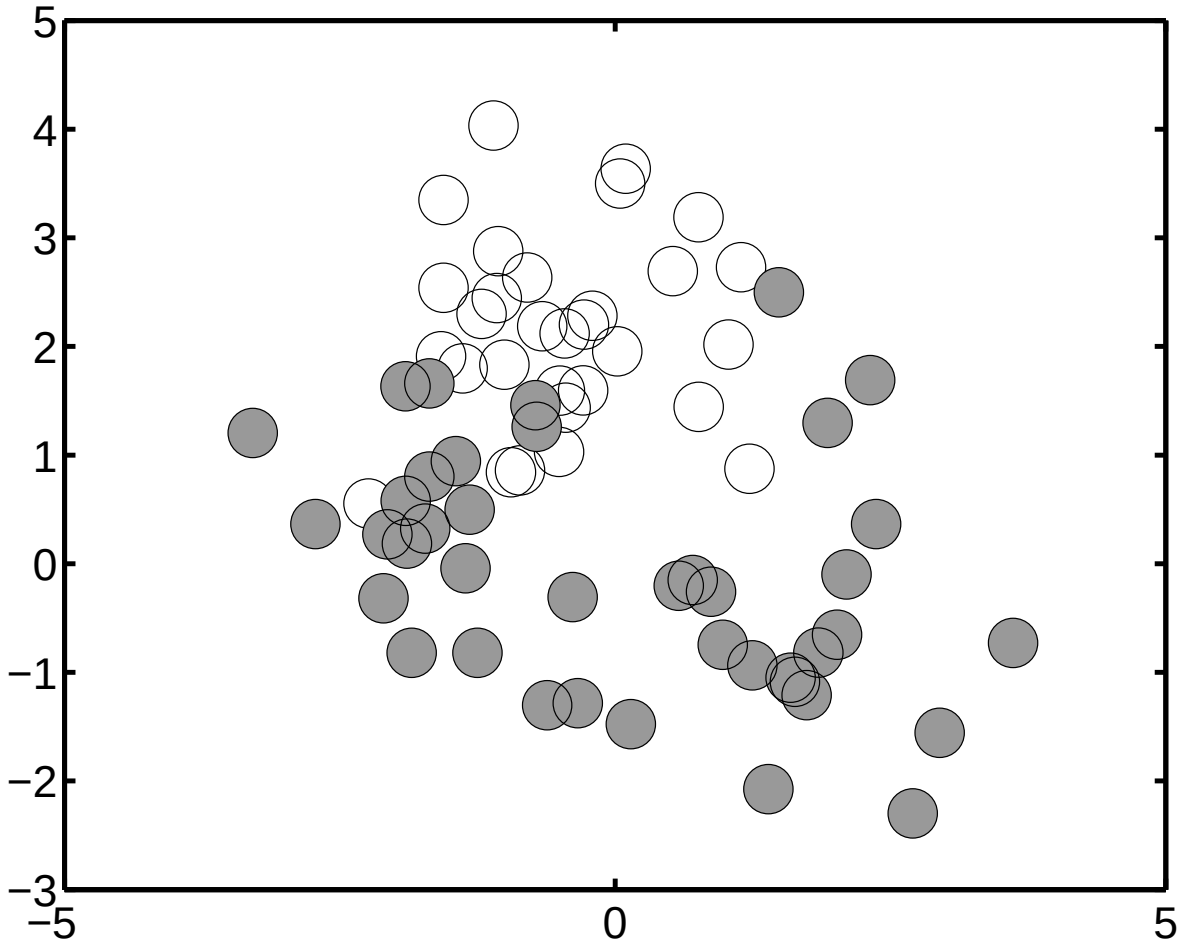


Figure 9 : Some simple classification data

Example - binary classification

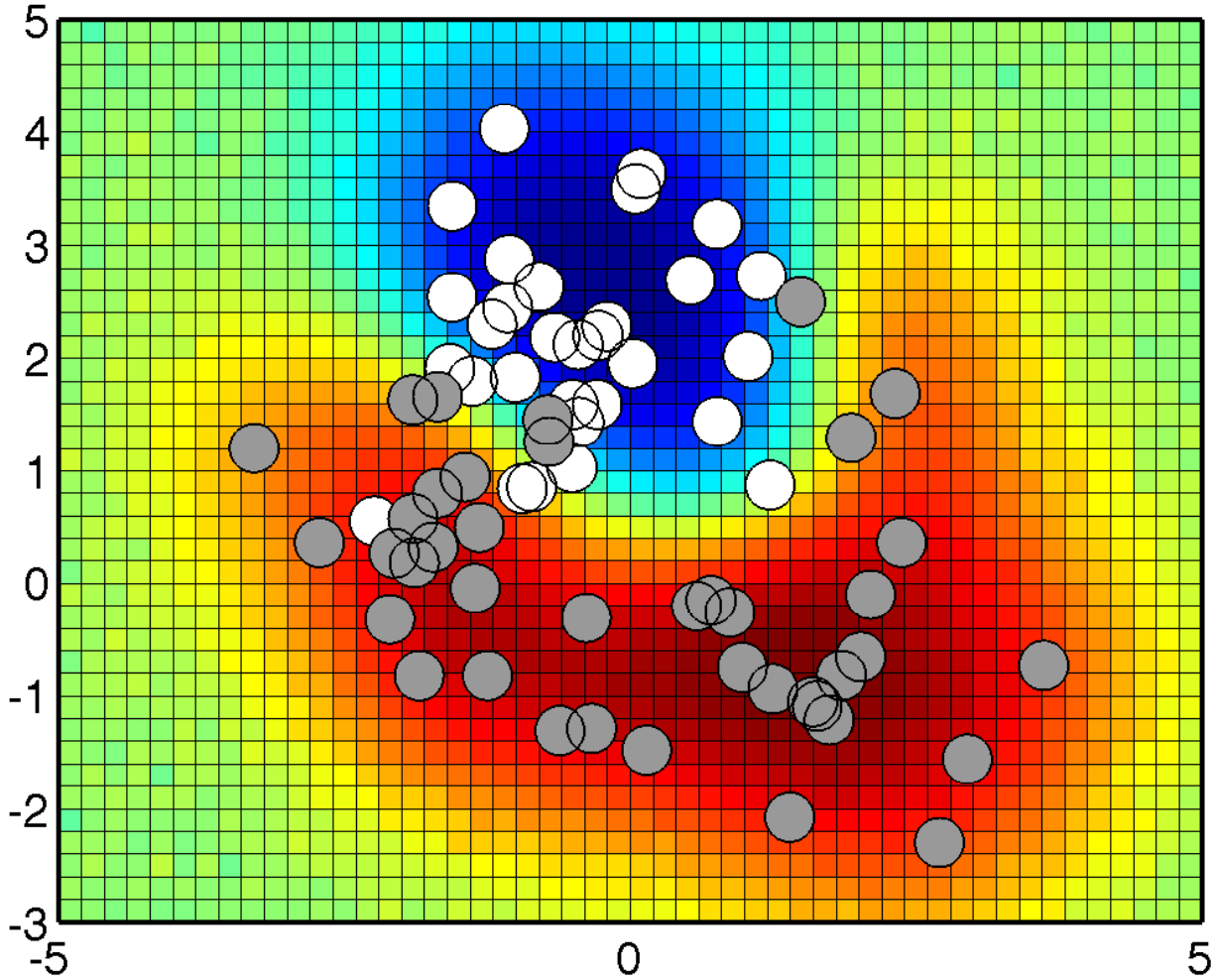


Figure 10 : Predictive probabilities averaged over 1000 Gibbs samples using an RBF covariance. As γ is increased, the model overfits.

Example - binary classification

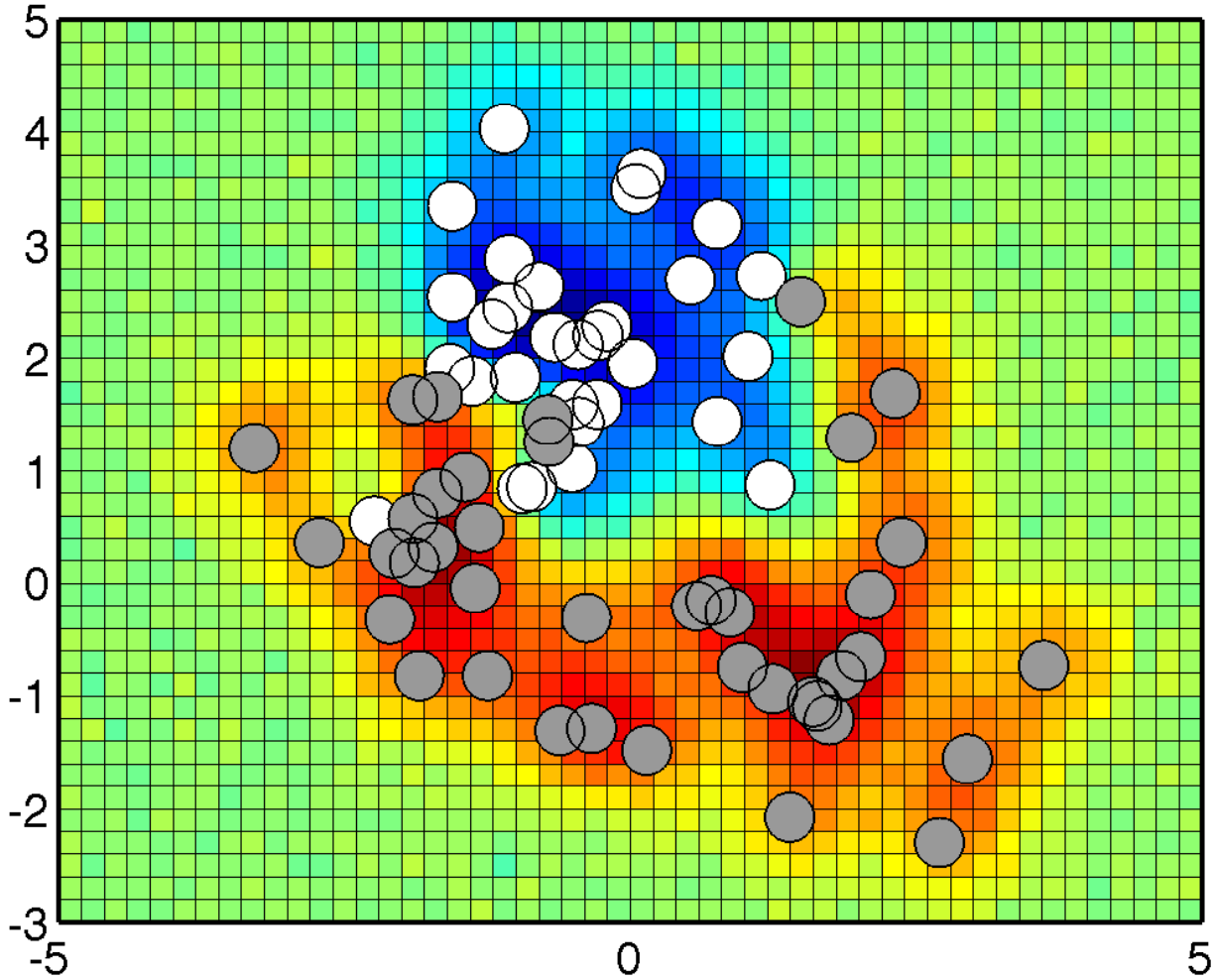


Figure 10 : Predictive probabilities averaged over 1000 Gibbs samples using an RBF covariance. As γ is increased, the model overfits.

Example - binary classification

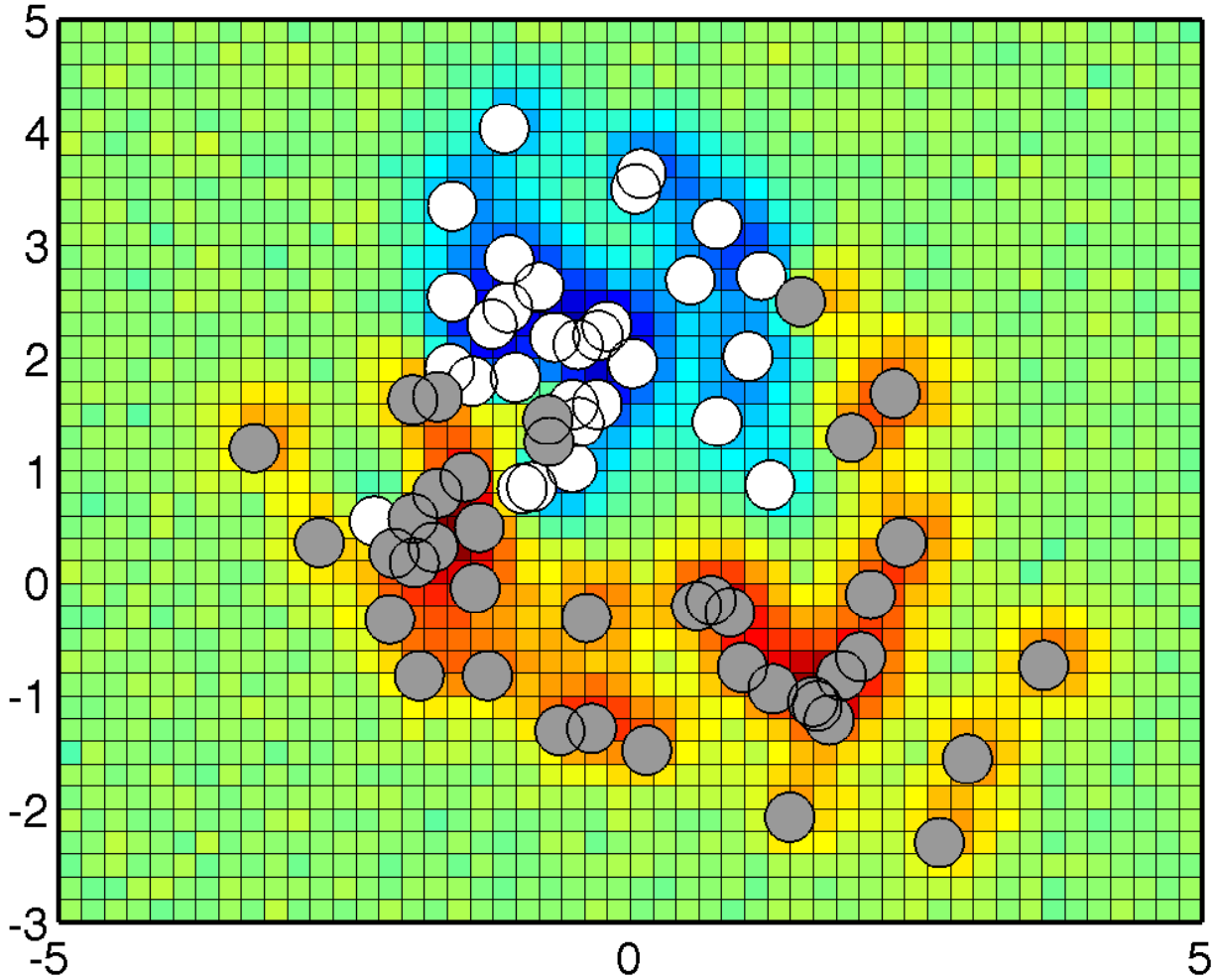


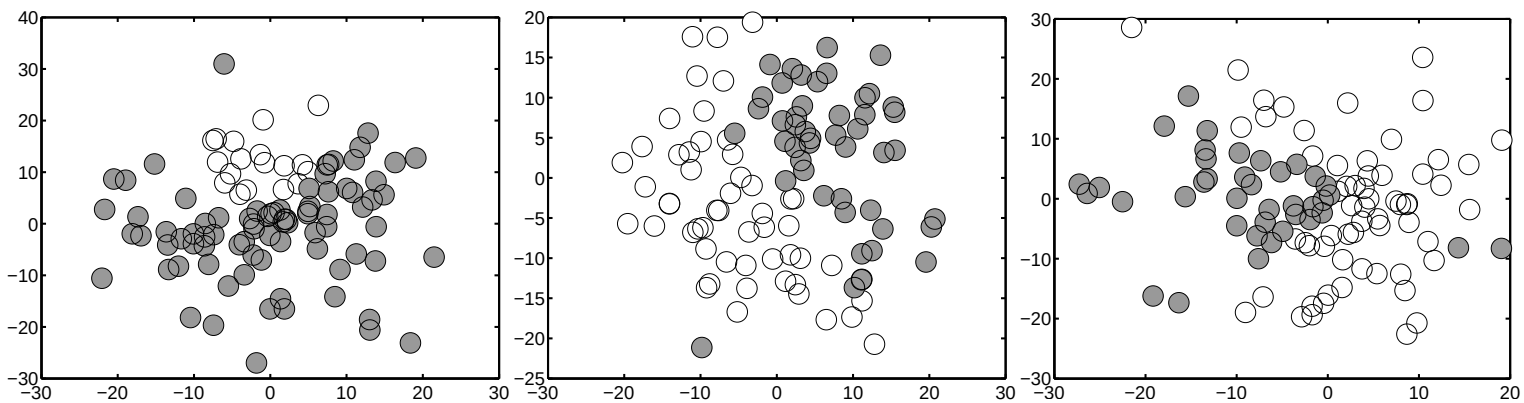
Figure 10 : Predictive probabilities averaged over 1000 Gibbs samples using an RBF covariance. As γ is increased, the model overfits.

Note

- Inference:
  - Gibbs sampling isn't the only option
  - A popular alternative is Variational Bayes

Note 2 – The Generative Process

- Sometimes it's useful to think of the generative process defined by the model.
- In this case, to generate N values of y_n given the associated x_n:
  - Sample $\mathbf{f}$ from a GP with mean $\mathbf{0}$ and Covariance matrix $\mathbf{C}$.
  - For each $n = 1 \ldots N$:
    - Sample $z_n \sim \mathcal{N}(f_n, 1)$
    - If $z_n > 0$ set $y_n = 1$, otherwise $y_n = 0$.
- Some examples:



GP classification exercise

TASK [2]

- Explore GP binary classification with auxiliary variables using `gp_class_task.m`
- Try:
  - Generating data from different distributions
  - Varying covariance function and parameters
  - Taking more posterior samples
- You will also need `plotClassdata.m` and `kernel.m`

A more general idea

- Models of this form:
  - $\mathbf{f} \sim GP$
  - $z_n \sim \mathcal{N}(f_n, 1)$
  - $P(y_n|z_n) = \delta(f(z_n))$
- Can be used for more than just binary classification.

A more general idea

- Models of this form:
  - $\mathbf{f} \sim GP$
  - $z_n \sim \mathcal{N}(f_n, 1)$
  - $P(y_n|z_n) = \delta(f(z_n))$
- Can be used for more than just binary classification.
- Ordinal Regression:
  - $P(y_n = k|z_n)$ is now chopped at both ends:

$$P(y_n = k|z_n) = \delta(b_k < z_n < b_{k+1})$$

  - Gibbs distribution for z_n therefore involves a Gaussian truncated at both ends.

A more general idea

- Models of this form:
  - $\mathbf{f} \sim GP$
  - $z_n \sim \mathcal{N}(f_n, 1)$
  - $P(y_n|z_n) = \delta(f(z_n))$
- Can be used for more than just binary classification.
- Ordinal Regression:
  - $P(y_n = k|z_n)$ is now chopped at both ends:

$$P(y_n = k|z_n) = \delta(b_k < z_n < b_{k+1})$$

  - Gibbs distribution for z_n therefore involves a Gaussian truncated at both ends.
- As well as multi-class and semi-supervised classification...

Multi-class classification

- The previous treatment can be extended to multiple classes.
- For a problem with K classes:
  - K GP priors, K N-dimensional latent vectors $\mathbf{f}_k$.
  - $N \times K$ auxiliary variables $z_{nk} \sim \mathcal{N}(f_{nk}, 1)$
  - And:

$$P(y_n = k | z_{n1}, \ldots, z_{nK}) = \delta(z_{nk} > z_{ni} \quad \forall i \neq k)$$

Multi-class classification

- The previous treatment can be extended to multiple classes.
- For a problem with K classes:
  - K GP priors, K N-dimensional latent vectors $\mathbf{f}_k$.
  - $N \times K$ auxiliary variables $z_{nk} \sim \mathcal{N}(f_{nk}, 1)$
  - And:

$$P(y_n = k | z_{n1}, \ldots, z_{nK}) = \delta(z_{nk} > z_{ni} \quad \forall i \neq k)$$

- Gibbs sampling is similar to the binary case:
  - Only tricky bit is efficently sampling from a K-dimensionl MVG truncated such that the kth element is largest.

Multi-class classification

- The previous treatment can be extended to multiple classes.
- For a problem with K classes:
  - K GP priors, K N-dimensional latent vectors $\mathbf{f}_k$.
  - $N \times K$ auxiliary variables $z_{nk} \sim \mathcal{N}(f_{nk}, 1)$
  - And:

$$P(y_n = k | z_{n1}, \ldots, z_{nK}) = \delta(z_{nk} > z_{ni} \quad \forall i \neq k)$$

- Gibbs sampling is similar to the binary case:
  - Only tricky bit is efficently sampling from a K-dimensionl MVG truncated such that the kth element is largest.
- Details of a Variational Bayes inference scheme in: Girolami and Rogers 2006

Multi-class Example

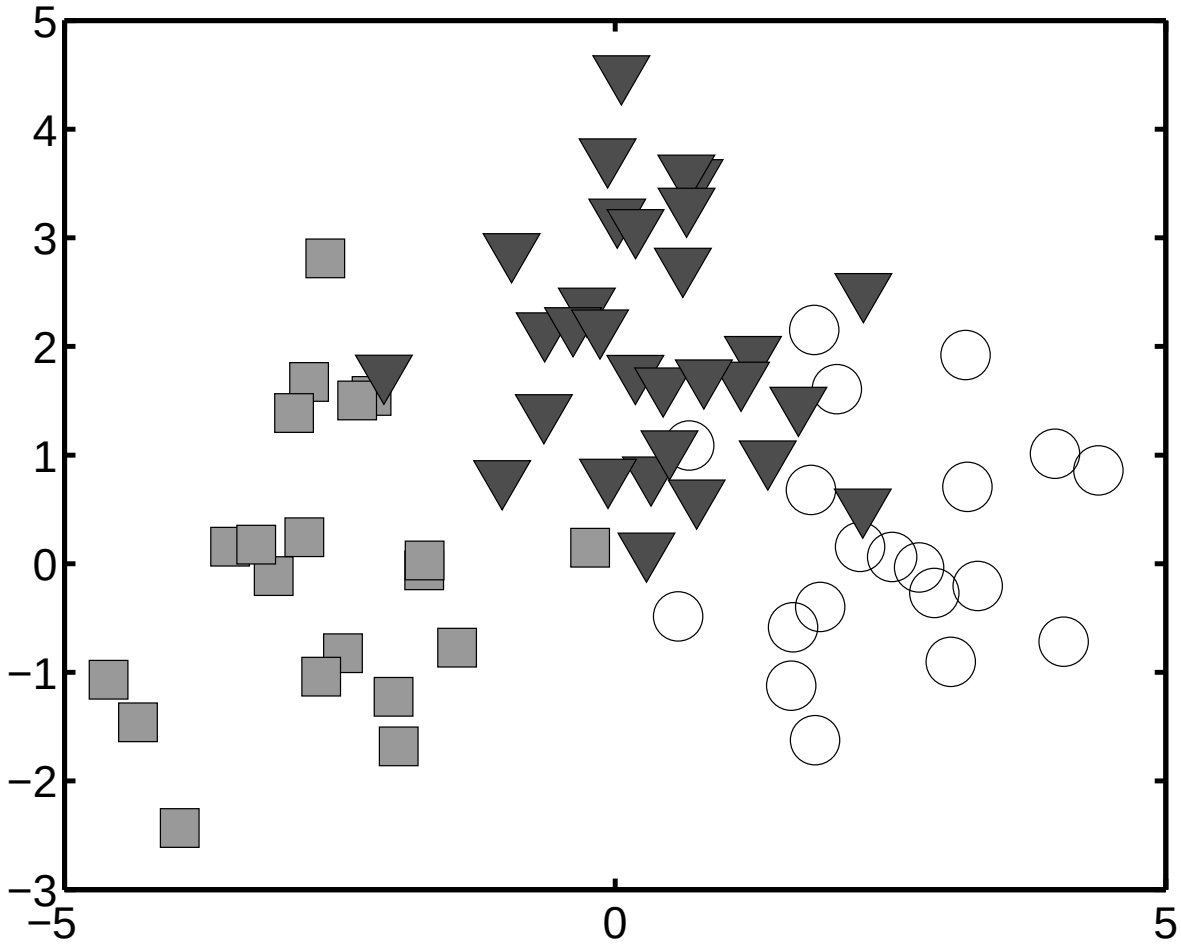


Figure 11 : Multi-class classification example. RBF covariance, $\gamma = 1$.

Multi-class Example

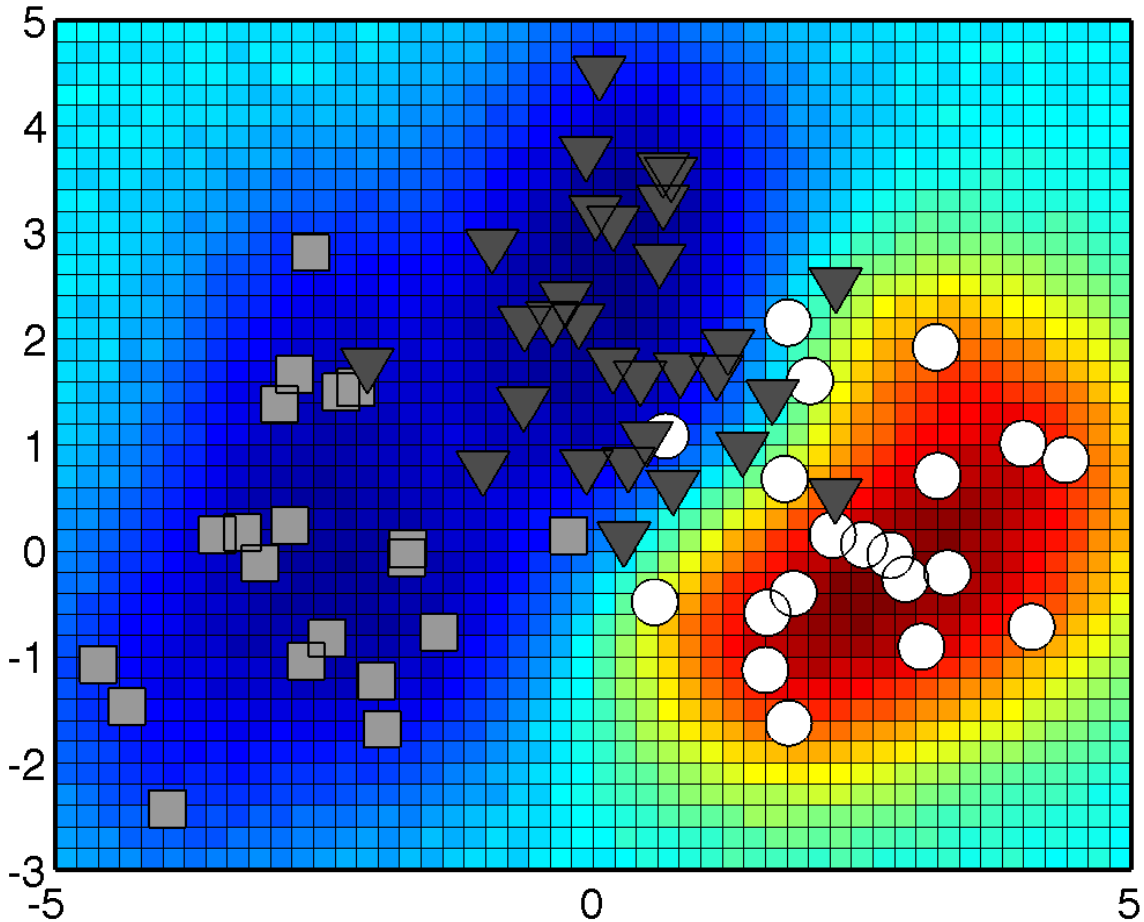


Figure 11 : Multi-class classification example. RBF covariance, $\gamma = 1$.

Multi-class Example

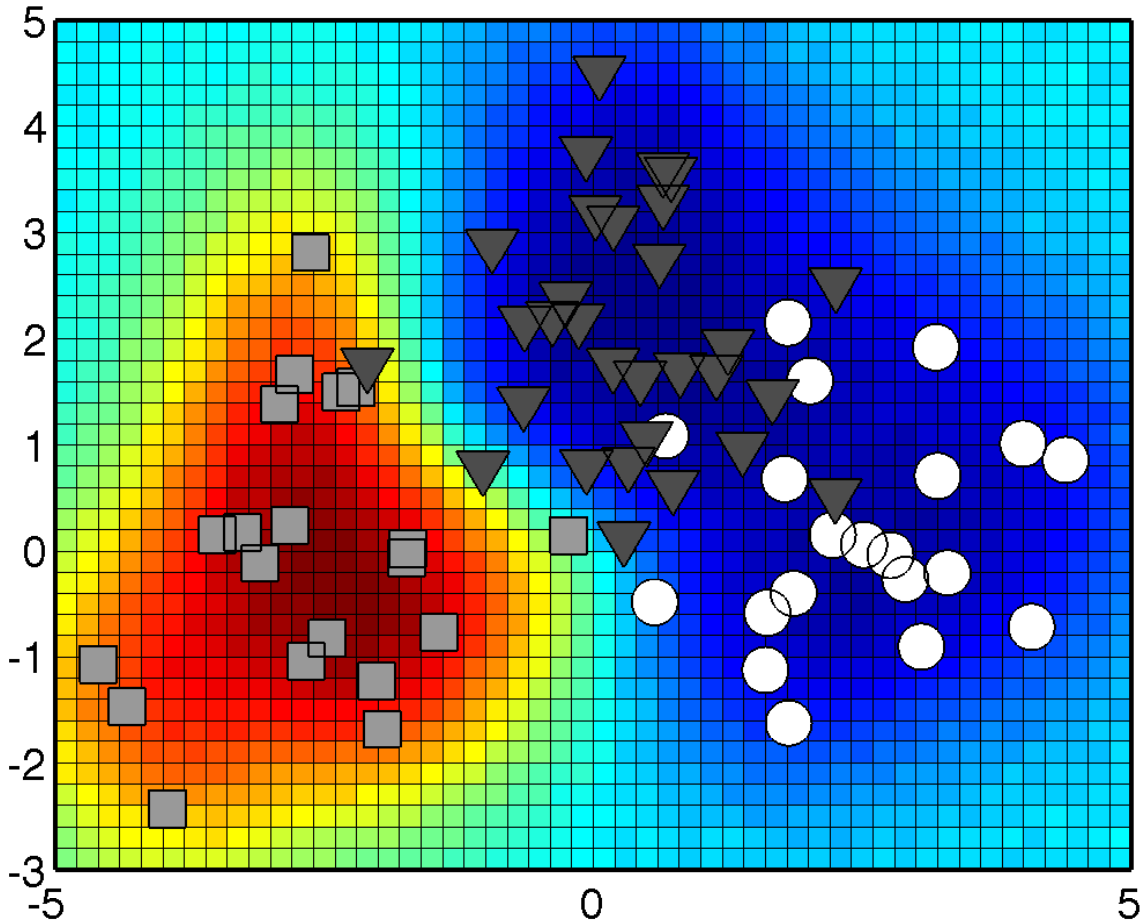


Figure 11 : Multi-class classification example. RBF covariance, $\gamma = 1$.

Multi-class Example

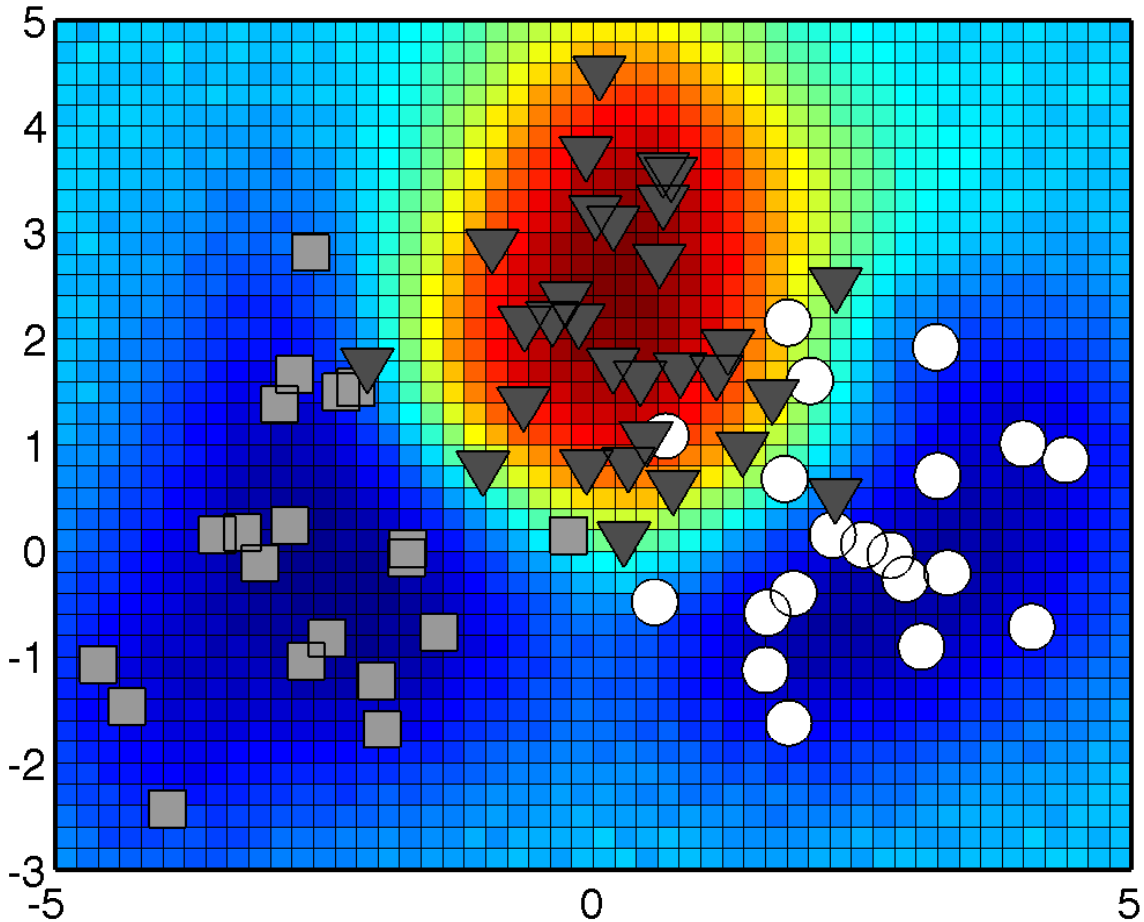


Figure 11 : Multi-class classification example. RBF covariance, $\gamma = 1$.

Semi-supervised Classification

- In some domains, only a subset of data are labeled [e.g. image classification]

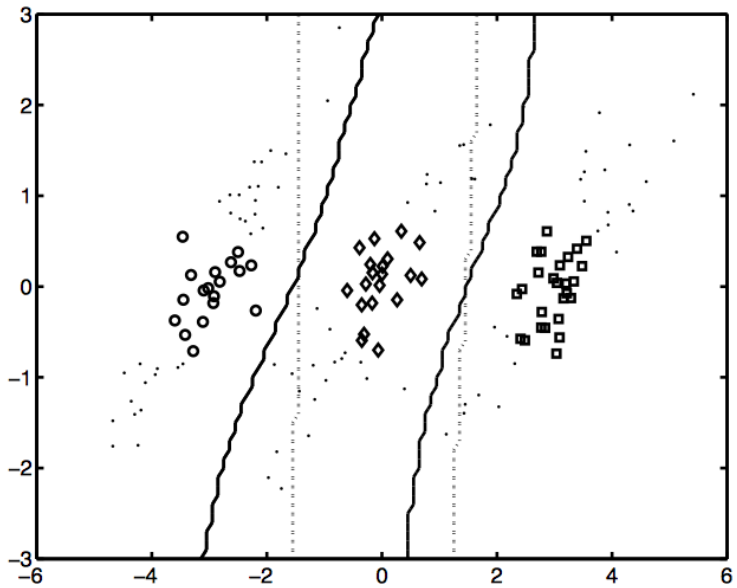


Figure 12 : A toy semi-supervised classification problem.

- Can be overcome using the Null Category Noise Model (NCNM) Lawrence and Jordan 2004

The NCNM

- Going back to binary classification, the auxiliary variable trick can be visualised:

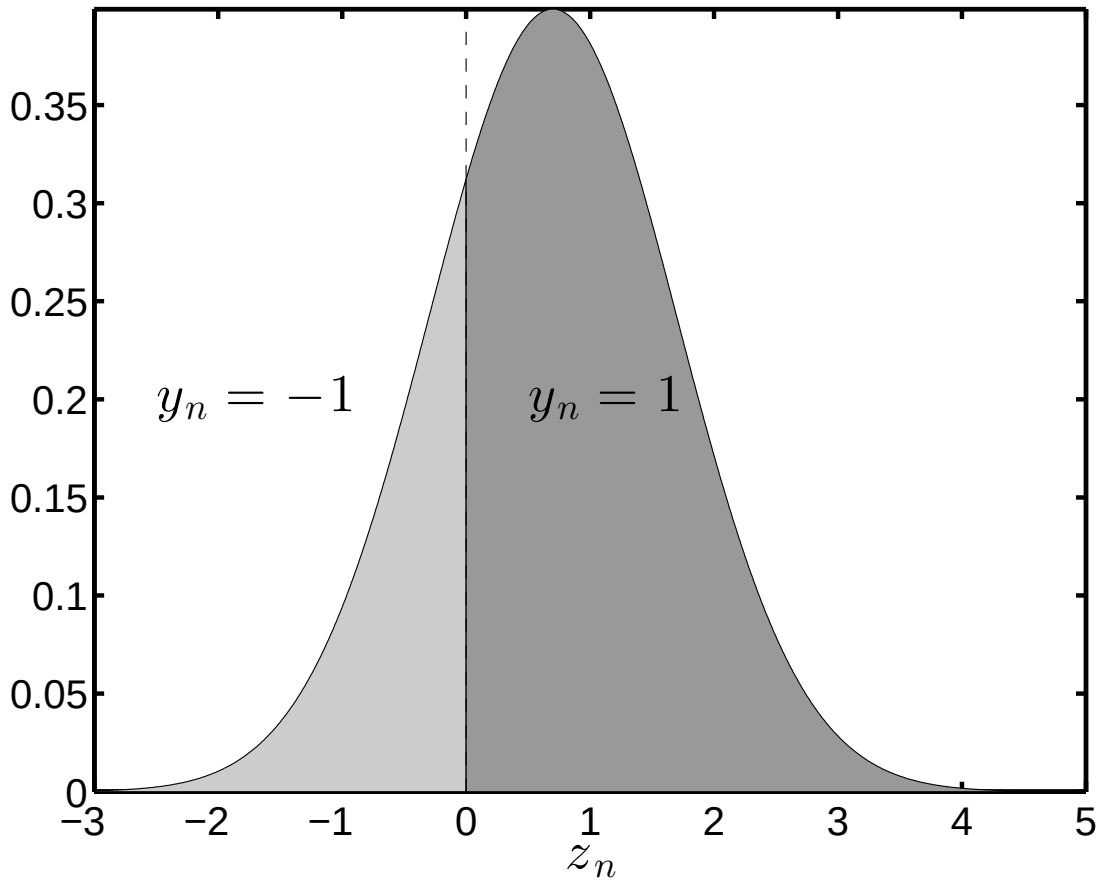


Figure 13 : Visualisation of the auxiliary variable trick. The Gaussian has mean f_n. Note that I'm not calling the classes ± 1.

The NCNM

- To include unlabeled data, we add a third category, for $y_n = 0$:

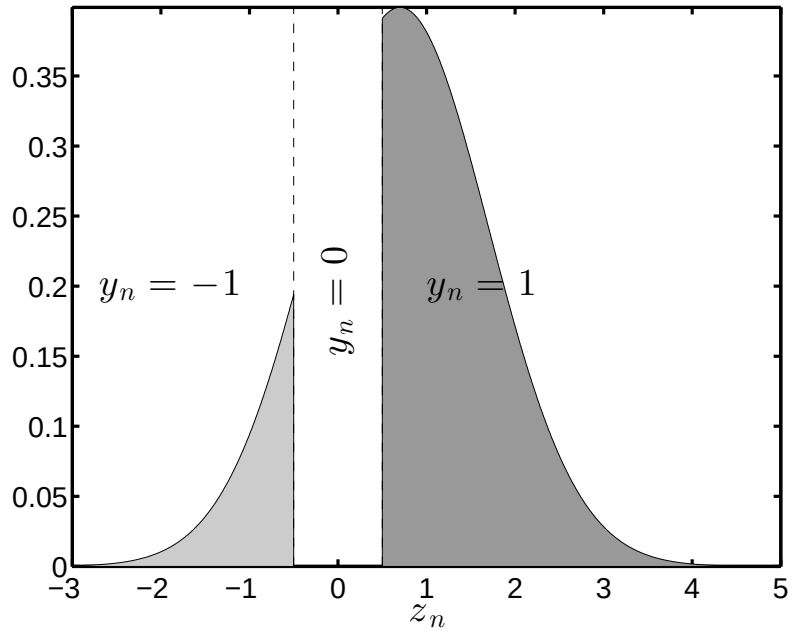


Figure 14 : Visualisation of the NCNM with a null region of width 1.

$$p(y_n|z_n) = \begin{cases} \delta(z_n < -a) & y_n = -1 \\ \delta(z_n > a) & y_n = 1 \\ \delta(z_n > -a) - \delta(z_n > a) & y_n = 0 \end{cases}$$

The NCNM

- The final step is to introduce another set of latent variables.
  - $g_n = 0$ if y_n is observed (i.e. labeled) and $g_n = 1$ otherwise.
- And enforce the constraint that no unlabeled points can exist in the null region:

$$P(y_n = 0 | g_n = 1) = 0$$

The NCNM

- The final step is to introduce another set of latent variables.
  - $g_n = 0$ if y_n is observed (i.e. labeled) and $g_n = 1$ otherwise.
- And enforce the constraint that no unlabeled points can exist in the null region:

$$P(y_n = 0 | g_n = 1) = 0$$

- This has the effect of introducing an empty region around the decision boundary
  - i.e. pushing the decision boundary into regions of empty space

The NCNM

- The final step is to introduce another set of latent variables.
  - $g_n = 0$ if y_n is observed (i.e. labeled) and $g_n = 1$ otherwise.
- And enforce the constraint that no unlabeled points can exist in the null region:

$$P(y_n = 0 | g_n = 1) = 0$$

- This has the effect of introducing an empty region around the decision boundary
  - i.e. pushing the decision boundary into regions of empty space
- Inference:
  - Gibbs sampling is the same as the binary case except $z_n | f_n, g_n = 1$.
  - This is a mixture of two truncated Gaussians – sample the component, and then sample z_n.

NCNM Example

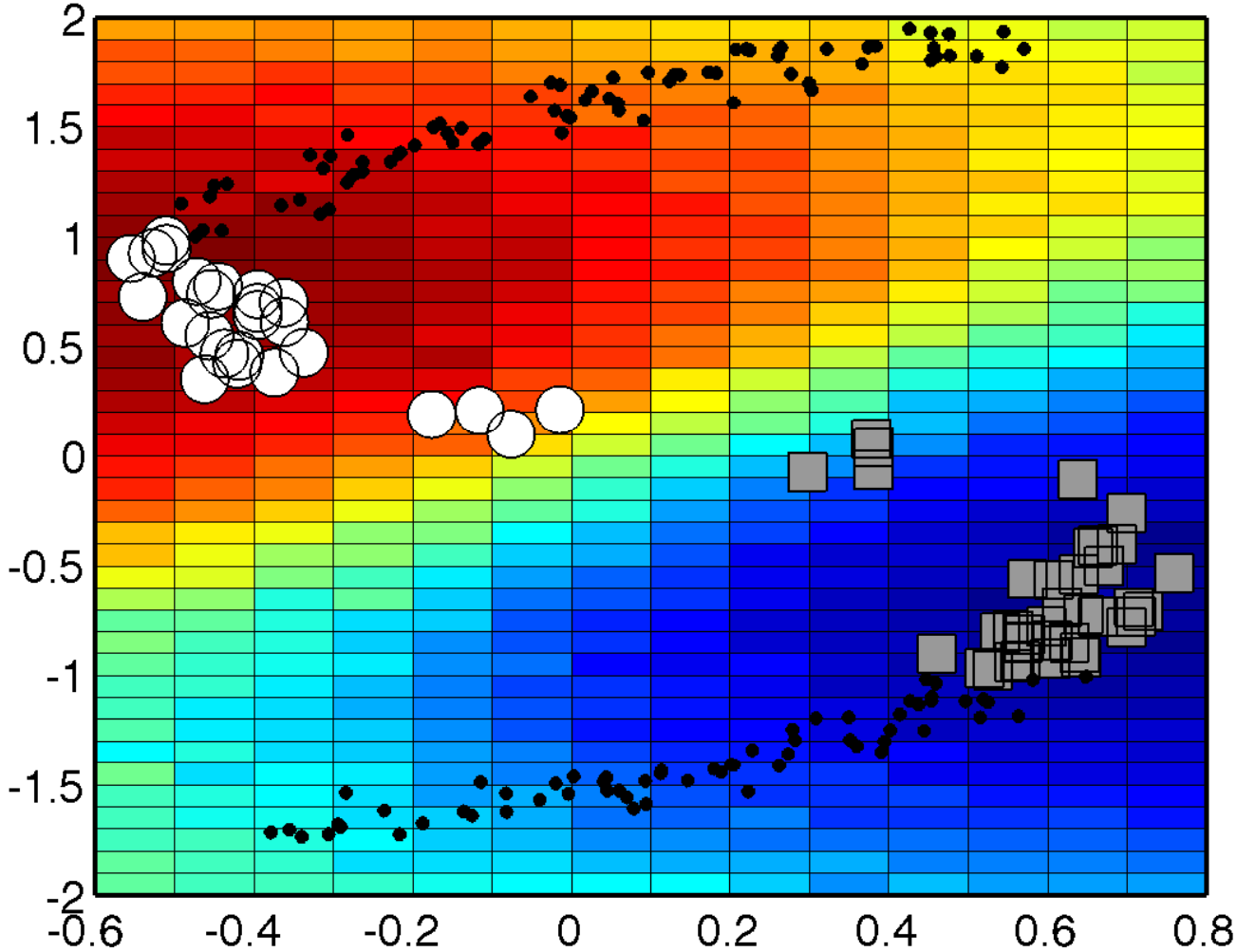


Figure 15 : Standard GP classification (unlabeled data ignored)

NCNM Example

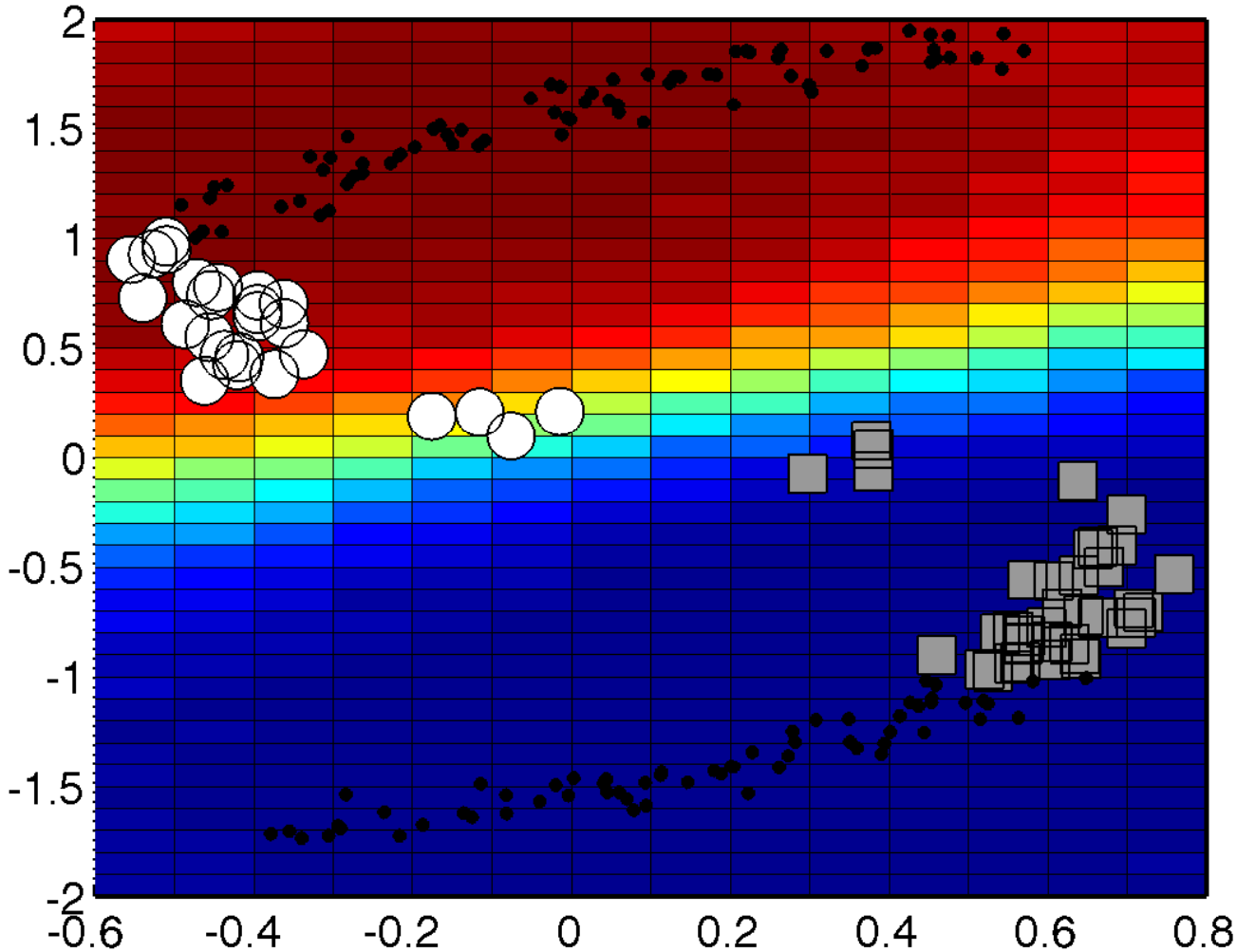


Figure 16 : NCNM GP classification

NCNM Exercise

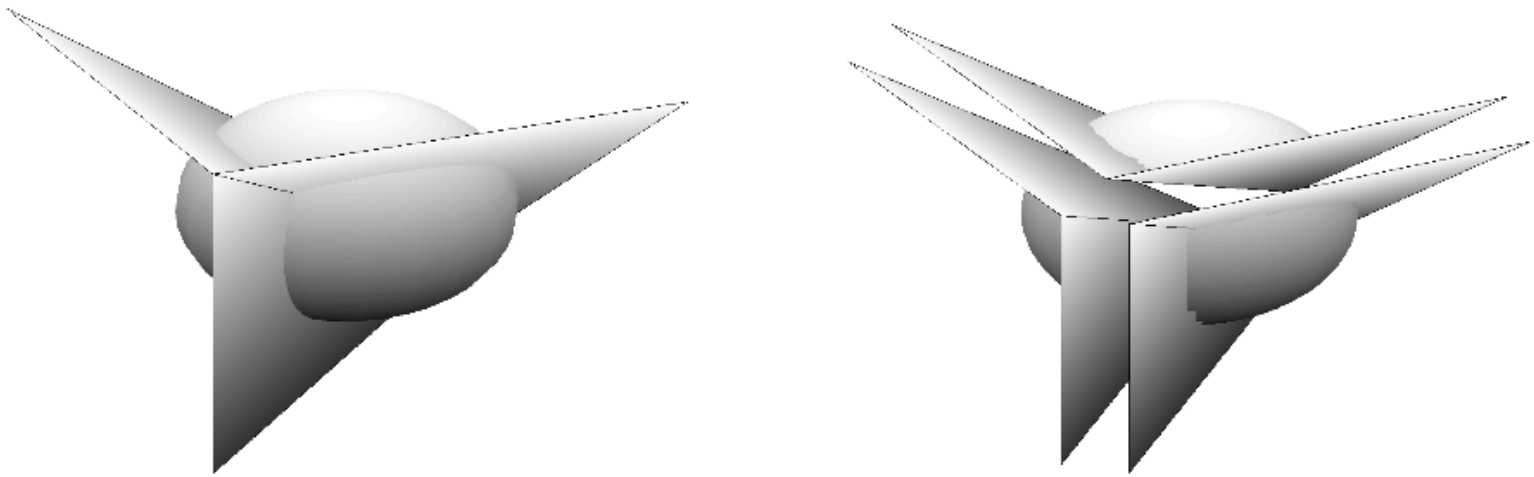
TASK [3]

- Experiment with the NCNM using `gp_ncnm_task.m`
- Setting `a=0` results in the standard model
- Setting `a>0` uses the NCNM
- It's not always easy to get the results you want to see!

Multi-class NCNM

- This idea can be extended to the multi-class setting.
- See Rogers and Girolami 2007

$$P(y_n = k|z_{n1}, \ldots, z_{nK}) = \begin{cases} \delta(z_{nk} > z_{ni} + a \quad \forall i \neq k) & y_n > 0 \\ 1 - \sum_j \delta(z_{nj} > z_{ni} + a \quad \forall i \neq j) & y_n = 0 \end{cases}$$



(a) A visualisation of the truncation caused by the standard multi-class probit model

(b) A visualisation of the truncation caused by the multi-class probit model with a null region

Figure 17 : Visualisation of truncation

Summary

- GP priors aren't restricted to regression.
- Analytical solutions aren't possible
- Auxiiliary Variable Trick makes inference (via Gibbs sampling or Variational Bayes) straightforward for:
  - Binary classification
  - Ordinal regression
  - Multi-class classification
  - Semi-supervised classification (binary and mutli-class)
  - As well as others (e.g. binary PCA)

Lecture 5: Application: Clinical Ratings

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

May 11, 2014

Clinicians disagree in AandE

- Patients in Accident and Emergency (A&E) are continually monitored.
  - Heart rate
  - Blood pressure
  - Temperature
  - etc

Clinicians disagree in AandE

- Patients in A&E are continually monitored.
  - Heart rate
  - Blood pressure
  - Temperature
  - etc
- Based on these hourly observations, clinicians (in a Glasgow hospital) give each patient an ordinal rating
  - A (healthy(ish)), B, C, D, E, F (critical)

Clinicians disagree in AandE

- Patients in A&E are continually monitored.
  - Heart rate
  - Blood pressure
  - Temperature
  - etc
- Based on these hourly observations, clinicians (in a Glasgow hospital) give each patient an ordinal rating
  - A (healthy(ish)), B, C, D, E, F (critical)
- These ratings are *subjective*
  - How do clinicians disagree? (variance? bias?)
- More details of this work in Rogers et al 2013 and Rogers et al 2010

Data

- $c = 1 \ldots C$ clinicians.
- $p = 1 \ldots P$ patients.
- For patient p, we have T_p observations at times $\mathbf{t}_p = [t_{p1}, \ldots, t_{pT_p}]^T$.
- $y^p_{\tau c}$ is rating at $t_{p\tau}$ ($\{A, B, C, D, E\}$).

The model

- Assumptions:
  - We assume that there is an unobserved continuous latent health function for each patient
  - Each clinician observes this and and corrupts it in two ways:
    - Adds Gaussian noise
    - Adds a constant offset (optimist, pessimist)
  - Corrupted health is then binned to give category

The model

- Assumptions:
  - We assume that there is an unobserved continuous latent health function for each patient
  - Each clinician observes this and and corrupts it in two ways:
    - Adds Gaussian noise
    - Adds a constant offset (optimist, pessimist)
  - Corrupted health is then binned to give category
- The model:
  - Patient health is modelled as a GP.
  - Corrupted health is the auxiliary variable (q) – one set of auxiliary variables per clinician
  - q is binned to produce rating.
  - Note that $p(q|f)$ has been generalised from standard normal.

Model

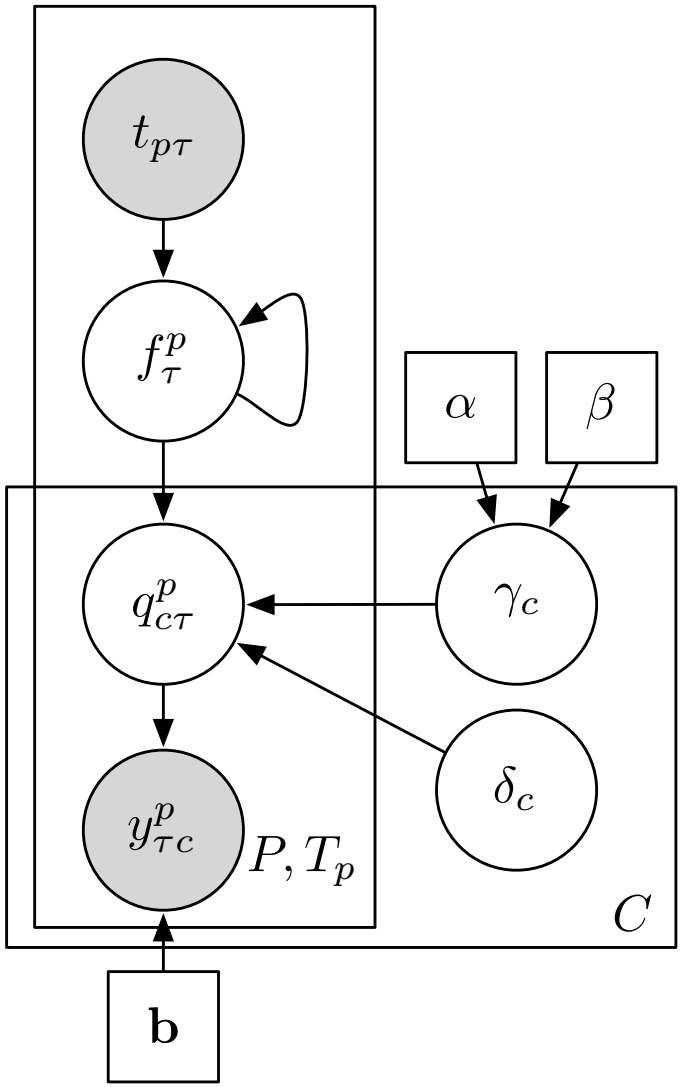


Figure 18 : Plates diagram

$$
\begin{aligned}
\mathbf{f}^p &\sim \mathcal{N}(\mathbf{0}, \mathbf{C}^p) \quad \text{[health]} \\
\delta_c &\sim \mathcal{N}(0, 1) \quad \text{[offset]} \\
\gamma_c &\sim \mathcal{G}(\alpha, \beta) \quad \text{[precision]} \\
q^p_{c\tau} &\sim \mathcal{N}(f^p_\tau + \delta_c, \gamma_c^{-1}) \\
P(y^p_{\tau c} = k) &= \delta(b_k < q^p_{c\tau} < b_{k+1})
\end{aligned}
$$

Previously auxiliary variables were $z_n \sim \mathcal{N}(f_n, 1)$. This model adds clinician-specific offsets and precisions: $q^p_{c\tau} \sim \mathcal{N}(f^p_\tau + \delta_c, \gamma_c^{-1})$.

Example data generation

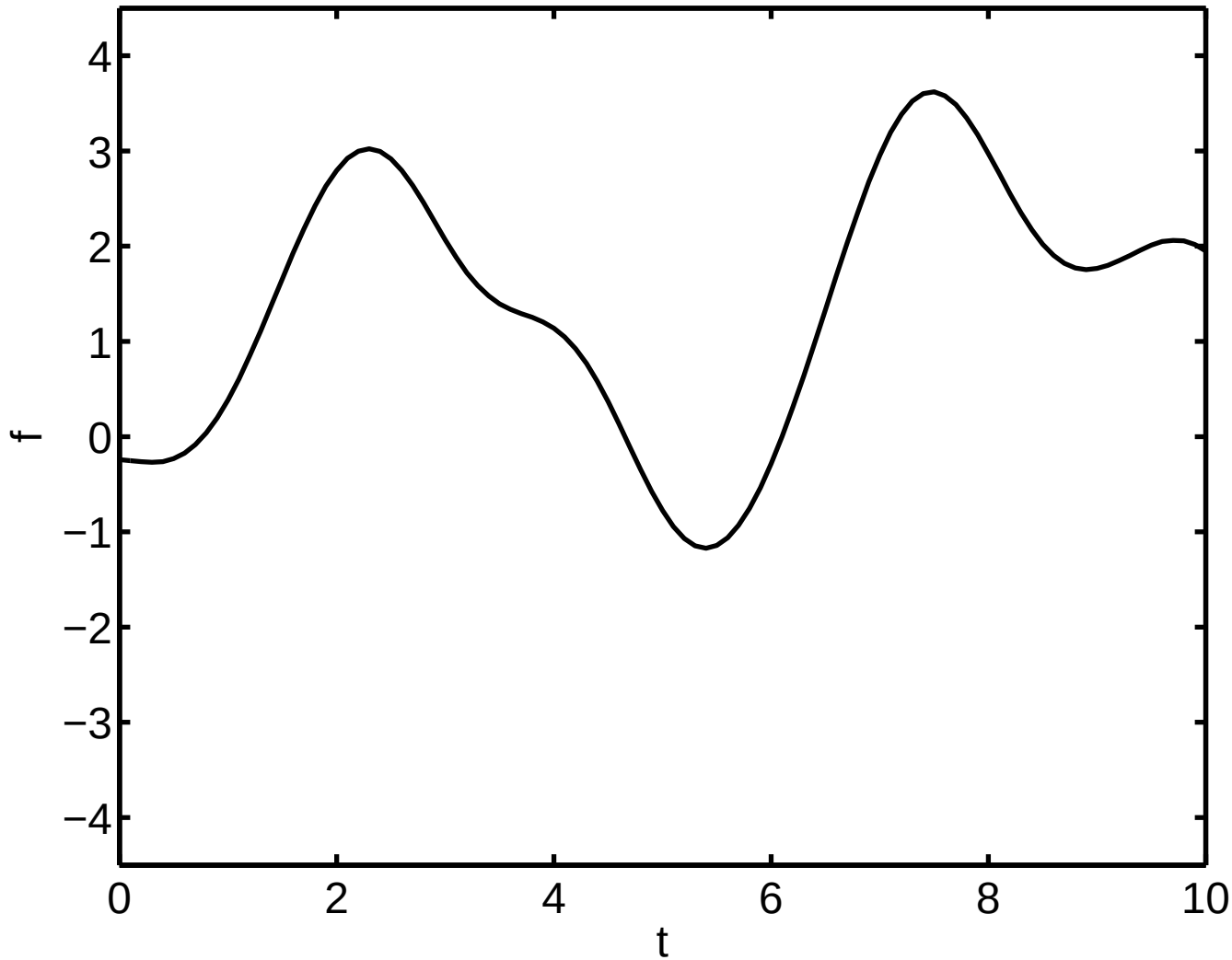


Figure 19 : Example of the generative process described by the model for three clinicans.

Example data generation

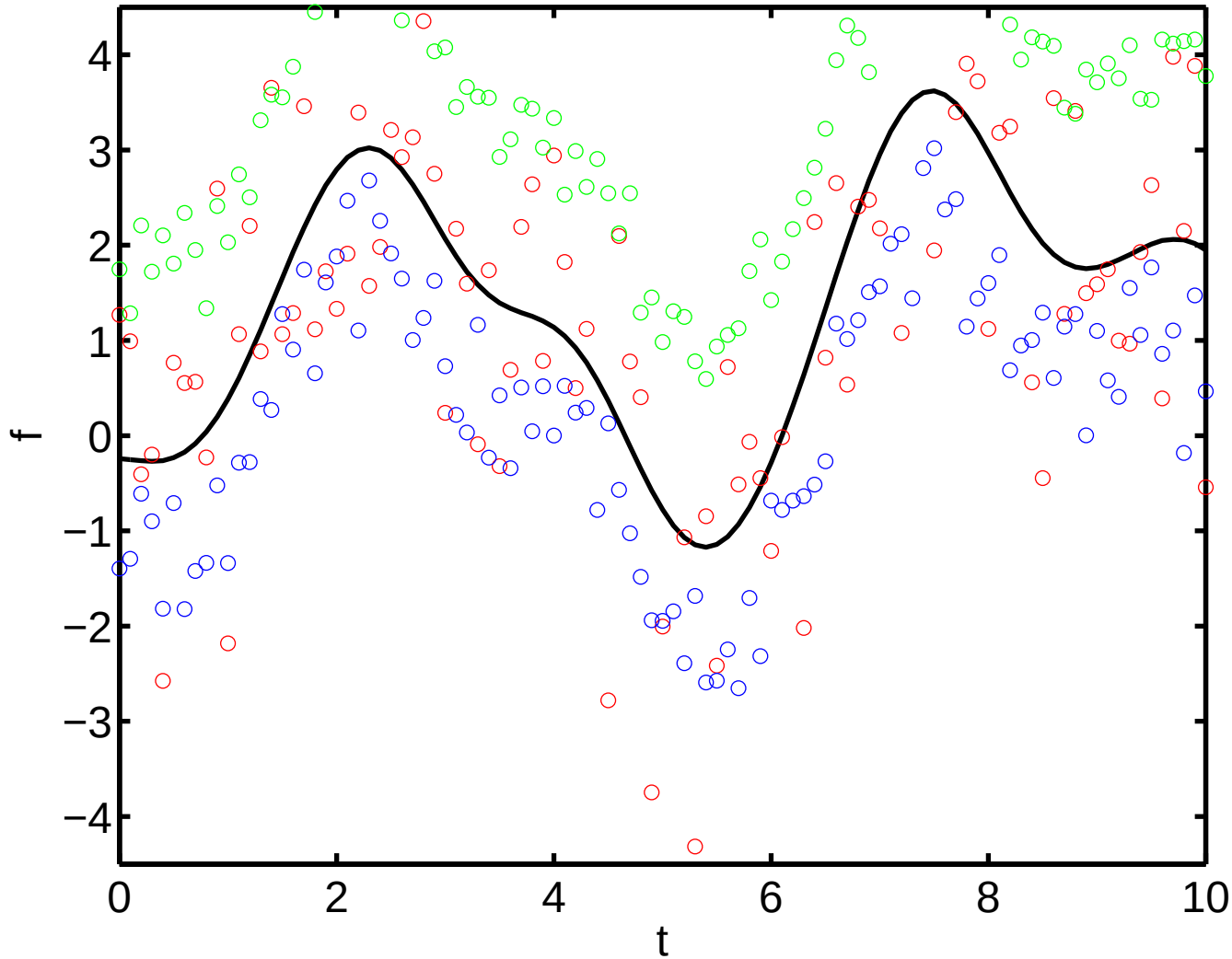


Figure 19 : Example of the generative process described by the model for three clinicans.

Example data generation

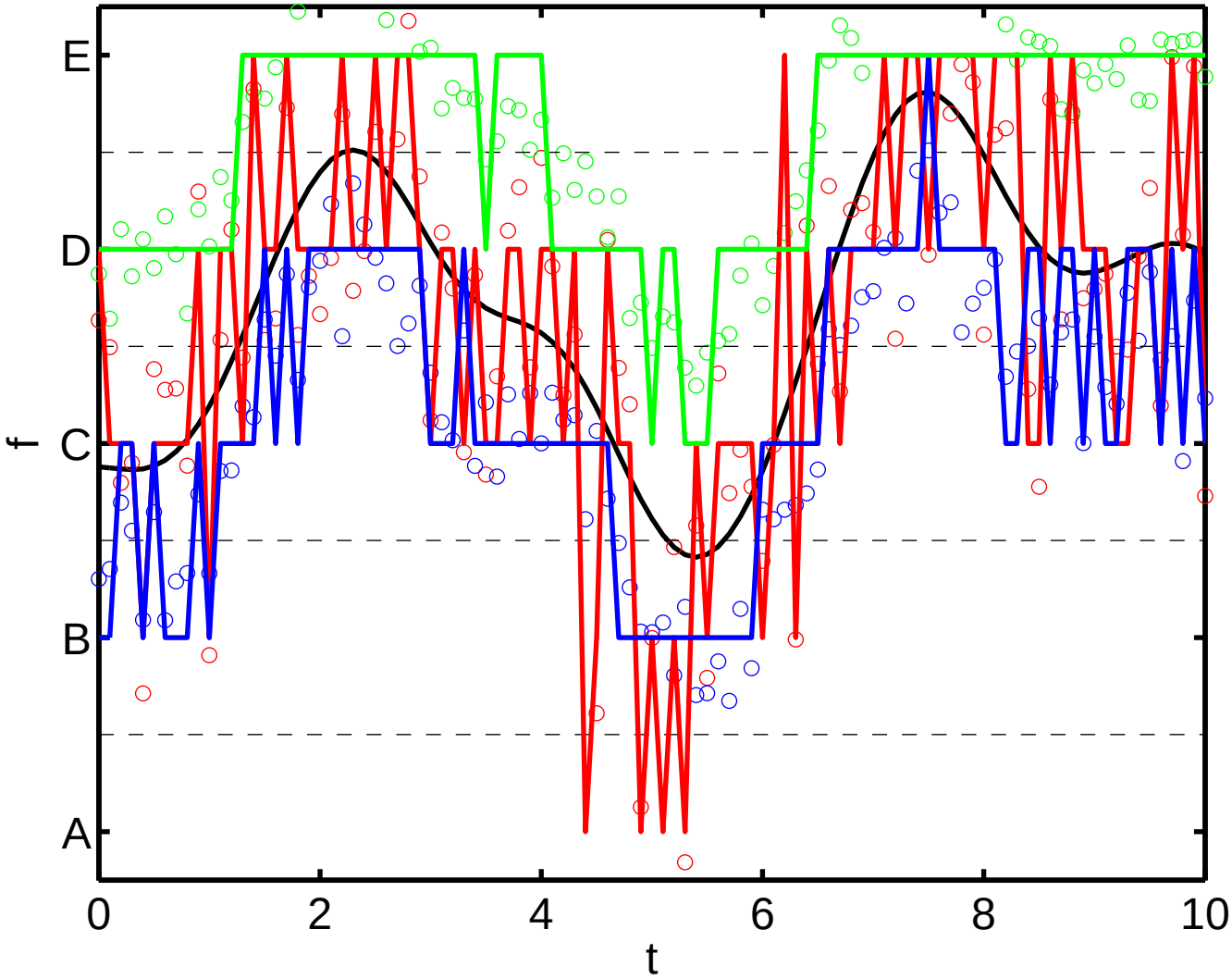


Figure 19 : Example of the generative process described by the model for three clinicans.

Model inference

- Gibbs sampling is straightforward
  - The latent health function for each patient.

$$p(\mathbf{f}^p|\ldots) = \mathcal{N}(\mathbf{f}^p|\boldsymbol{\mu}_{\mathbf{f}^p}, \boldsymbol{\Sigma}_{\mathbf{f}^p})$$

where:

$$\boldsymbol{\Sigma}_{\mathbf{f}^p} = \left((\mathbf{C}^p)^{-1} + \sum_c \gamma_c \mathbf{I}\right)^{-1}, \quad \boldsymbol{\mu}_{\mathbf{f}^p} = \boldsymbol{\Sigma}_{\mathbf{f}^p} \sum_c \gamma_c (\mathbf{q}^p_{c\cdot} - \delta_c)$$

  - The auxiliary variables:

$$p(q^p_{c\tau}|y^p_{c\tau} = k, \ldots) \propto \delta(b_k < q^p_{c\tau} < b_{k+1}) \mathcal{N}(q^p_{c\tau}|f^p_\tau + \delta_c, \gamma_c^{-1})$$

- Gibbs sampling continued:
  - The offset and precision for each clinician:

$$p(\delta_c|\ldots) = \mathcal{N}(\delta_c|\mu_c, \sigma_c^2), \quad p(\gamma_c|\ldots) = \mathcal{G}(a_c, b_c)$$

where:

$$\sigma_c^2 = (1 + N_c)^{-1}, \quad \mu_c = \gamma_c \sigma_c^2 \sum_p \sum_\tau (q_{c\tau}^p - f_\tau^p)$$

and:

$$a_c = \alpha + N_c/2, \quad b_c = \beta + \frac{1}{2} \sum_p \sum_\tau (q_{c\tau}^p - f_\tau^p)^2$$

and N_c is the total number of observations for clinician c.

Inference and Interpretation

- The offset and precision tell us how the clinicians disagree.
  - Low precision indicates a clinician who is very unpredictable (w.r.t the rest)
  - High offset indicates a precision who consistently rates higher or lower than the norm.

Inference and Interpretation

- The offset and precision tell us how the clinicians disagree.
  - Low precision indicates a clinician who is very unpredictable (w.r.t the rest)
  - High offset indicates a precision who consistently rates higher or lower than the norm.
- Identifiability: offset for one clinician fixed to 0.
- Covariance parameter for GP inferred via Metropolis-Hastings (see e.g. Rasmussen 2000)

Results

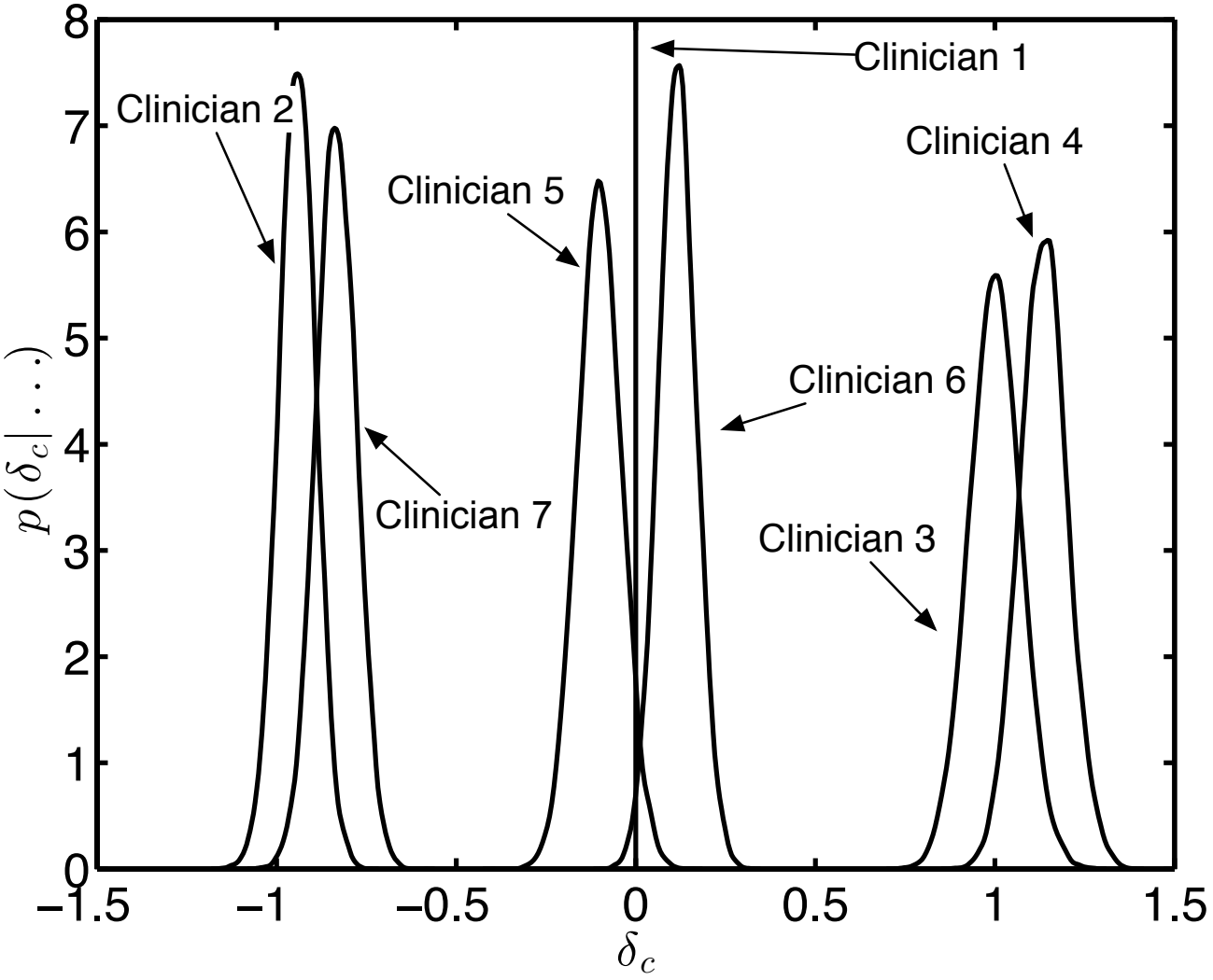
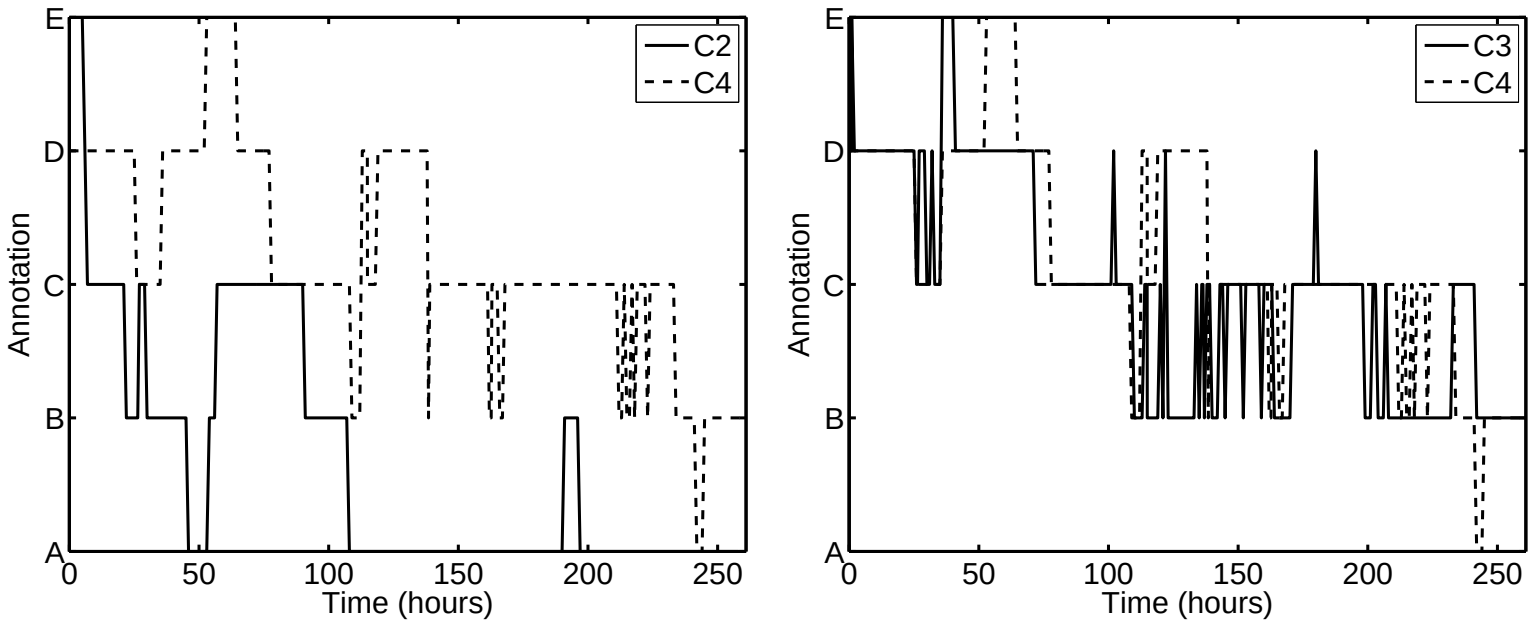


Figure 20 : Marginal offset posteriors

Results



(a) Clinicians 2 and 4

(b) Clinicians 3 and 4

Figure 21 : Inferred offsets make sense on inspection of the data.

Results

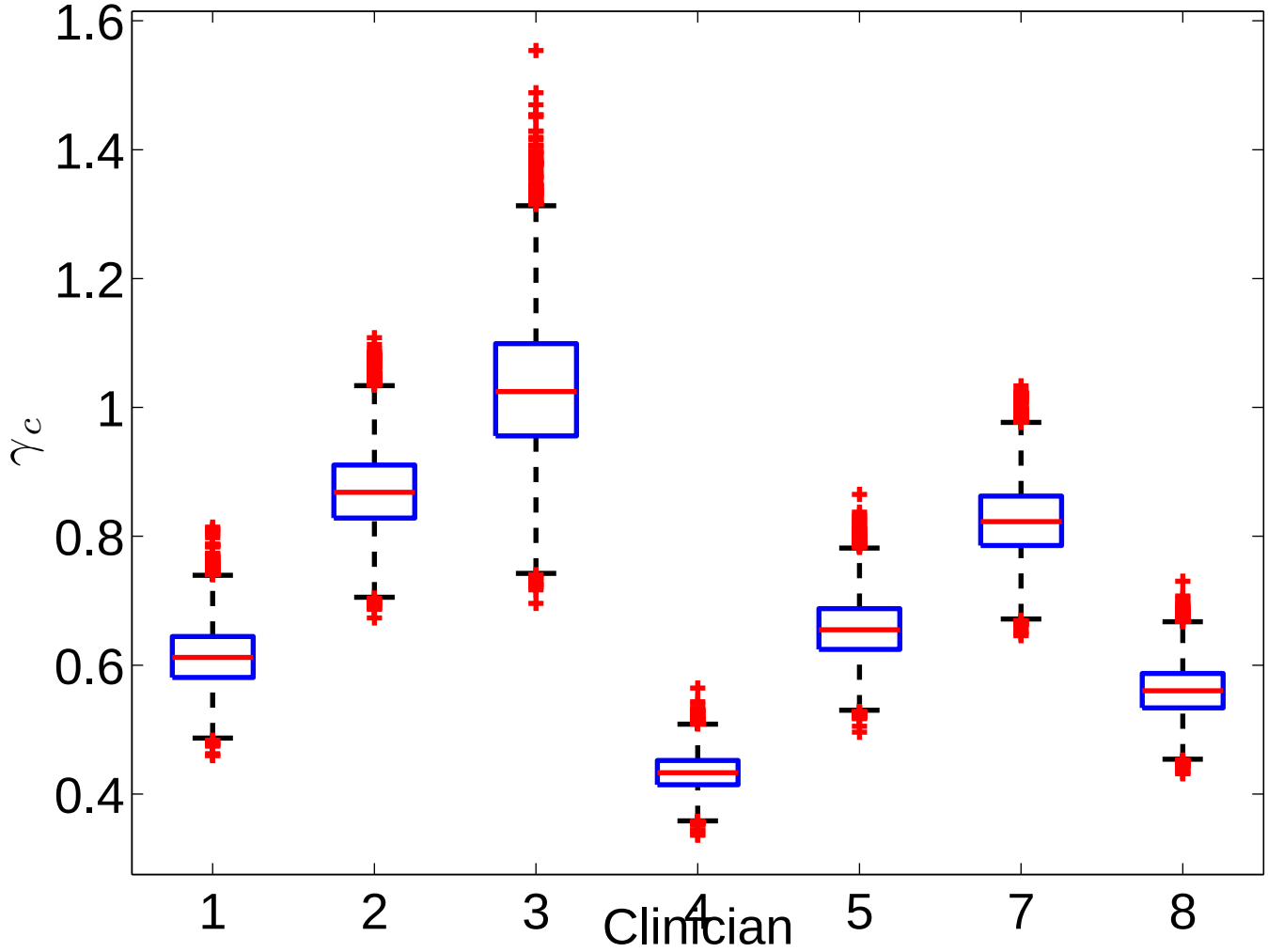
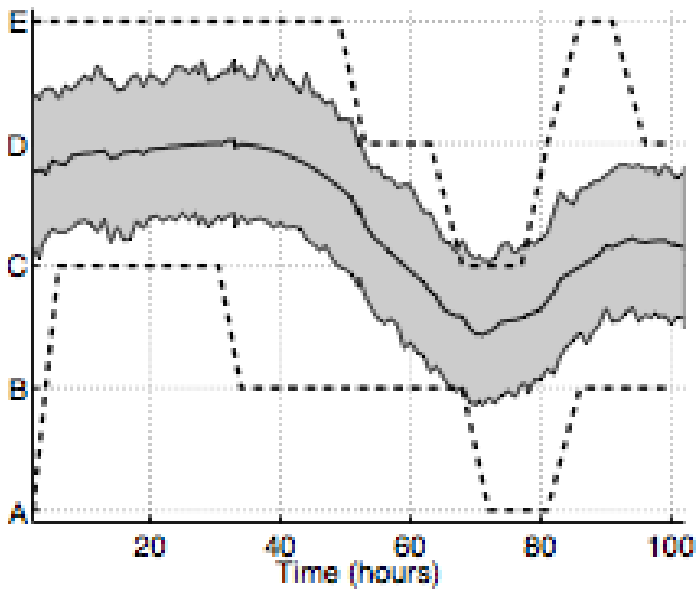


Figure 22 : Marginal precision posteriors. Clinicians 1, 4, and 8 appear to be the least consistent (wrt the majority)

Results

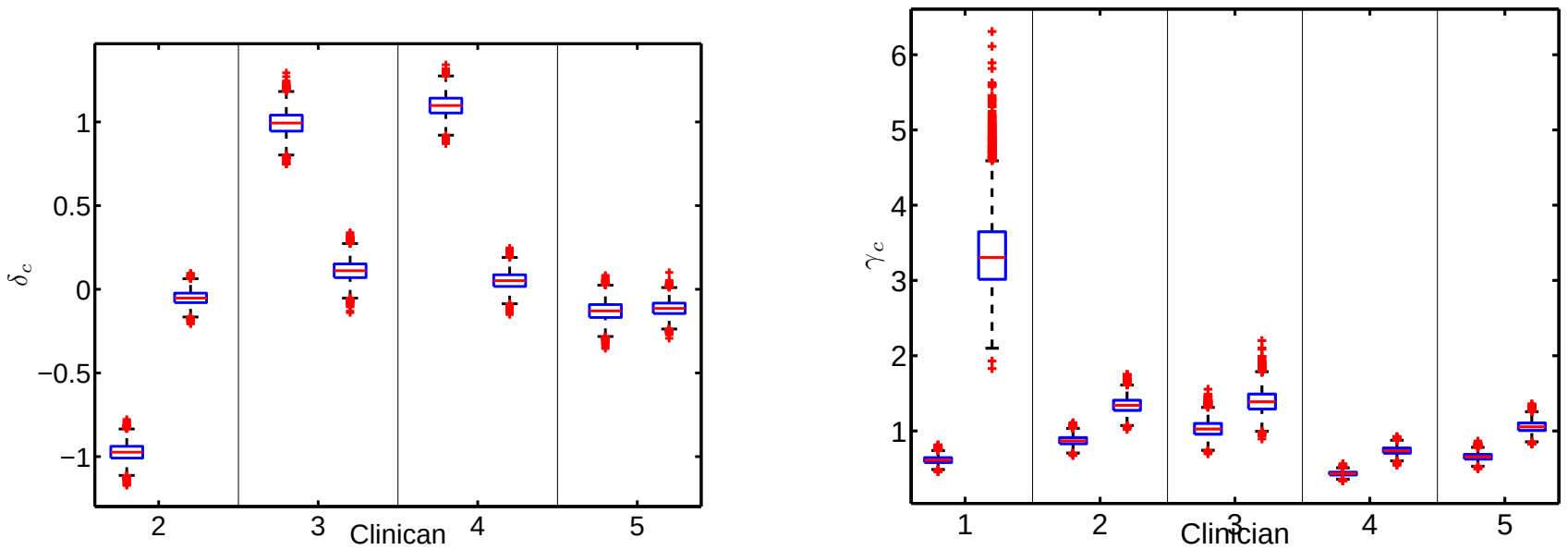


- Posterior health function for one patient
- Shaded area shows boundaries of posterior samples
- Dashed lines show maximum and minimum clinician ratings
  - Note the range: initially patient rated both A and E!

INSIGHT

- After the initial annotation, clinicians went through INSIGHT procedure.
- The goal was to make ratings more consistent.
- If it succeeded, we should see a reduction in offset and increase in precision in the post-INSIGHT data.
- Note: only 5 clinicians remained after INSIGHT

Post-INSIGHT results

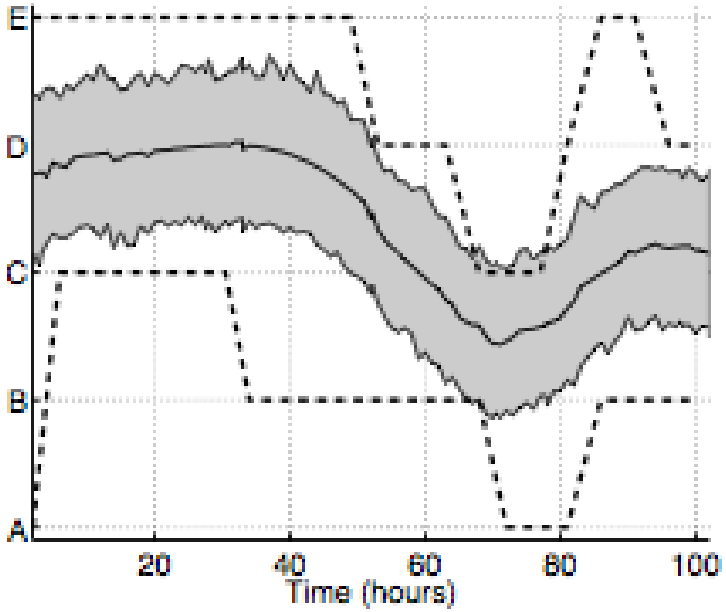


(a) Offsets

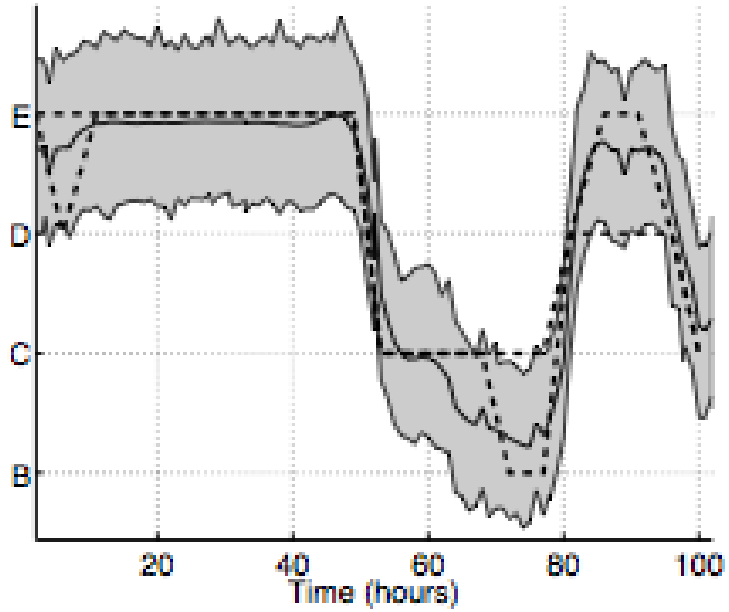
(b) Precisions

Figure 23 : Offsets and precision before and after INSIGHT. Offsets get closer to 0, whilst precision increase suggesting greater agreement amongsth clinicians.

Post-INSIGHT results



(a) Before INSIGHT

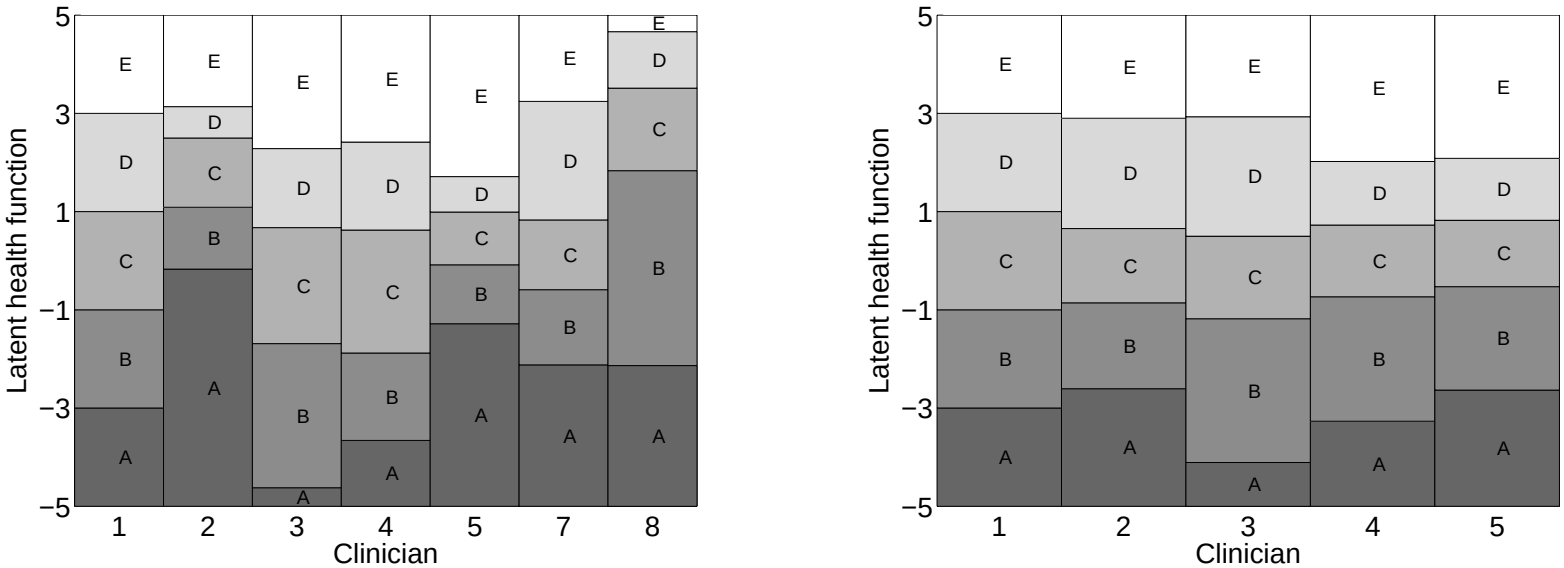


(b) After INSIGHT

- Patient health function before and after INSIGHT
- Less smooth after INSIGHT
- Range of ratings much reduced

Inferring category boundaries

- So far, it has been assumed that all categories are the same size (i.e. the elements of $\mathbf{b}$ are equally spaced).
- We can also infer these (with fixed end-points and $\delta_c = 0$).
- Removes uniform prior assumption over categories.



(a) Before INSIGHT

(b) After INSIGHT

Figure 24 : Posterior mean cateogory boundaries.

Summary and Conclusions

- Model allows us to:
  - learn something about *how* clinicians disagree and how they rate.
  - assess the effectiveness of the INSIGHT procedure.

Summary and Conclusions

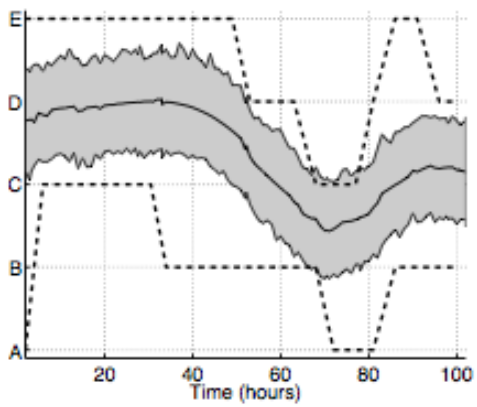
- Model allows us to:
  - learn something about *how* clinicians disagree and how they rate.
  - assess the effectiveness of the INSIGHT procedure.
- GP prior:
  - Flexible
  - Required no parametric assumptions about health function
  - Hyper-parameter (γ) was inferred in the model (could be patient-specific)

Summary and Conclusions

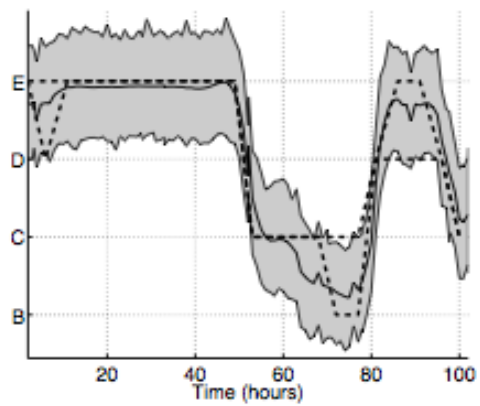
- Model allows us to:
  - learn something about *how* clinicians disagree and how they rate.
  - assess the effectiveness of the INSIGHT procedure.
- GP prior:
  - Flexible
  - Required no parametric assumptions about health function
  - Hyper-parameter (γ) was inferred in the model (could be patient-specific)
- Auxiliary Variable Trick:
  - Not restricted to a standard Gaussian centered on the GP variable.
  - Incorporated offset and precision without causing additional inference challenges.

Future work

- Incorporate measured covariates:
  - HR, BP, etc
  - Predictive model? (would need better ground truth)
- Patient-specific covariance parameters
- Non-stationary covariance parameters
  - Long periods of no change followed by short periods of fast change



(a) Before INSIGHT



(b) After INSIGHT