CMPUT 466
Introduction to Gaussian Processes

Dan Lizotte

The Plan

- Introduction to Gaussian Processes
- Fancier Gaussian Processes
  - The current DFF. (*de facto* fanciness)
- Uses for:
  - Regression
  - Classification
  - Optimization
- Discussion

Why GPs?

- Here are some data points! What function did they come from?
  - I have *no idea*.
- Oh. Okay. Uh, you think this point is likely in the function too?
  - I have *no idea*.

Why GPs?

- Here are some data points, and here's how I rank the likelihood of functions.
  - Here's where the function will most likely be
  - Here are some examples of what it might look like
  - Here is the likelihood of your hypothesis function
  - Here is a prediction of what you'll see if you evaluate your function at x', with confidence

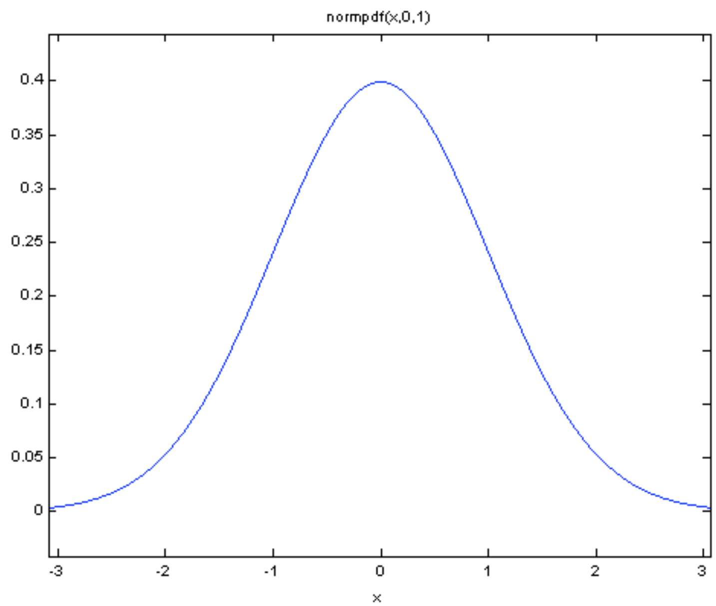
Why GPs?

- You can’t get anywhere without making some assumptions
- GPs are a nice way of expressing this ‘prior on functions’ idea.
- Like a more ‘complete’ view of least-squares regression
- Can do a bunch of cool stuff
  - Regression
  - Classification
  - Optimization

Gaussian

- Unimodal
- Concentrated
- Easy to compute with
    - Sometimes
- Tons of crazy properties

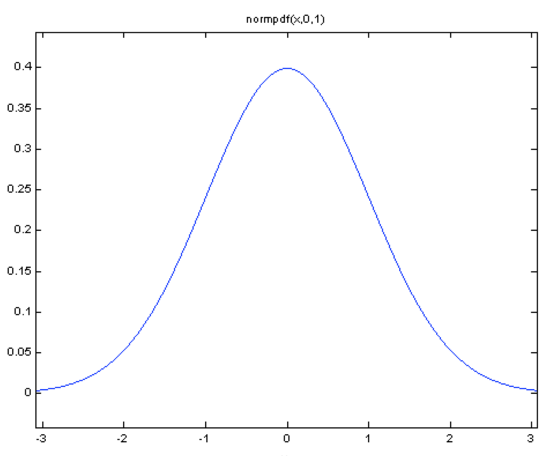
$$\frac{e^{\frac{-(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}$$

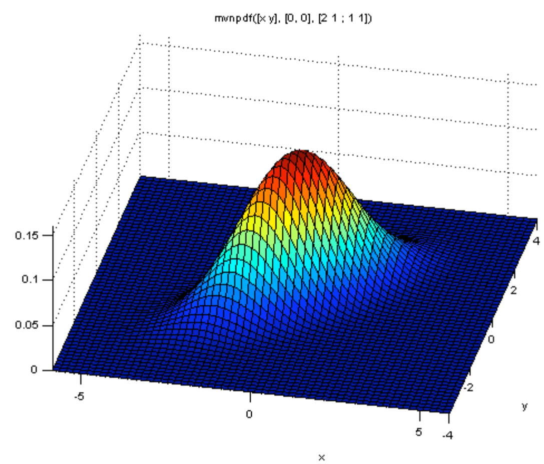


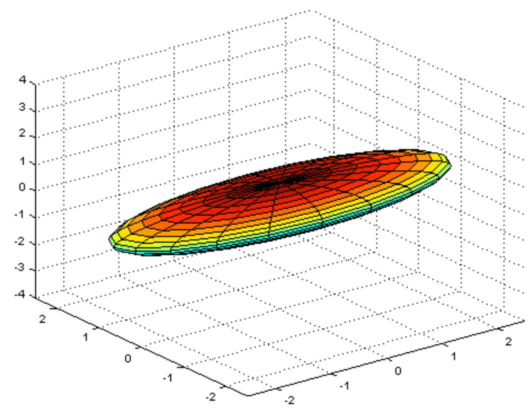
Multivariate Gaussian

- Same thing, but more so
- Some things are harder
  - No nice form for cdf
- 'Classical' view: Points in $\mathbb{R}^d$

$$\frac{e^{-\frac{1}{2}(\mathbf{x}-\mu)^{\mathbf{T}}\Sigma^{-1}(\mathbf{x}-\mu)}}{\sqrt{(2\pi)^n\,|\Sigma|}}$$

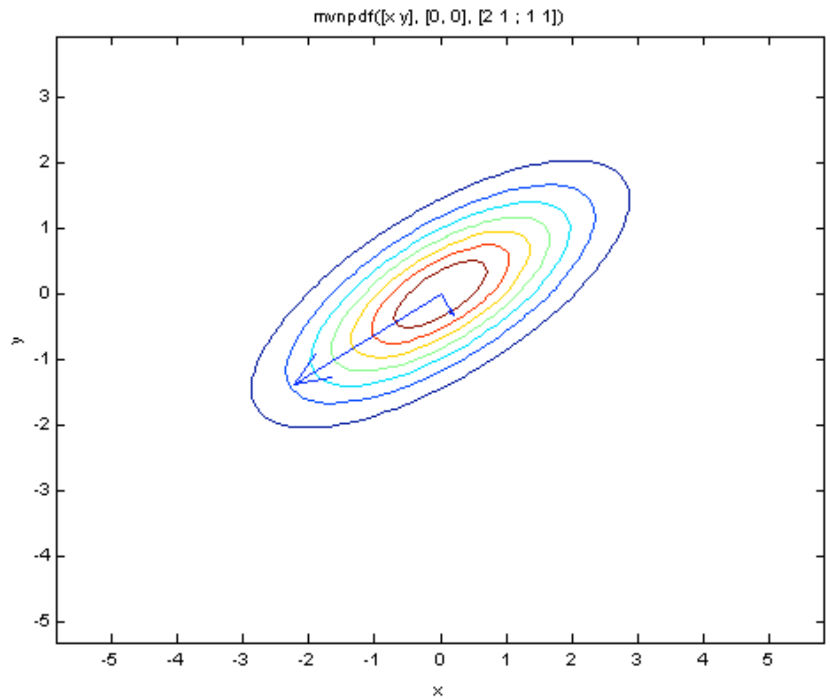






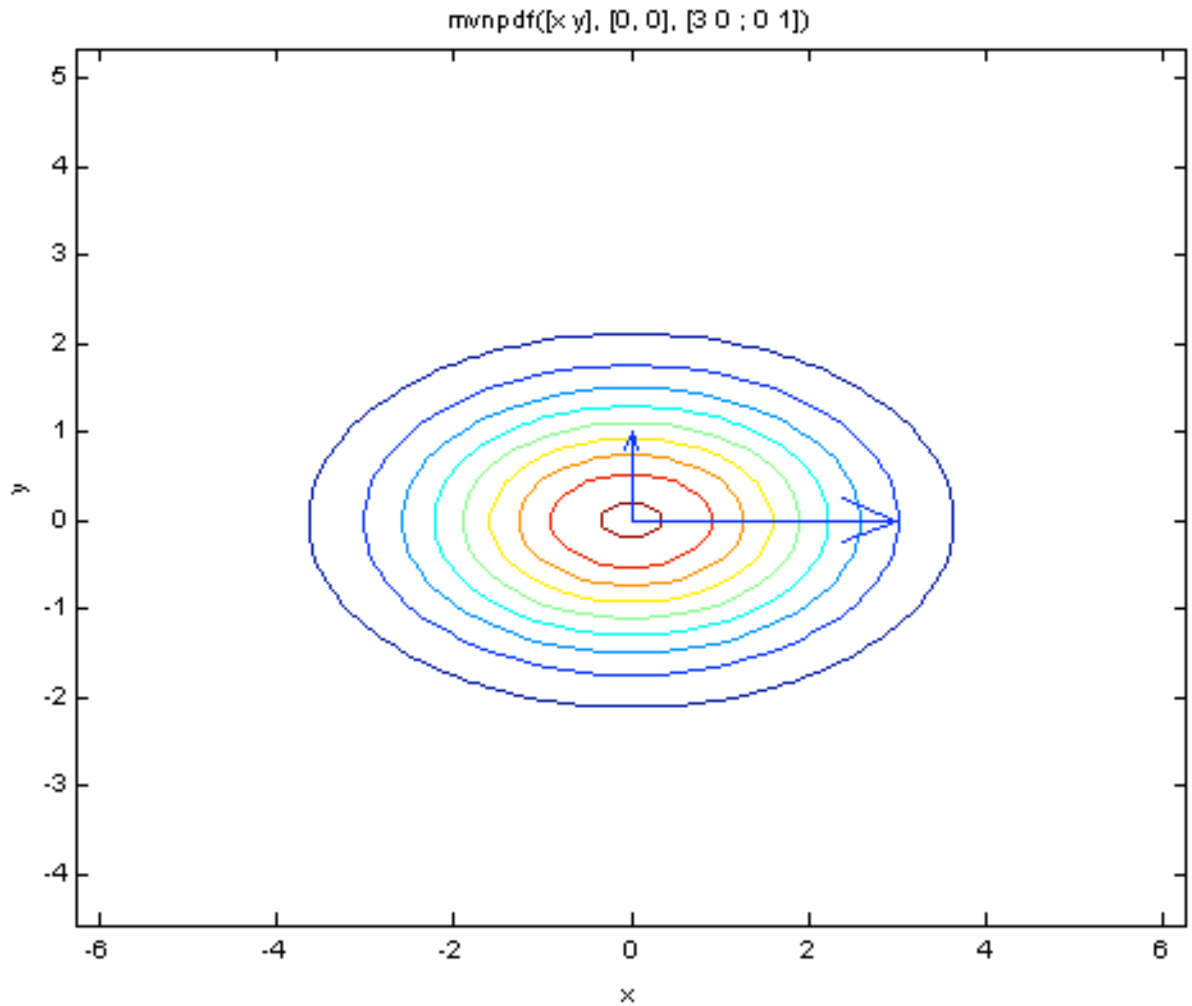
Covariance Matrix

- Shape param
- Eigenstuff indicates variance and correlations



$$\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.53 & -0.85 \\ -0.85 & -0.53 \end{bmatrix} \begin{bmatrix} 0.38 & 0 \\ 0 & 2.62 \end{bmatrix} \begin{bmatrix} 0.53 & -0.85 \\ -0.85 & -0.53 \end{bmatrix}$$

$$P(y \mid x) \neq P(y)$$



$$P(y \mid x) = P(y)$$

David's Demo #1

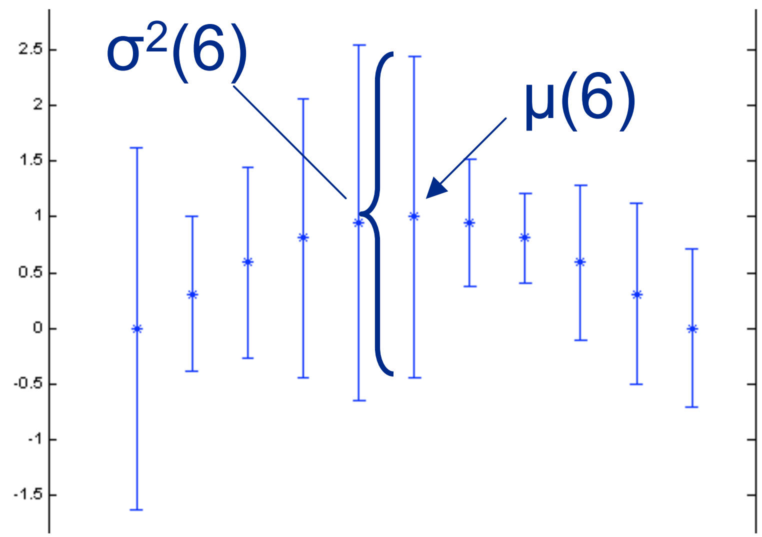
- Yay for David MacKay!

- Professor of Natural Philosophy, and Gatsby Senior Research Fellow
- Department of Physics
- Cavendish Laboratory, University of Cambridge

- http://www.inference.phy.cam.ac.uk/mackay/

Higher Dimensions

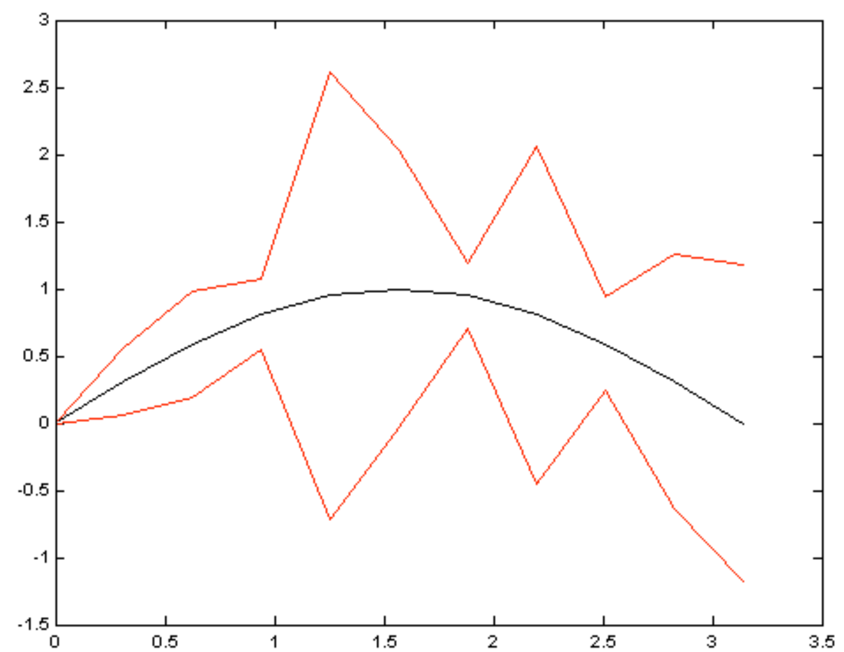
- Visualizing > 3 dimensions is…difficult
- Thinking about vectors in the '*i,j,k*' engineering sense is a trap
- Means and marginals is practical
  - But then we don't see correlations
- Marginal distributions are Gaussian
- ex., F|6 ~ N(μ(6), σ²(6))



David’s Demos #2,3

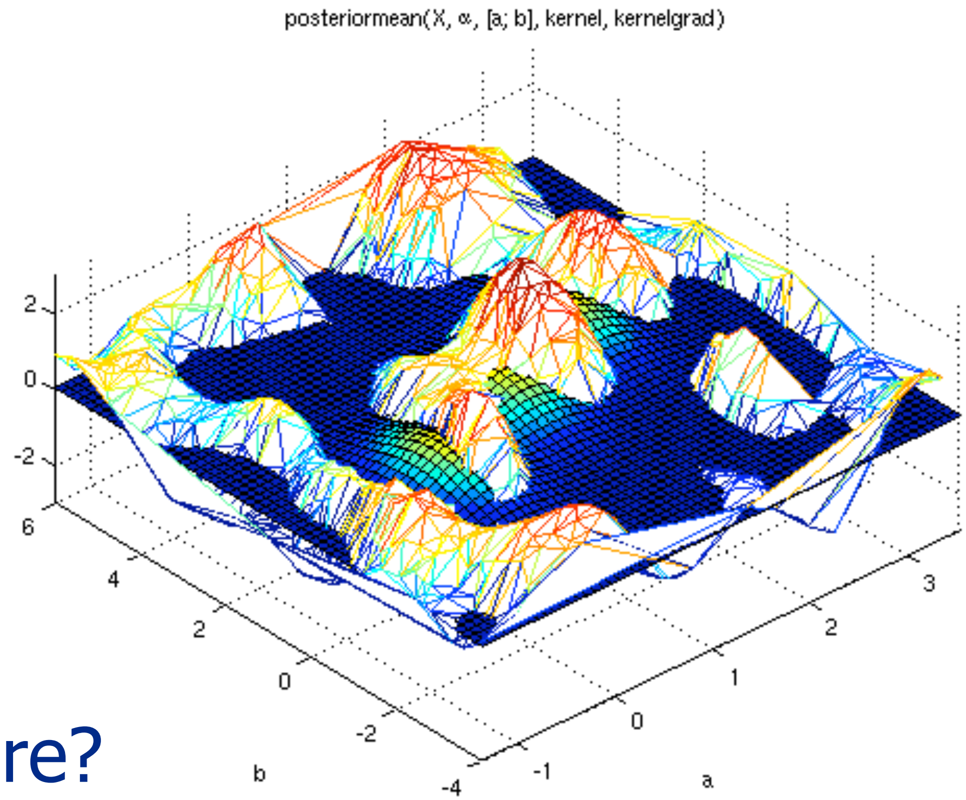
Yet Higher Dimensions

- Why stop there?
- We indexed before with $\mathbb{Z}$. Why not $\mathbb{R}$?
- Need functions $\mu(x)$, $k(x,z)$ for all $x, z \in \mathbb{R}$
- x and z are *indices*
- F is now an uncountably infinite dimensional vector
- Don't panic: It's just a function



David’s Demo #5

Getting Ridiculous



- Why stop there?
- We indexed before with $\mathbb{R}$. Why not $\mathbb{R}^d$?
- Need functions $\mu(x)$, $k(x,z)$ for all $x, z \in \mathbb{R}^d$

David’s Demo #11 (Part 1)

Gaussian Process

- Probability distribution *indexed by* an arbitrary set
- Each element gets a Gaussian distribution over the reals with mean $\mu(x)$
- These distributions are dependent/correlated as defined by $k(x,z)$
- Any finite subset of indices defines a multivariate Gaussian distribution
  - Crazy mathematical statistics and measure theory ensures this

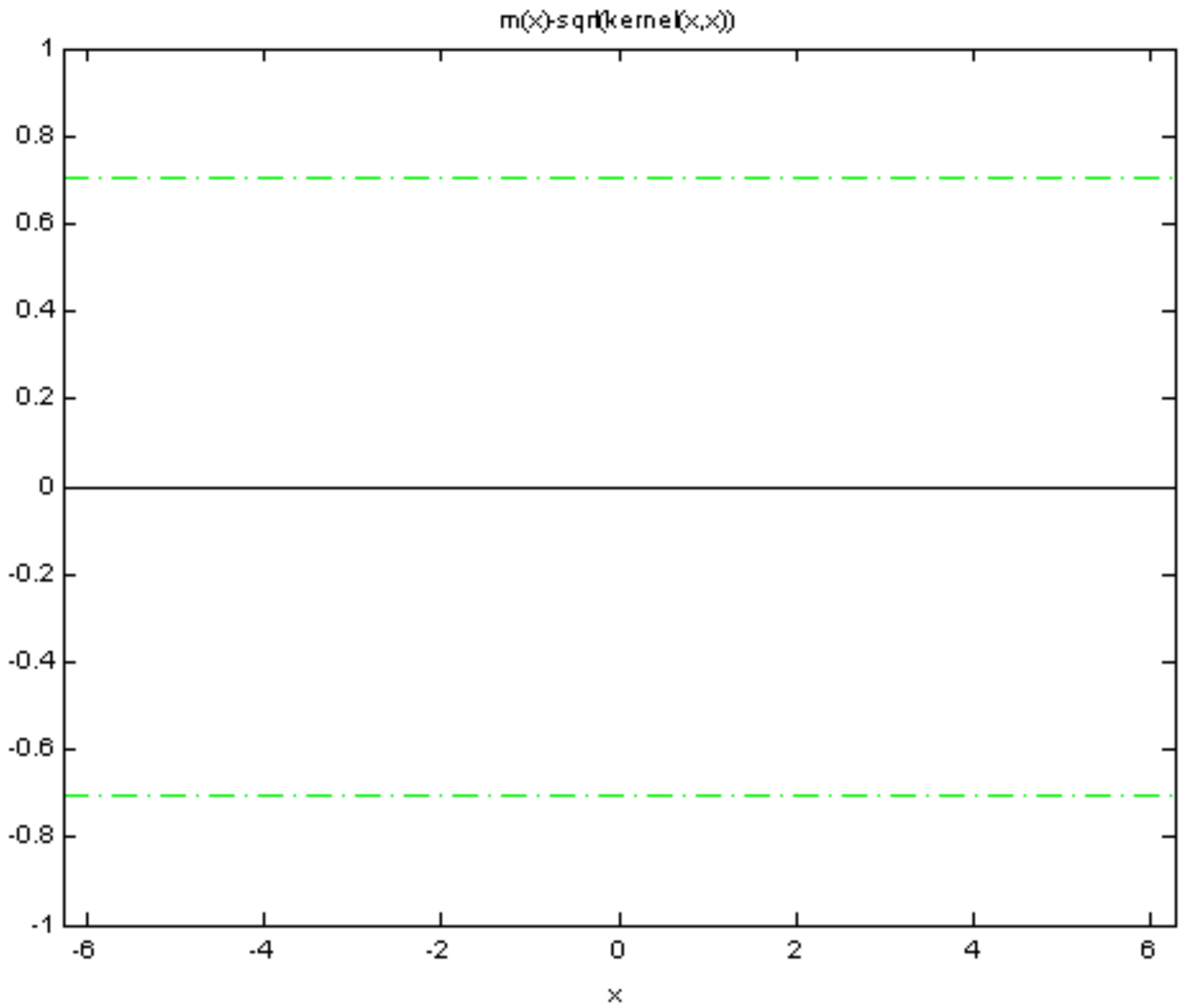
Gaussian Process

- Distribution over *functions*
- Index set can be pretty much whatever
  - Reals
  - Real vectors
  - Graphs
  - Strings
  - …
- Most interesting structure is in k(x,z), the 'kernel.'

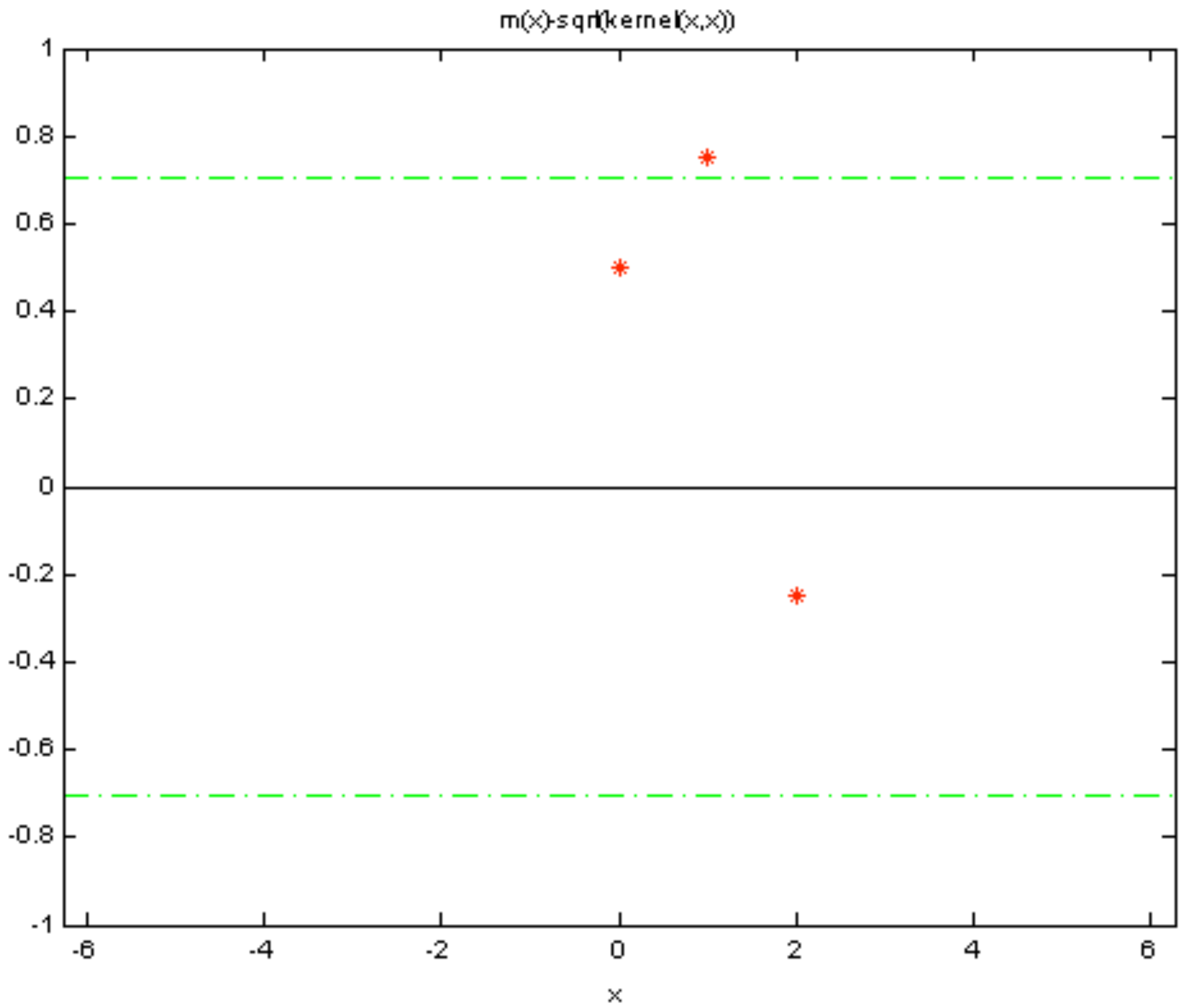
Bayesian Updates for GPs

- How do Bayesians use a Gaussian Process?
  - Start with GP prior
  - Get some data
  - Compute a posterior
- Ask interesting questions about the posterior

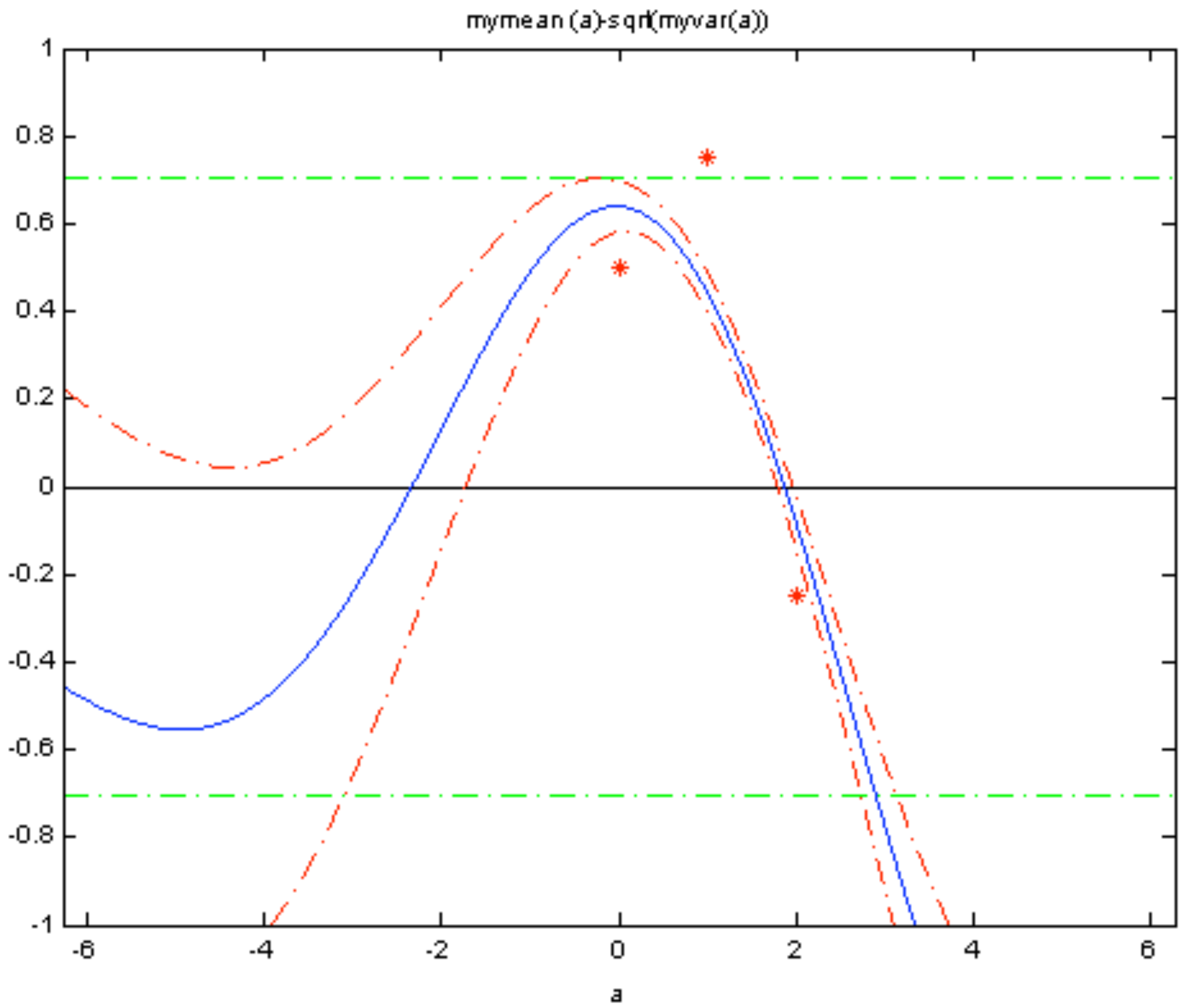
Prior



Data



Posterior



Computing the Posterior

- Given
  - Prior, and list of observed data points F|x
    - indexed by a list x_1, x_2, …, x_j
  - A query point F|x'

$$F|x'|\boldsymbol{F}|\boldsymbol{x} \sim \mathcal{N}(\hat{\mu}(x'), \hat{\sigma}^2(x'))$$

where

$$\hat{\mu}(y) = \boxed{\mu(x')} + \boxed{\boldsymbol{k}(\boldsymbol{x}, x')}^T \mathsf{K}(\boldsymbol{x}, \boldsymbol{x})^{-1}(\boldsymbol{f}|\boldsymbol{x} - \boldsymbol{\mu}(\boldsymbol{x}))$$

$$\hat{\sigma}^2(y) = \boxed{k(x', x')} - \boxed{\boldsymbol{k}(\boldsymbol{x}, x')}^T \mathsf{K}(\boldsymbol{x}, \boldsymbol{x})^{-1}\boxed{\boldsymbol{k}(\boldsymbol{x}, x')}$$

so $\hat{\mu}(x')$ is linear in $\boldsymbol{k}(\boldsymbol{x}, x')$,
and $\hat{\sigma}^2(x')$ is quadratic in $\boldsymbol{k}(\boldsymbol{x}, x')$

Computing the Posterior

- Given
  - Prior, and list of observed data points F|x
    - indexed by a list x_1, x_2, …, x_j
  - A query point F|x'

$$F|x'|\boldsymbol{F}|\boldsymbol{x} \sim \mathcal{N}(\hat{\mu}(x'), \hat{\sigma}^2(x'))$$

where

$$\hat{\mu}(y) = \boxed{\mu(x')} + \boxed{\boldsymbol{k}(\boldsymbol{x}, x')}^T \boxed{\boldsymbol{\alpha}}$$

$$\hat{\sigma}^2(y) = \boxed{k(x', x')} - \boxed{\boldsymbol{k}(\boldsymbol{x}, x')}^T \boxed{\mathbf{A}} \boxed{\boldsymbol{k}(\boldsymbol{x}, x')}$$

so $\hat{\mu}(x')$ is linear in $\boldsymbol{k}(\boldsymbol{x}, x')$,
and $\hat{\sigma}^2(x')$ is quadratic in $\boldsymbol{k}(\boldsymbol{x}, x')$

Computing the Posterior

- Posterior mean function is sum of kernels
  - Like basis functions
- Posterior variance is quadratic form of kernels

$$F|x'|\boldsymbol{F}|\boldsymbol{x} \sim \mathcal{N}(\hat{\mu}(x'), \hat{\sigma}^2(x'))$$

where

$$\hat{\mu}(y) = \boxed{\mu(x')} + \boxed{\boldsymbol{k}(\boldsymbol{x}, x')}^T \boxed{\boldsymbol{\alpha}}$$

$$\hat{\sigma}^2(y) = \boxed{k(x', x')} - \boxed{\boldsymbol{k}(\boldsymbol{x}, x')}^T \boxed{\mathbf{A}} \boxed{\boldsymbol{k}(\boldsymbol{x}, x')}$$

so $\hat{\mu}(x')$ is linear in $\boldsymbol{k}(\boldsymbol{x}, x')$,
and $\hat{\sigma}^2(x')$ is quadratic in $\boldsymbol{k}(\boldsymbol{x}, x')$

Parade of Kernels

Regression

- We've already been doing this, really
- The posterior mean is our 'fitted curve'
  - We saw linear kernels do linear regression
- But we also get error bars

Hyperparameters

- Take the SE kernel for example

$$k(\mathbf{x},\mathbf{x}') = \boxed{\sigma^2} \cdot \mathrm{e}^{-\frac{(\mathbf{x}-\mathbf{x}')^{\mathsf{T}}\boxed{\mathsf{L}}(\mathbf{x}-\mathbf{x}')}{2}} + \delta_{xx'}\boxed{\sigma_\epsilon^2}$$

- Typically, $\mathsf{L} = \mathrm{diag}(\ell_1^{-2}, \ell_2^{-2} \ldots \ell_N^{-2})$
- σ^2 is the process variance
- $\sigma^2_\in$ is the noise variance

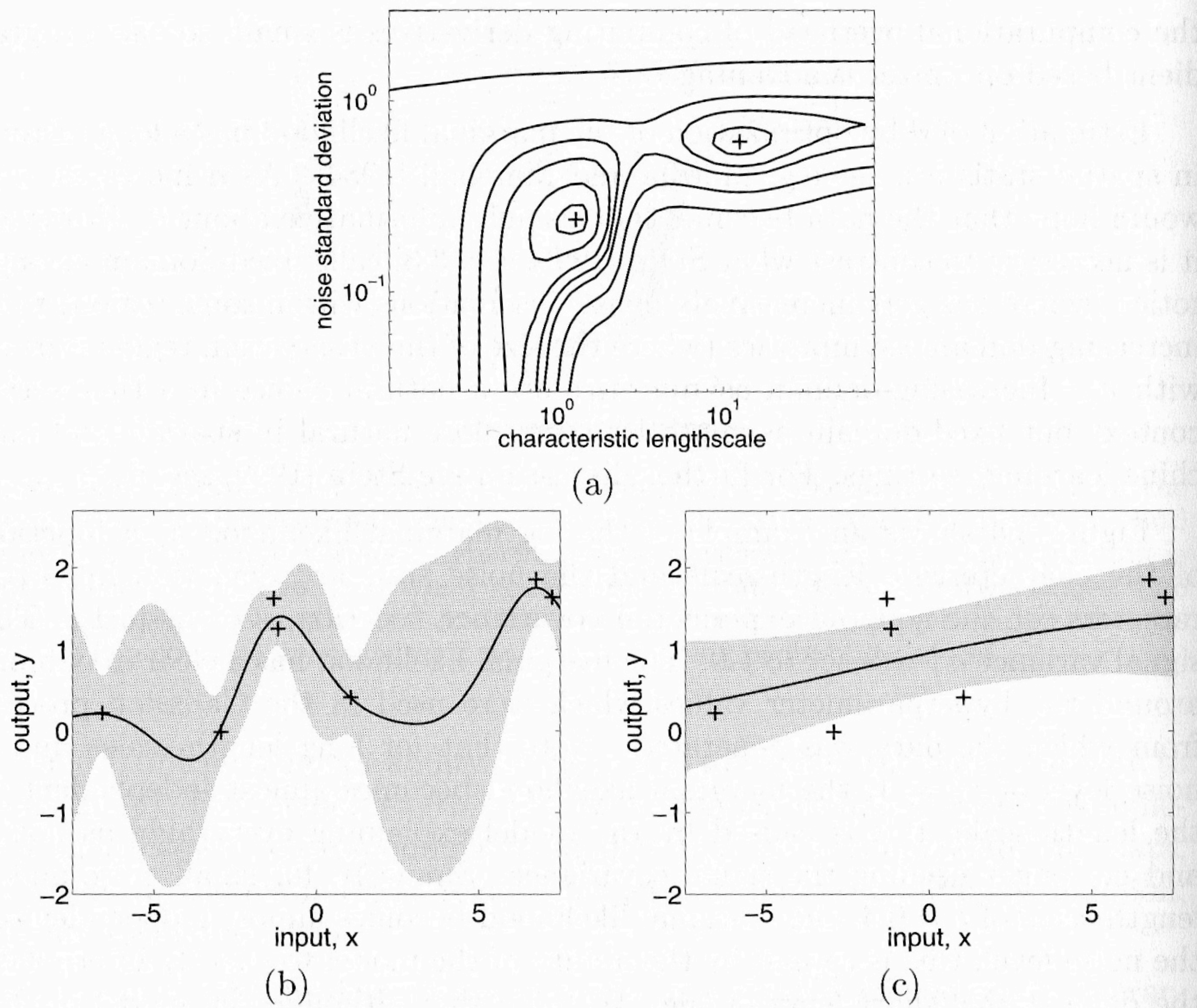
Model Selection

- How do we pick these?
  - What do you mean pick them? Aren't you Bayesian? Don't you have a prior over them?
  - If you're really Bayesian, skip this section and do MCMC instead.
- Otherwise, use Maximum Likelihood, or Cross Validation. (But don't use cross validation.)

$$\log P(\mathbf{y}|X,\theta) = -\frac{1}{2}\mathrm{y}^\top K^{-1}\mathrm{y} - \frac{1}{2}\log|K_y| - \frac{N}{2}\log 2\pi$$

- Terms for data fit, complexity penalty
- It's differentiable if k(x,x') is; just hill climb

David’s Demo #6, 7, 8, 9, 11


noise standard deviation
10^0
10^{-1}
10^0
10^1
characteristic lengthscale
(a)
output, y
2
1
0
−1
−2
−5
0
5
input, x
(b)
output, y
2
1
0
−1
−2
−5
0
5
input, x
(c)

De Facto Fanciness

- *At least* learn your length scale(s), mean, and noise variance from data
- Automatic Relevance Detection using the Squared Exponential kernel seems to be the current default
- Matérn Polynomials becoming more used; these are less smooth

Classification

$$P(c = 1|X = x) = \frac{1}{1+\mathrm{e}^{-GP(x)}}$$

- That's it. Just like Logistic Regression.
- The GP is the *latent function* we use to describe the distribution of c|x
- We squash the GP to get probabilities

David’s Demo #12

Classification

- We’re not Gaussian anymore
- Need methods like Laplace Approximation, or Expectation Propagation, or…
- Why do this?
  - “Like an SVM” (kernel trick available) but probabilistic. (I know; no margin, etc. etc.)
  - Provides confidence intervals on predictions

Optimization

- Given $f: X \to \mathbb{R}$, find $\min_{x \in X} f(x)$
- Everybody's doing it
- Can be easy or hard, depending on
  - Continuous vs. Discrete domain
  - Convex vs. Non-convex
  - Analytic vs. Black-box
  - Deterministic vs. Stochastic

What's the Difference?

- Classical Function Optimization
  - Oh, I have this function f(x)
  - Gradient is ∇f…
  - Hessian is *H*…
- Bayesian Function Optimization
  - Oh, I have this random variable F|x
  - I think its distribution is…
  - Oh well, now that I've seen a sample I think the distribution is…

Common Assumptions

- $F|x = f(x) + \varepsilon|x$
- What they don't tell you:
  - $f(x)$ 'arbitrary' deterministic function
  - $\varepsilon|x$ is a r.v., $E(\varepsilon) = 0$, (i.e. $E(F|x) = f(x)$)
- Really only makes sense if $\varepsilon|x$ is unimodal
  - Any given sample is probably close to f
- But maybe not Gaussian

What's the Plan?

- Get samples of $F|x = f(x) + \varepsilon|x$
- Estimate and minimize $m(x)$
  - Regression + Optimization
- i.e., reduce to deterministic global minimization

Bayesian Optimization

- Views optimization as a decision process
- At which x should we sample F|x next, given what we know so far?
- Uses model and objective
- What model?
  - I wonder… Can anybody think of a probabilistic model for functions?

Bayesian Optimization

- We constantly have a model F_{post} of our function F
  - Use a GP over m, and assume $\varepsilon \sim N(0,s)$
- As we accumulate data, the model improves
- How should we accumulate data?
- Use the posterior model to select which point to sample next

The Rational Thing

- Minimize $\int_F (f(x') - f(x^*))\, dP(f)$
- One-step
  - Choose x′ to maximize ‘expected improvement’
- *b*-step
  - Consider all possible length *b* trajectories, with the last step as described above
- As if.

The Common Thing

- Cheat!
- Choose x' to maximize 'expected improvement by at least c'
- $c = 0 \Rightarrow$ max posterior mean
- $c = \infty \Rightarrow$ max posterior var
- "How do I pick c?"
- "Beats me."
  - Maybe my thesis will answer this! Exciting.

The Problem with Greediness

- For which point x does F(x) have the lowest posterior mean?
- This is, in general, a non-convex, global optimization problem.
- WHAT??!!
  - I know, but remember F is expensive
  - Also remember quantities are linear/quadratic in **k**
- Problems
  - *Trajectory* trapped in local minima
    - (below prior mean)
  - Does not acknowledge model uncertainty

An Alternative

- Why not select
  - $x' = \text{argmax } P((F|x' \leq F|x) \; \forall \; x \in X)$
  - i.e., sample F(x) next where x is most likely to be the minimum of the function
- Because it's hard
  - Or at least I can't do it. Domain is too big.

An Alternative

- Instead, choose
  - $x' = \text{argmin } P((F|x' \leq c)\ \forall\ x \in X)$
- What about c?
  - Set it to the best value seen so far
  - Worked for us
- It would be really nice to relate c (or ε) to the number of samples remaining

AIBO Walking

- Set up a Gaussian process over R^{15}
- Kernel is Squared Exponential (careful!)
- Parameters for priors found by maximum likelihood
  - We could be more Bayesian here and use priors over the model parameters
- Walk, get velocity, pick new parameters, walk

Stereo Matching

- What?
- Daniel Neilson has been using GPs to optimize his stereo matching code.
- It's been working surprisingly well; we're going to augment the model soon.(-ish.)
- Ask him!

That's It

- No it's not. I didn't cover:
  - RL! Yaki and Mohammad are currently working on this. Right guys?
  - A reasonable amount on classification. Sorry; not my thing.
  - Anything not in R^N. We can do strings, trees, graphs…
  - Approximation methods for large datasets
  - Deeper kernel analysis (eigenfunctions…)
  - Other processes…

That's It

- But too bad. That's it.
- Who has questions?

This is a good book by Carl Rasmussen and Chris Williams. Also it's only $35 on Amazon.ca