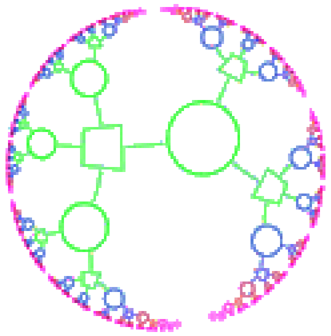
Gaussian Process Basics

David MacKay

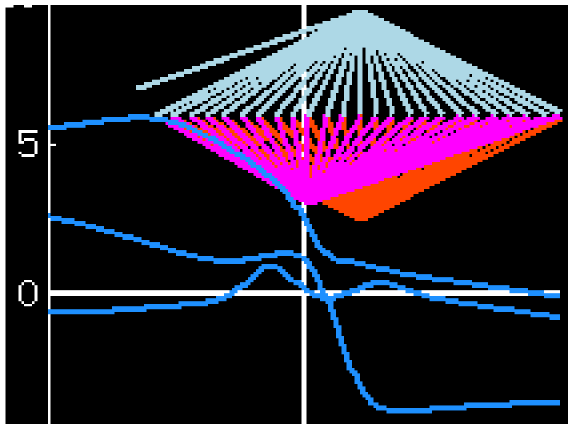


Department of Physics, University of Cambridge

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

f9 then f11

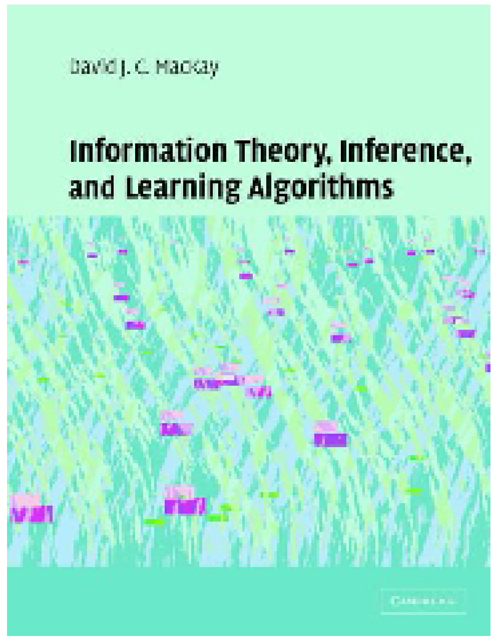
Nonlinear regression with neural networks



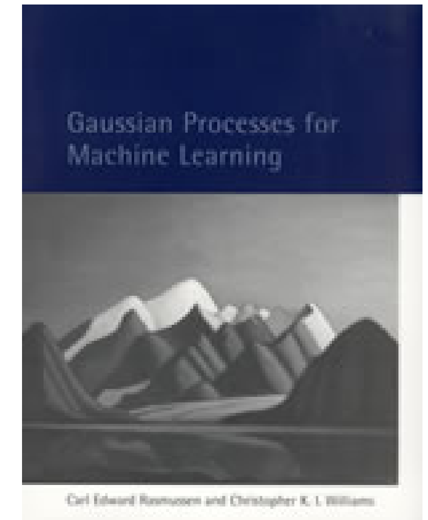
- Multi-layer perceptron with regularization (weight decay)
  - Bayesian complexity control
  - Automatic relevance determination

References

www.inference.phy.cam.ac.uk/mackay/



Chapter 45

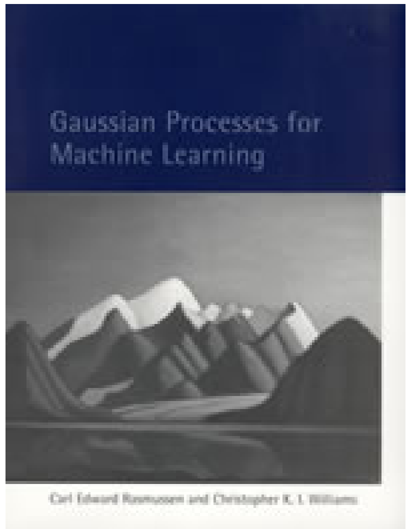


Rasmussen and Williams

Bletchley Park, 265

Notation

		Rasmussen and Williams notation, if different
Model	$\mathbf{x} \to y = f(\mathbf{x}) + \epsilon$	
Data	$\{\mathbf{x}_n, y_n\}_{n=1}^{N}$	$i = 1 \dots n$ replaced by $n = 1 \dots N$
Noise level	σ_ν^2	σ_n^2
Dimension of input space	D	
Horizontal lengthscale	l	
Vertical lengthscale	σ_f	
Covariance function	$k(\mathbf{x}_n, \mathbf{x}_{n'}) \equiv \text{cov}\left(f(\mathbf{x}_n), f(\mathbf{x}_{n'})\right)$	
'squared, exponential'	$= \sigma_f^2 \exp\left(-\frac{1}{2l^2}\left(\mathbf{x}_n - \mathbf{x}_{n'}\right)^2\right)$	
	$\text{cov}\left(y_n, y_{n'}\right) = k(\mathbf{x}_n, \mathbf{x}_{n'}) + \sigma_\nu^2 \delta_{nn'}$	n, n' input point labels rather than p, q
Parametric model		
Parameterized function	$f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w}$	
Dimension of parameter space	$h = 1 \dots H$	N replaced by H



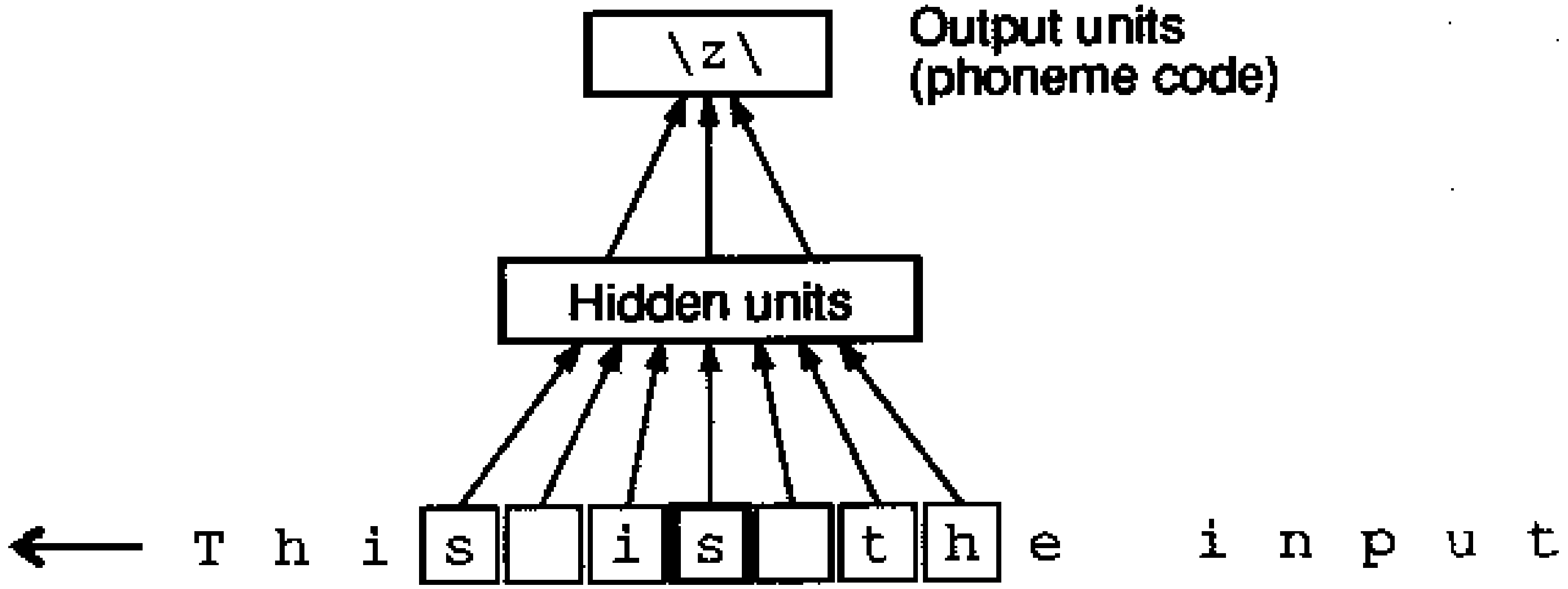
Motivation: Machine learning

- Reading aloud - Nettalk
  - Sejnowski and Rosenberg
- Handwriting recognition - LeNet
  - Yann LeCun
- Weld toughness
  - Bhadeshia et al
- Focussing multiple-mirror telescopes
  - Roger Angel

Data: $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$

Motivation: Machine learning

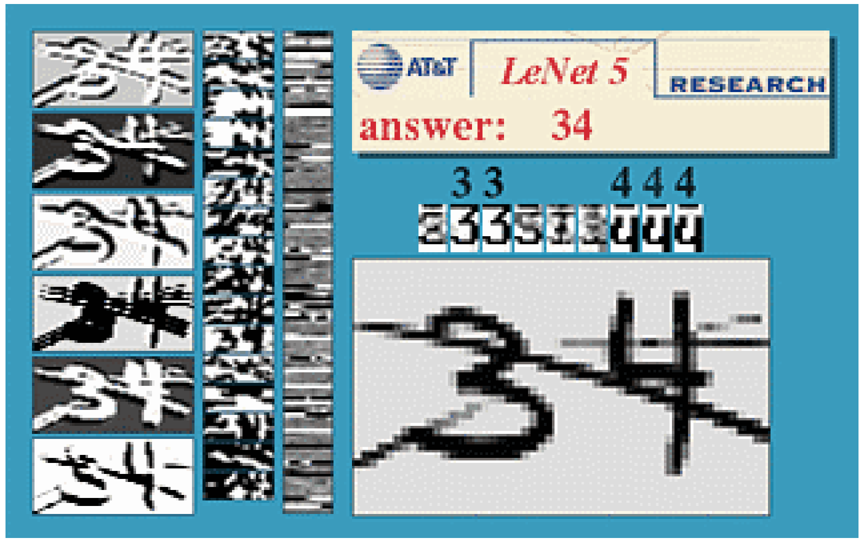
- Reading aloud - Nettalk
  - Sejnowski and Rosenberg



- Handwriting recognition - LeNet
  - Yann LeCun
- Weld toughness
  - Bhadeshia et al
- Focussing multiple-mirror telescopes
  - Roger Angel

Motivation: Machine learning

- Reading aloud - Nettalk
  - Sejnowski and Rosenberg
- Handwriting recognition - LeNet
  - Yann LeCun



- Weld toughness
  - Bhadeshia et al
- Focussing multiple-mirror telescopes
  - Roger Angel

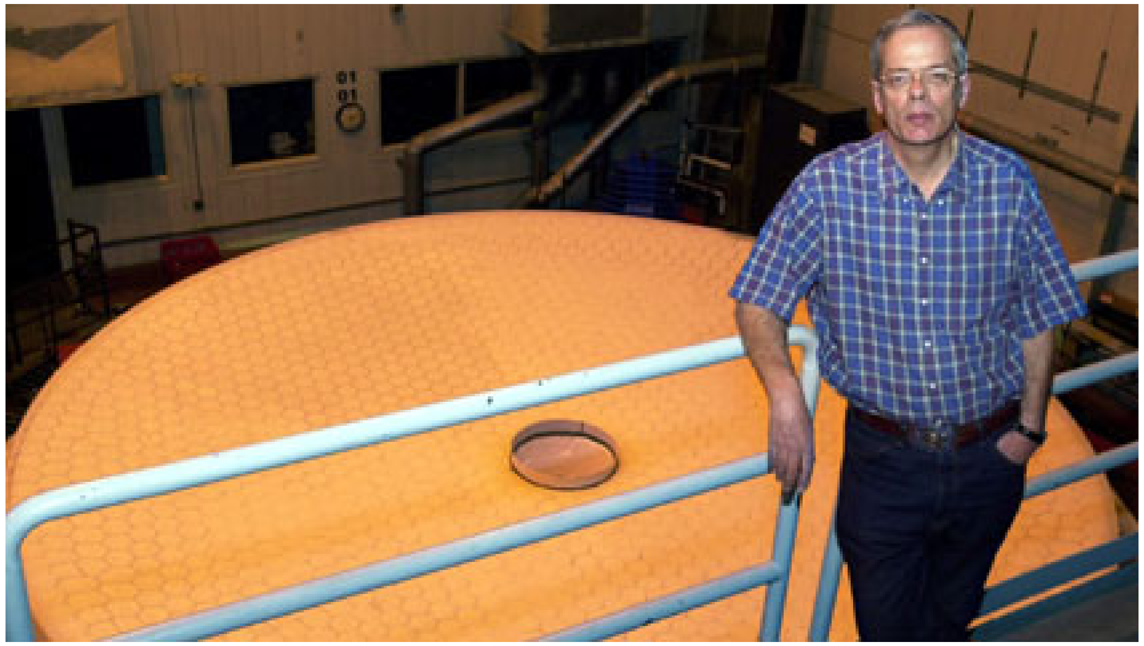
Motivation: Machine learning

- Reading aloud - Nettalk
  - Sejnowski and Rosenberg
- Handwriting recognition
  - Yann LeCun
- Weld toughness
  - Bhadeshia et al

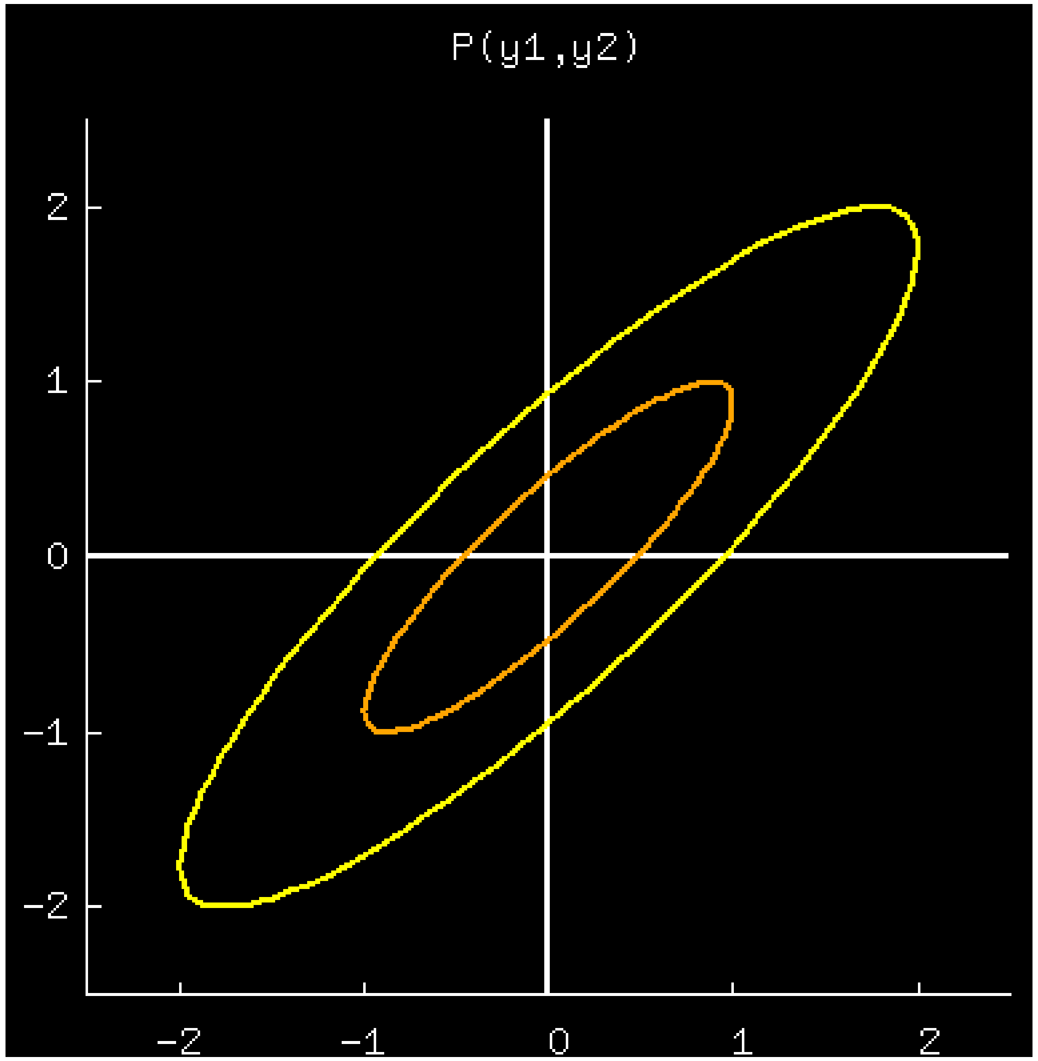
- Focussing telescopes
  - Roger Angel

Motivation: Machine learning

- Reading aloud - Nettalk
  - Sejnowski and Rosenberg
- Handwriting recognition - LeNet
  - Yann LeCun
- Weld toughness
  - Bhadeshia et al
- Focussing multiple-mirror telescopes
  - Roger Angel



Can all this be done by a plain old Gaussian distribution?



$$P(\mathbf{y} \,|\, \mathbf{K}) = \frac{1}{[\det 2\pi\mathbf{K}]^{1/2}} e^{-\frac{1}{2}\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}$$

$\mathbf{K}$ is the covariance matrix

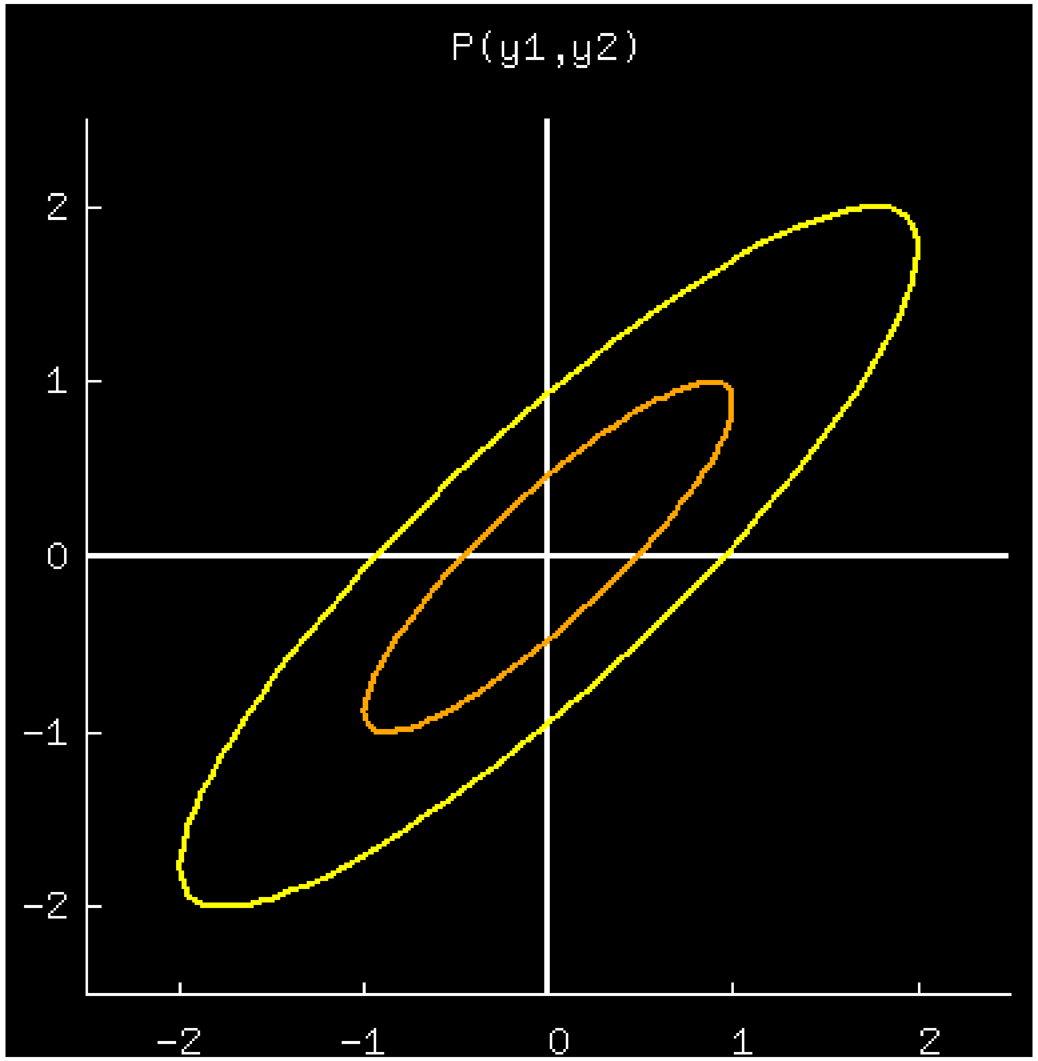
Two-dimensional Gaussian

- Covariance matrix

$$\mathbf{K} = \begin{bmatrix} 1.00010 & 0.88250 \\ 0.88250 & 1.00010 \end{bmatrix}$$

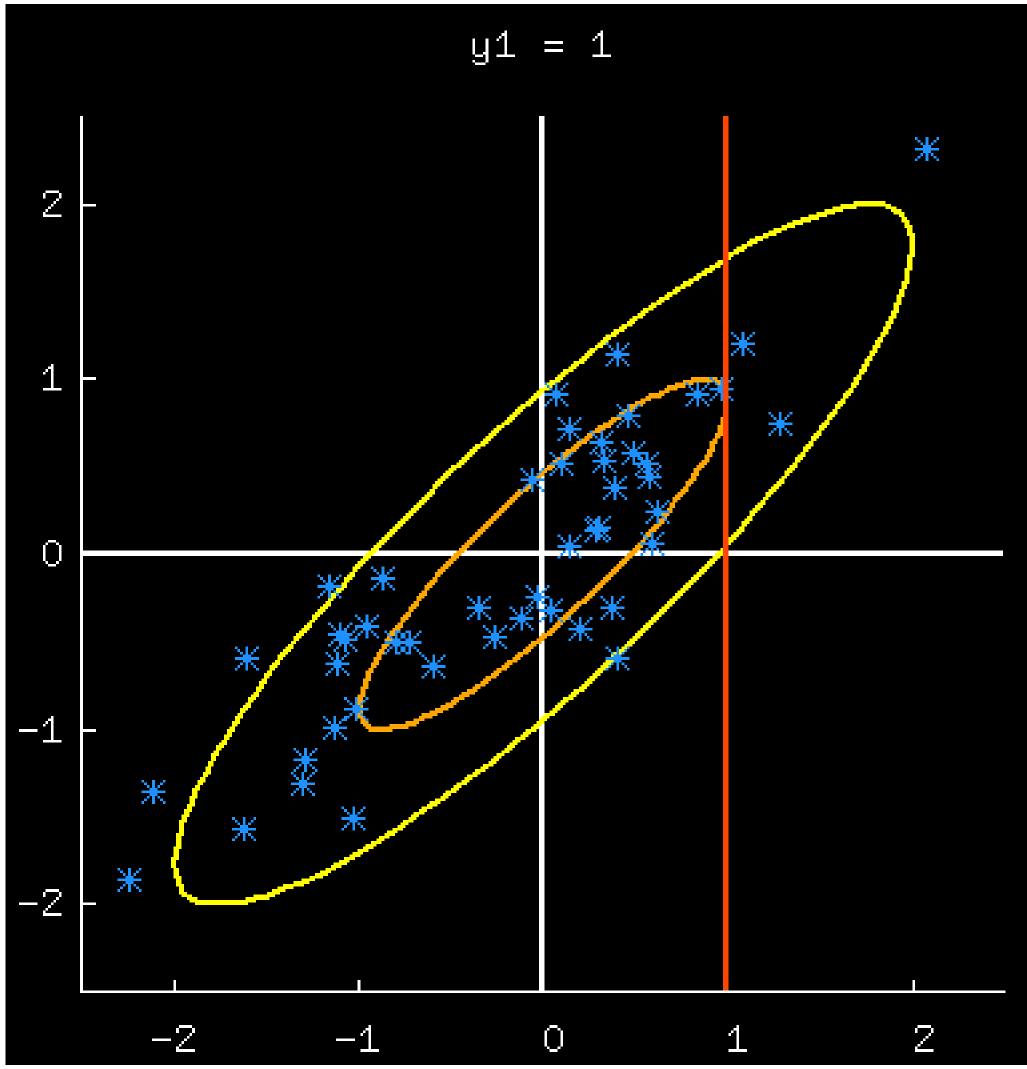
$$\mathbf{K} = \begin{bmatrix} 1.00010 & 0.32465 \\ 0.32465 & 1.00010 \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} 1.00010 & 0.98621 \\ 0.98621 & 1.00010 \end{bmatrix}$$



Inference

- Posterior, conditional on $\mathbf{y}_1$, is Gaussian with mean that depends on $\mathbf{y}_1$ and $\mathbf{K}$



Inference

Write $\mathbf{K}^{-1} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{A}_2 \end{bmatrix}$.

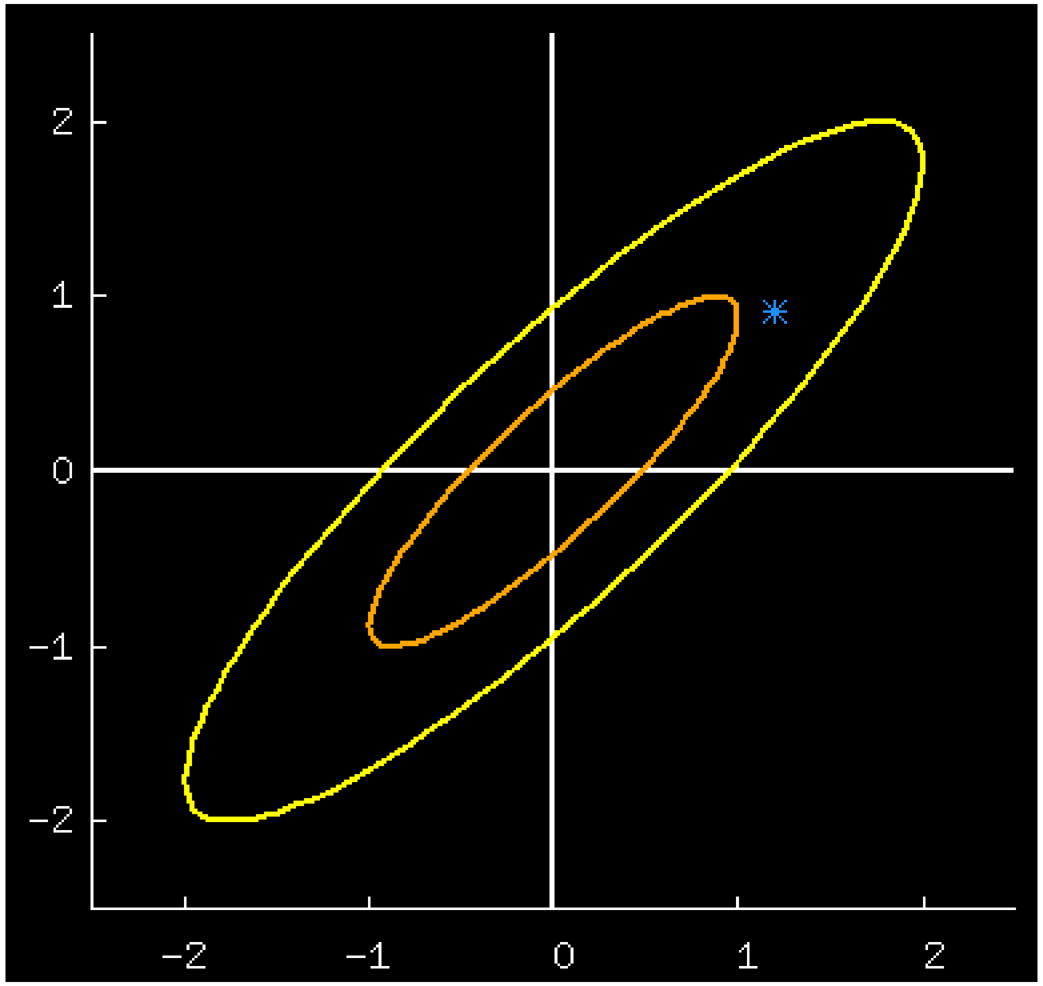
$$
\begin{aligned}
P(\mathbf{y}_2 \,|\, \mathbf{y}_1, \mathbf{K}) &= \frac{P(\mathbf{y}_1, \mathbf{y}_2 \,|\, \mathbf{K})}{P(\mathbf{y}_1 \,|\, \mathbf{K})} \\
&\propto \exp -\frac{1}{2} \begin{bmatrix} \mathbf{y}_1^\top & \mathbf{y}_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \\
&\propto \exp -\frac{1}{2} \begin{bmatrix} (\mathbf{y}_2 - \overline{\mathbf{y}}_2)^\top \end{bmatrix} \begin{bmatrix} \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} (\mathbf{y}_2 - \overline{\mathbf{y}}_2) \end{bmatrix},
\end{aligned}
$$

where

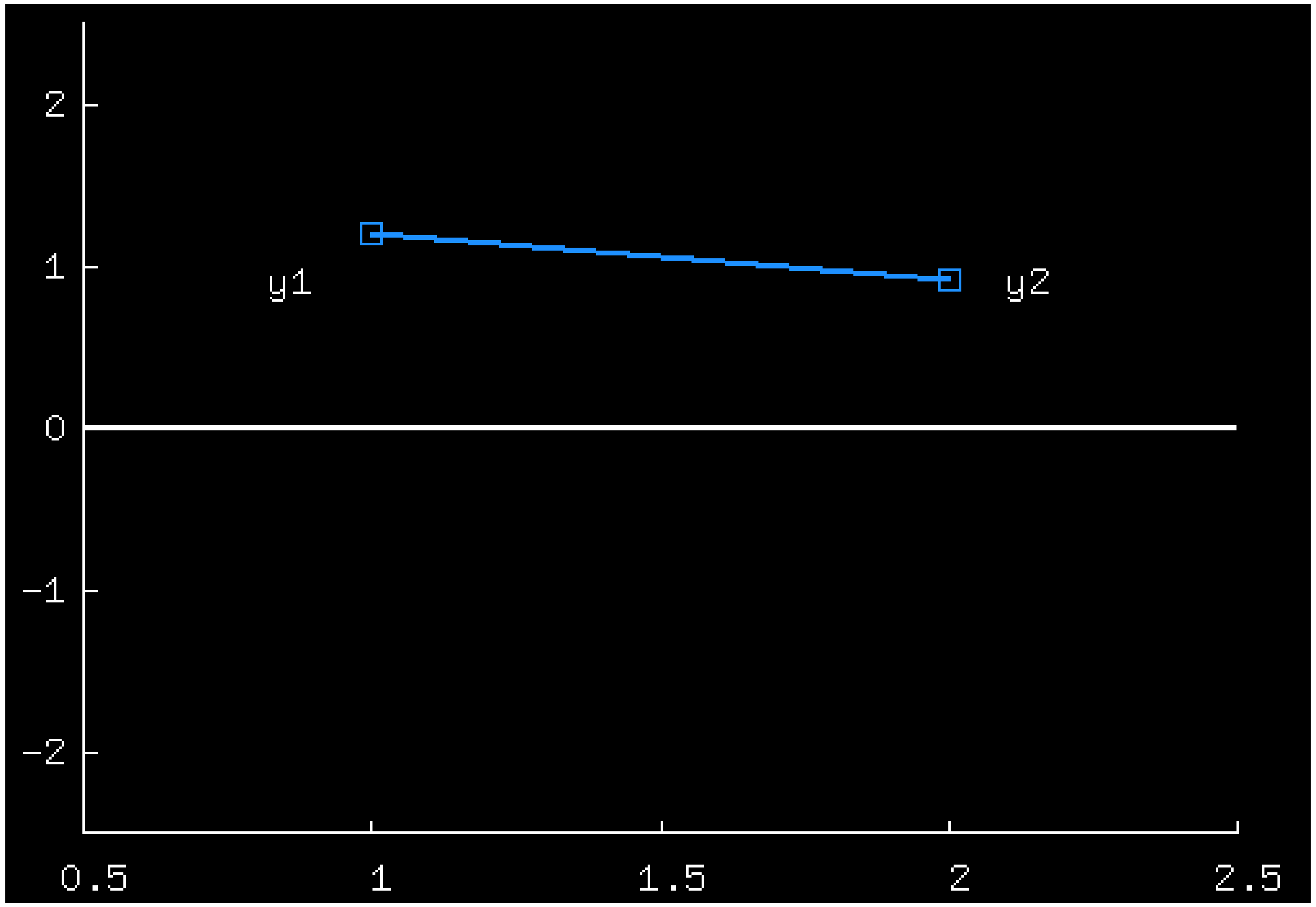
$$
\begin{aligned}
\text{mean } \overline{\mathbf{y}}_2 &= \mathbf{A}_2^{-1} \mathbf{B}^\top \mathbf{y}_1 \\
\text{posterior variance} &= \mathbf{A}_2^{-1}
\end{aligned}
$$

just matrix algebra

Another representation

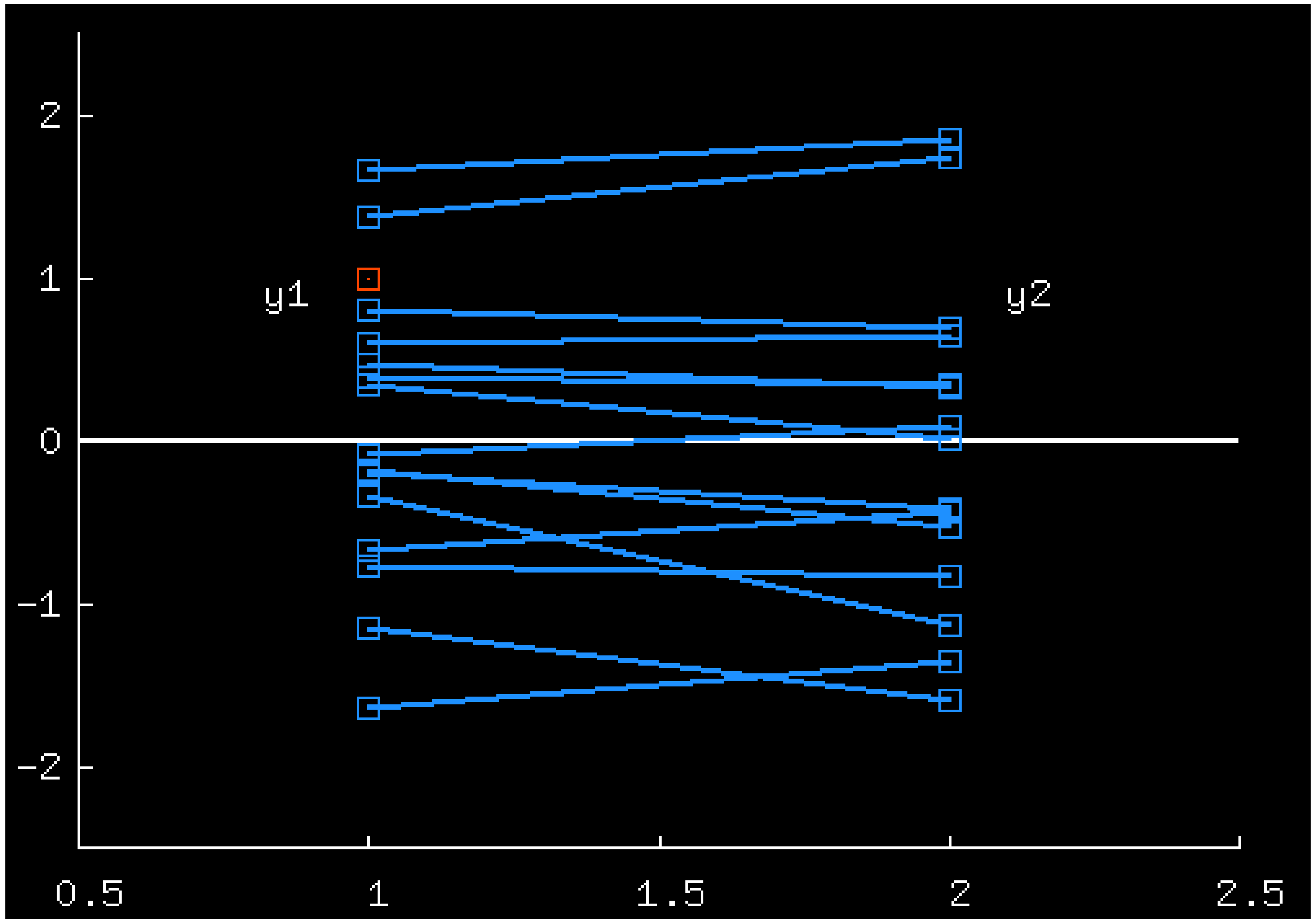


$$\mathbf{y} = \left[\begin{array}{cc} 1.2 & 0.9 \end{array}\right]$$



Now do the 2d demo again

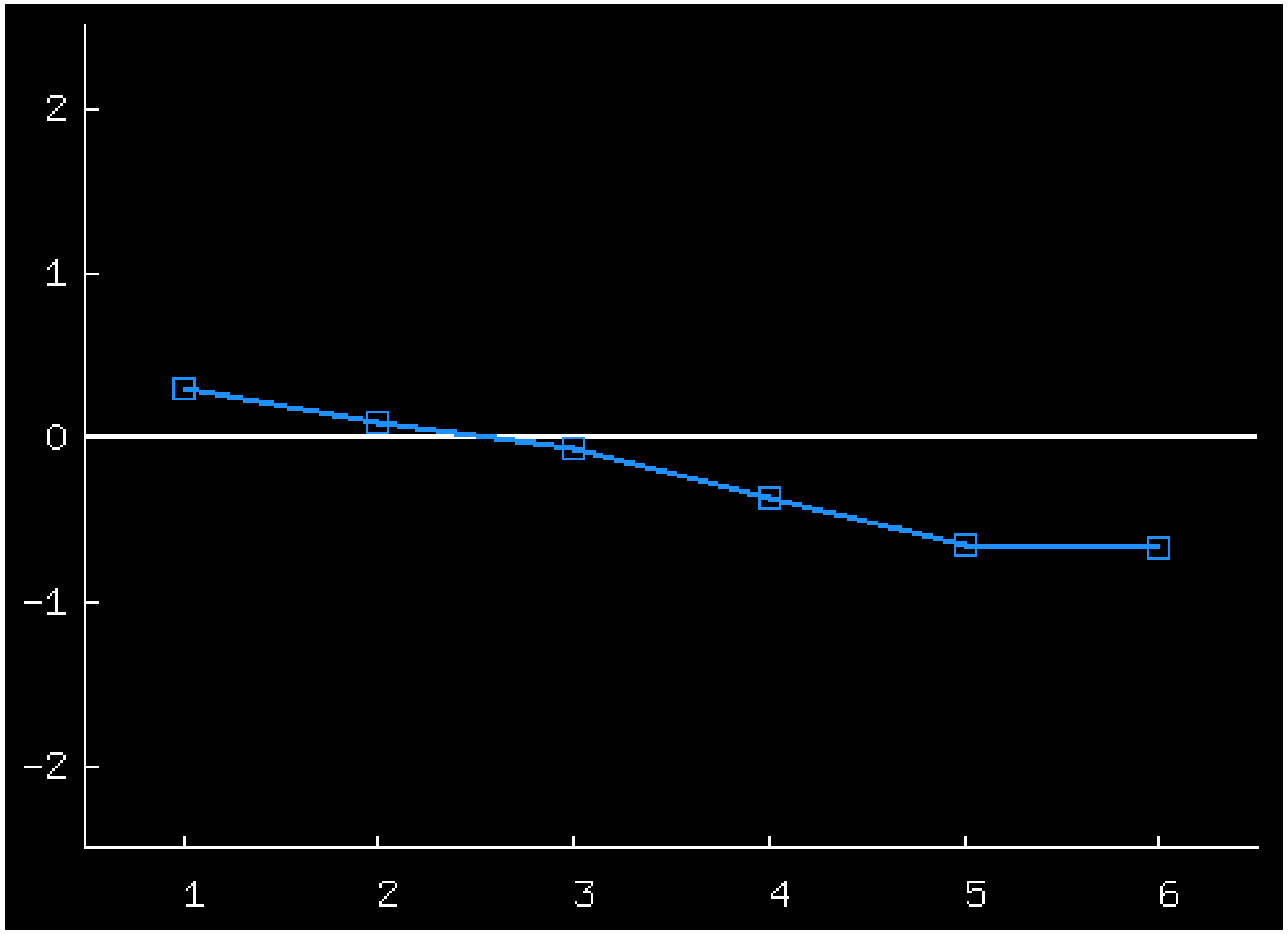
This representation allows visualization of higher-dimensional Gaussians



K =

1.000001	0.882497	0.606531	0.324652	0.135335	0.043937
0.882497	1.000001	0.882497	0.606531	0.324652	0.135335
0.606531	0.882497	1.000001	0.882497	0.606531	0.324652
0.324652	0.606531	0.882497	1.000001	0.882497	0.606531
0.135335	0.324652	0.606531	0.882497	1.000001	0.882497
0.043937	0.135335	0.324652	0.606531	0.882497	1.000001

This representation allows visualization of higher-dimensional Gaussians

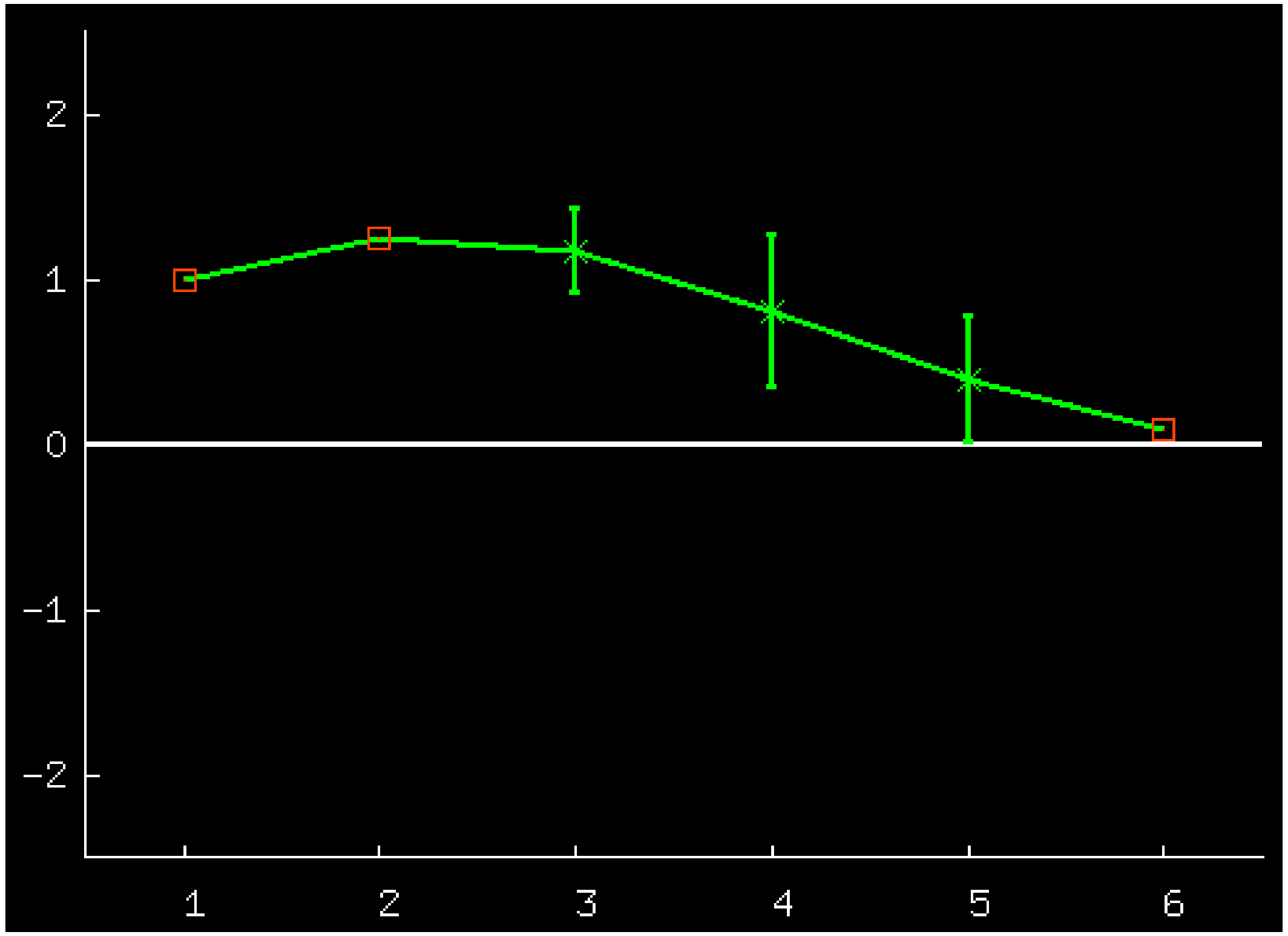


K =

1.000001	0.882497	0.606531	0.324652	0.135335	0.043937
0.882497	1.000001	0.882497	0.606531	0.324652	0.135335
0.606531	0.882497	1.000001	0.882497	0.606531	0.324652
0.324652	0.606531	0.882497	1.000001	0.882497	0.606531
0.135335	0.324652	0.606531	0.882497	1.000001	0.882497
0.043937	0.135335	0.324652	0.606531	0.882497	1.000001

Aha!

- Looks like nonlinear regression



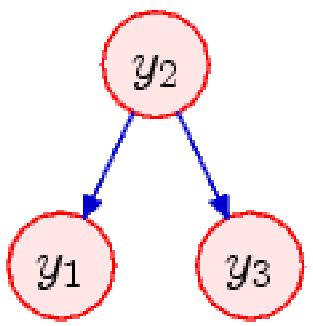
So, where did this 6x6 matrix come from?

K =

1.000001	0.882497	0.606531	0.324652	0.135335	0.043937
0.882497	1.000001	0.882497	0.606531	0.324652	0.135335
0.606531	0.882497	1.000001	0.882497	0.606531	0.324652
0.324652	0.606531	0.882497	1.000001	0.882497	0.606531
0.135335	0.324652	0.606531	0.882497	1.000001	0.882497
0.043937	0.135335	0.324652	0.606531	0.882497	1.000001

Gaussian quiz

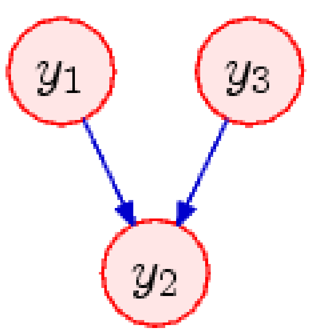
$\mathcal{H}_1$



1. Assuming the variables in H1 have a joint Gaussian distribution, which of the following could be the covariance matrix?

$$\text{A} \begin{bmatrix} 9 & 3 & 1 \\ 3 & 9 & 3 \\ 1 & 3 & 9 \end{bmatrix} \quad \text{B} \begin{bmatrix} 8 & -3 & 1 \\ -3 & 9 & -3 \\ 1 & -3 & 8 \end{bmatrix} \quad \text{C} \begin{bmatrix} 9 & 3 & 0 \\ 3 & 9 & 3 \\ 0 & 3 & 9 \end{bmatrix} \quad \text{D} \begin{bmatrix} 9 & -3 & 0 \\ -3 & 10 & -3 \\ 0 & -3 & 9 \end{bmatrix}$$

$\mathcal{H}_2$



2. which of the matrices could be the inverse covariance matrix?

3. Now H2's variables: which of the matrices could be the covariance matrix?

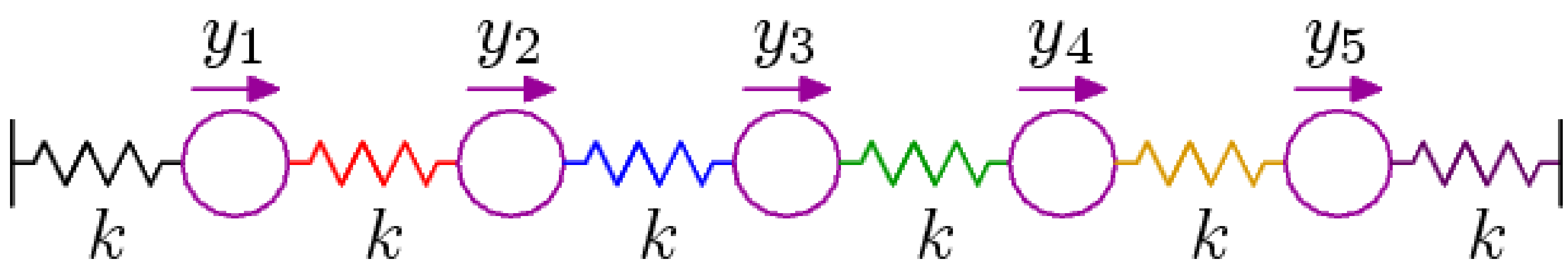
4. which of the matrices could be the inverse covariance matrix?

5. Let y_1, y_2, y_3 have covariance matrix $\mathbf{K}_{(3)} = \begin{bmatrix} 1 & .5 & 0 \\ .5 & 1 & .5 \\ 0 & .5 & 1 \end{bmatrix}$ and inverse $\mathbf{K}_{(3)}^{-1} = \begin{bmatrix} 1.5 & -1 & .5 \\ -1 & 2 & -1 \\ .5 & -1 & 1.5 \end{bmatrix}$.

Focus on the variables y_1 and y_2. Which statements about *their* covariance matrix $\mathbf{K}_{(2)}$ and inverse covariance matrix $\mathbf{K}_{(2)}^{-1}$ are true?

$$\text{(A)} \quad \mathbf{K}_{(2)} = \begin{bmatrix} 1 & .5 \\ .5 & 1 \end{bmatrix} \qquad \text{(B)} \quad \mathbf{K}_{(2)}^{-1} = \begin{bmatrix} 1.5 & -1 \\ -1 & 2 \end{bmatrix}$$

How do we build a Gaussian distribution?



inverse-covariance matrix or covariance matrix?

$$\mathbf{K}^{-1} = \frac{k}{T}\begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \qquad \mathbf{K} = \frac{T}{k}\begin{bmatrix} 0.83 & 0.67 & 0.50 & 0.33 & 0.17 \\ 0.67 & 1.33 & 1.00 & 0.67 & 0.33 \\ 0.50 & 1.00 & 1.50 & 1.00 & 0.50 \\ 0.33 & 0.67 & 1.00 & 1.33 & 0.67 \\ 0.17 & 0.33 & 0.50 & 0.67 & 0.83 \end{bmatrix}$$

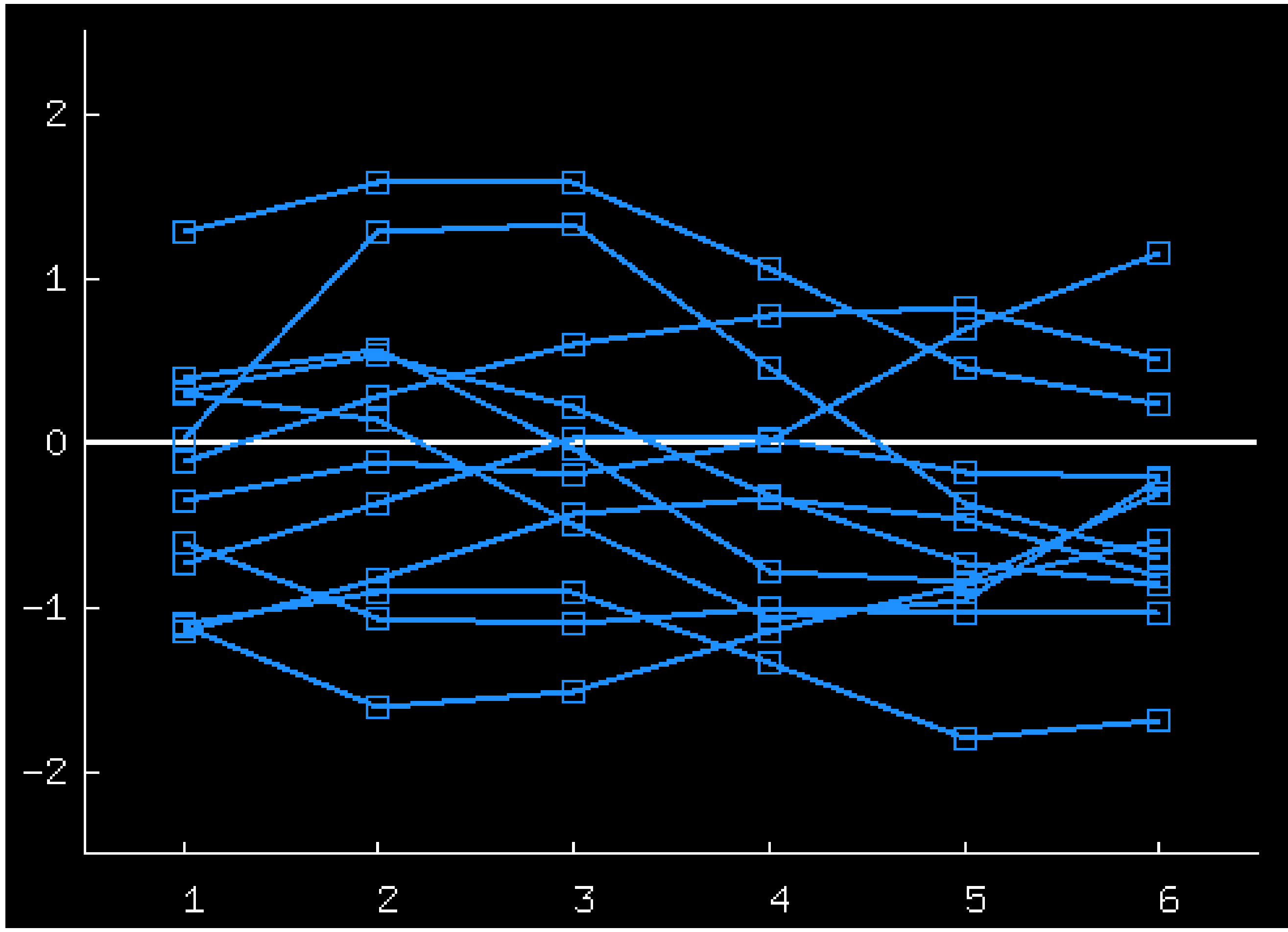
How the matrix was made

$$\text{cov}(y_n, y_{n'}) = k(\mathbf{x}_n, \mathbf{x}_{n'}) + \sigma_\nu^2 \delta_{nn'}$$

$$k(\mathbf{x}_n, \mathbf{x}_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x}_n - \mathbf{x}_{n'})^2\right)$$

'squared, exponential' covariance function

Model	$\mathbf{x} \rightarrow y = f(\mathbf{x}) + \epsilon$
Noise level	σ_ν^2
Horizontal lengthscale	l
Vertical lengthscale	σ_f



K =

1.000001	0.882497	0.606531	0.324652	0.135335	0.043937
0.882497	1.000001	0.882497	0.606531	0.324652	0.135335
0.606531	0.882497	1.000001	0.882497	0.606531	0.324652
0.324652	0.606531	0.882497	1.000001	0.882497	0.606531
0.135335	0.324652	0.606531	0.882497	1.000001	0.882497
0.043937	0.135335	0.324652	0.606531	0.882497	1.000001

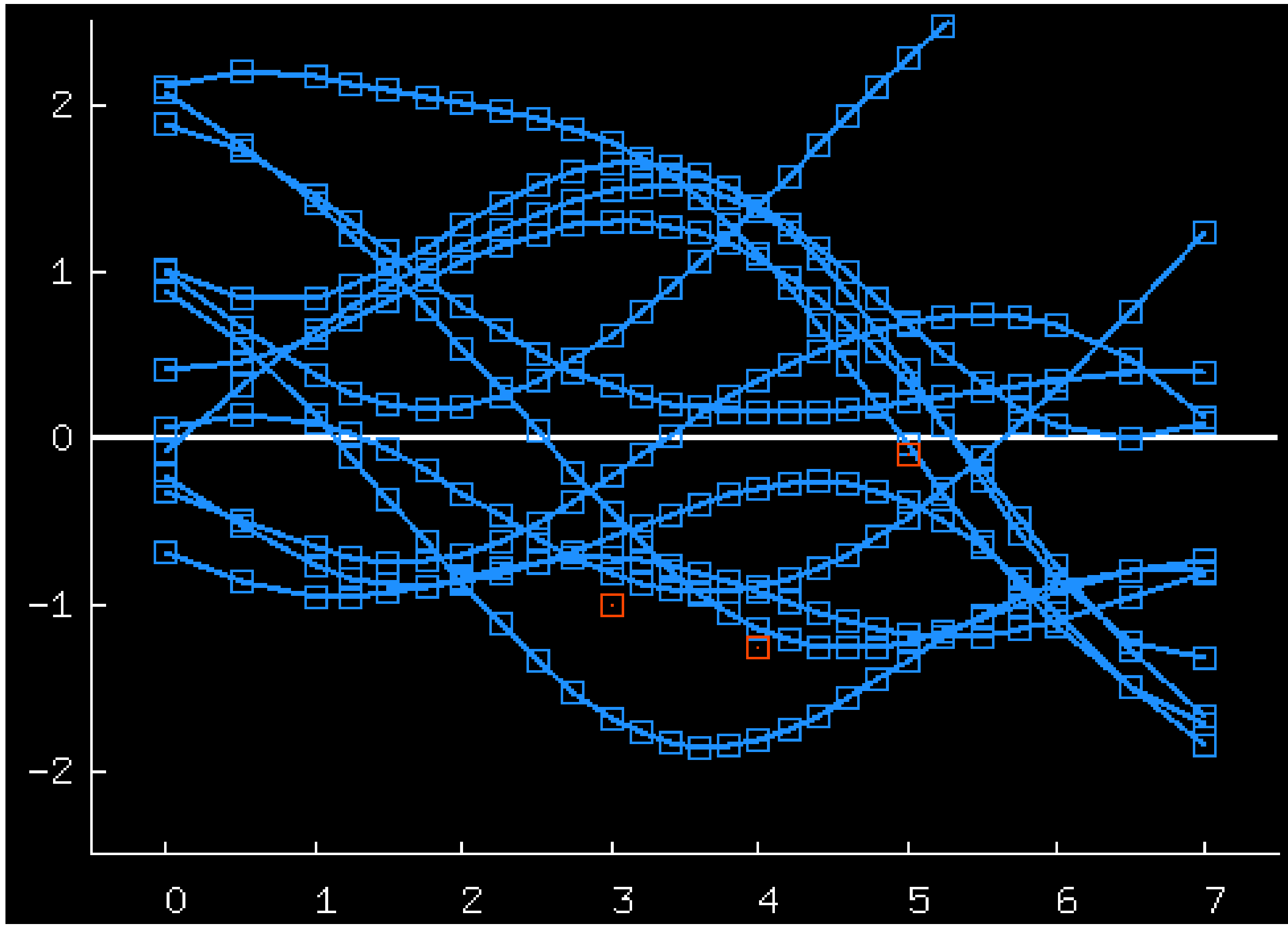
Extend to more points

$$\text{cov}\left(y_n, y_{n'}\right) = k(\mathbf{x}_n, \mathbf{x}_{n'}) + \sigma_\nu^2 \delta_{nn'}$$

$$k(\mathbf{x}_n, \mathbf{x}_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\left(\mathbf{x}_n - \mathbf{x}_{n'}\right)^2\right)$$

'squared, exponential' covariance function

Model	$\mathbf{x} \rightarrow y = f(\mathbf{x}) + \epsilon$
Noise level	σ_ν^2
Horizontal lengthscale	l
Vertical lengthscale	σ_f



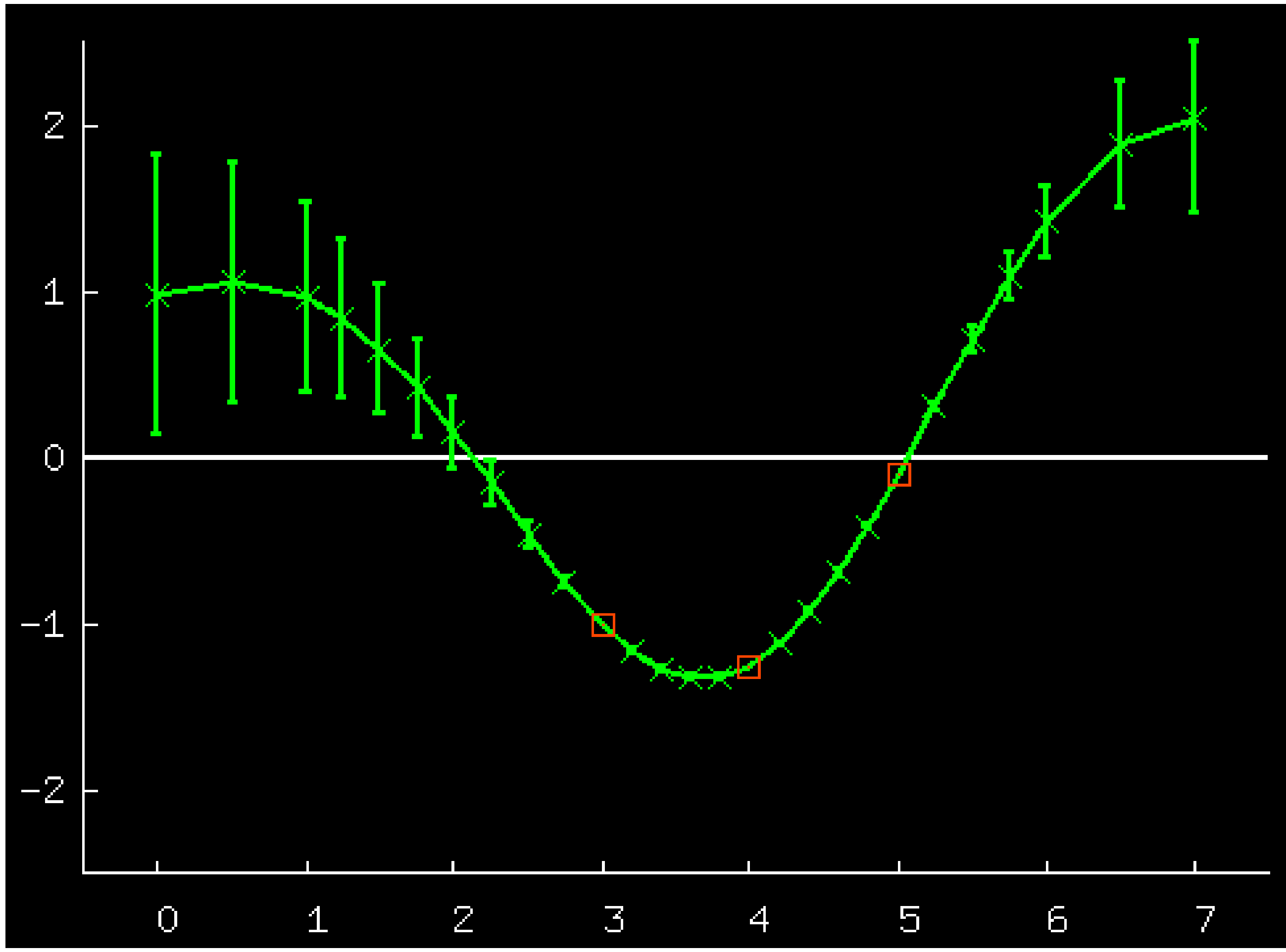
A Gaussian process

$$\text{cov}\left(y_n, y_{n'}\right) = k(\mathbf{x}_n, \mathbf{x}_{n'}) + \sigma_\nu^2 \delta_{nn'}$$

$$k(\mathbf{x}_n, \mathbf{x}_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\left(\mathbf{x}_n - \mathbf{x}_{n'}\right)^2\right)$$

'squared, exponential' covariance function

Model	$\mathbf{x} \rightarrow y = f(\mathbf{x}) + \epsilon$
Noise level	σ_ν^2
Horizontal lengthscale	l
Vertical lengthscale	σ_f



A Gaussian process is a collection of random variables with the property that the joint distribution of any finite subset is a Gaussian

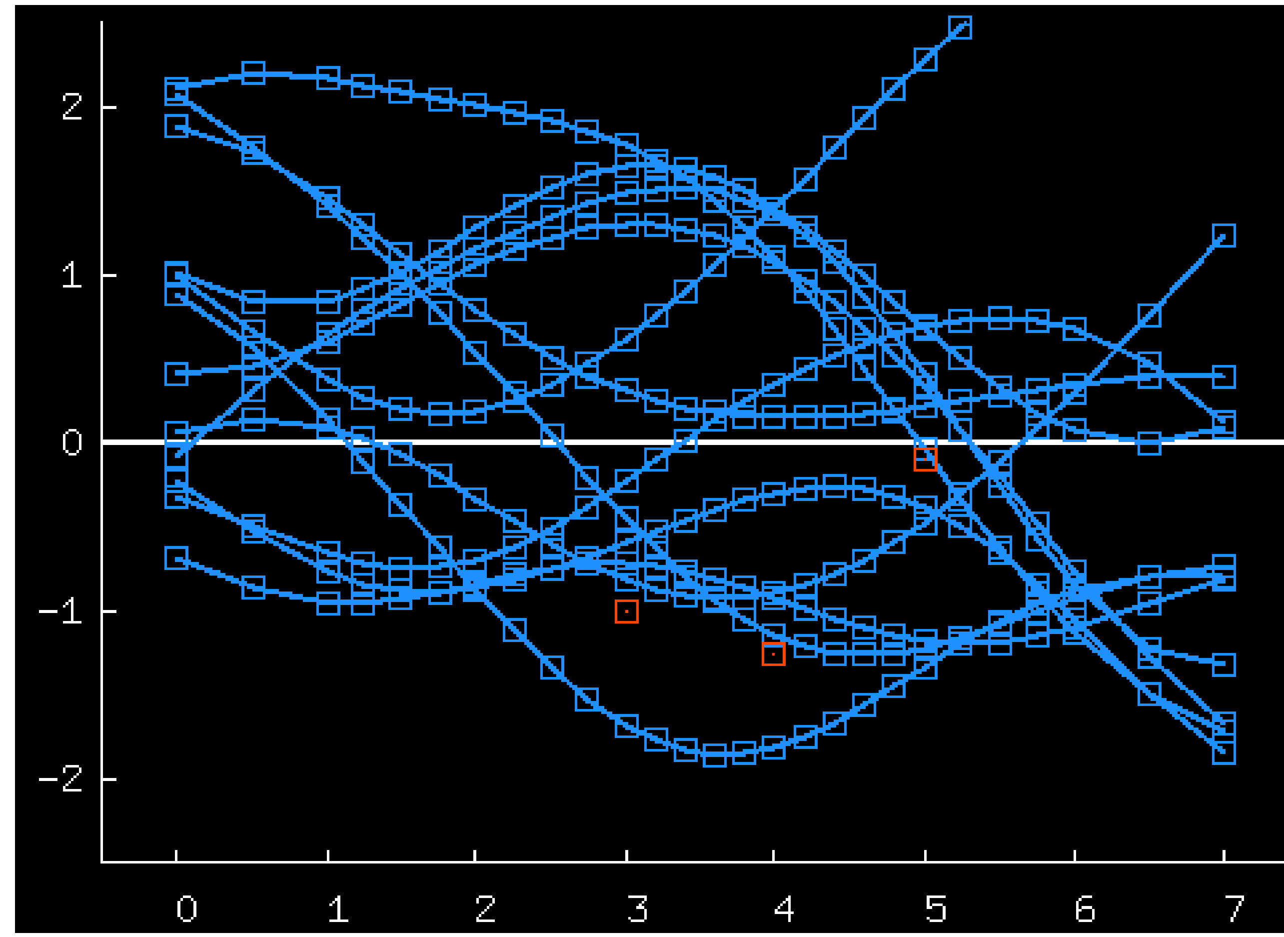
Effect of hyperparameters

$$\mathrm{cov}\left(y_n, y_{n'}\right) = k(\mathbf{x}_n, \mathbf{x}_{n'}) + \sigma_\nu^2 \delta_{nn'}$$

$$k(\mathbf{x}_n, \mathbf{x}_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\left(\mathbf{x}_n - \mathbf{x}_{n'}\right)^2\right)$$

'squared, exponential' covariance function

Model	$\mathbf{x} \rightarrow y = f(\mathbf{x}) + \epsilon$
Noise level	σ_ν^2
Horizontal lengthscale	l
Vertical lengthscale	σ_f



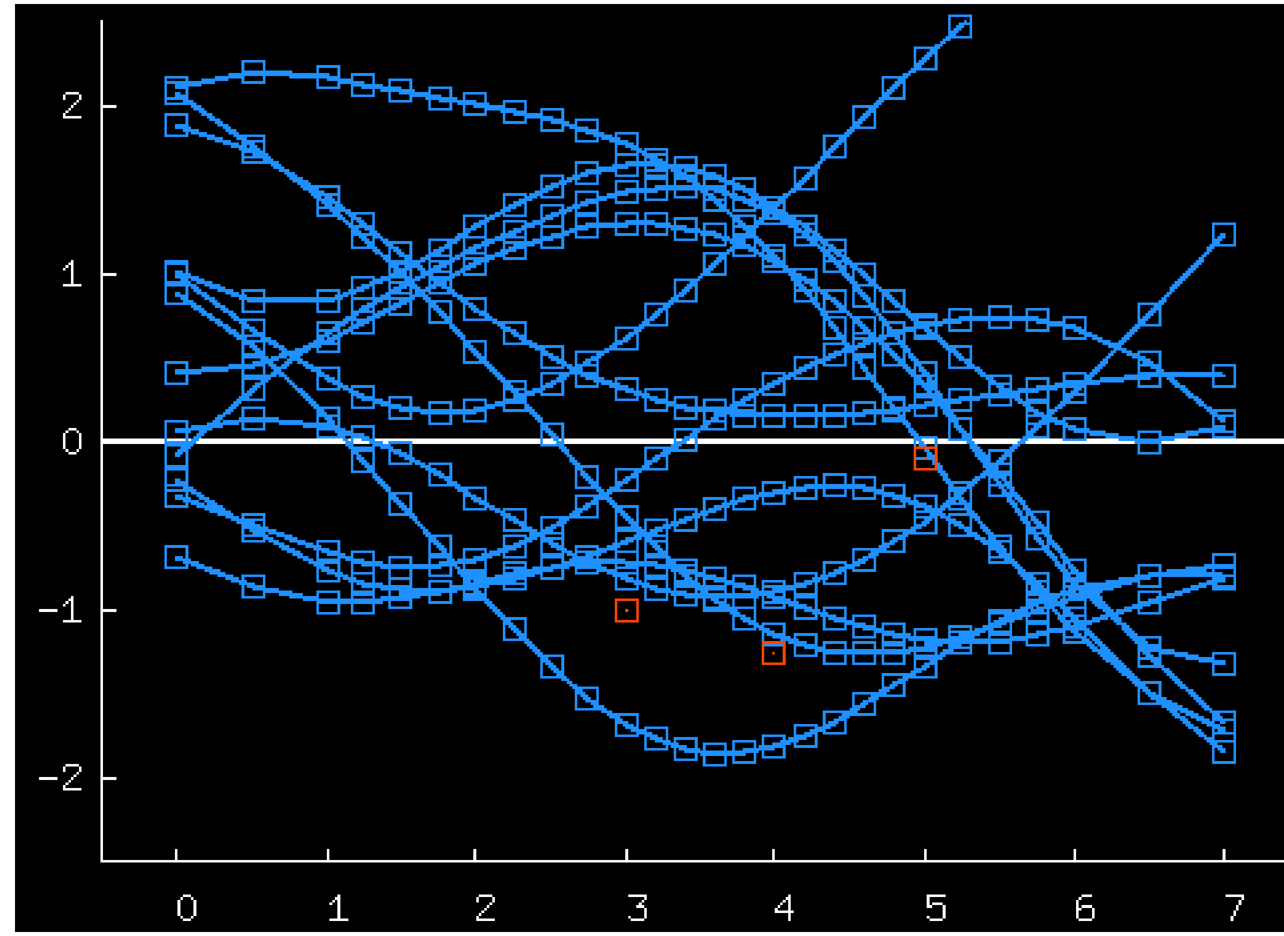
Inference of hyperparameters

$$\text{cov}\,(y_n, y_{n'}) = k(\mathbf{x}_n, \mathbf{x}_{n'}) + \sigma_\nu^2 \delta_{nn'}$$

$$k(\mathbf{x}_n, \mathbf{x}_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\,(\mathbf{x}_n - \mathbf{x}_{n'})^2\right)$$

'squared, exponential' covariance function

Model	$\mathbf{x} \to y = f(\mathbf{x}) + \epsilon$
Noise level	σ_ν^2
Horizontal lengthscale	l
Vertical lengthscale	σ_f

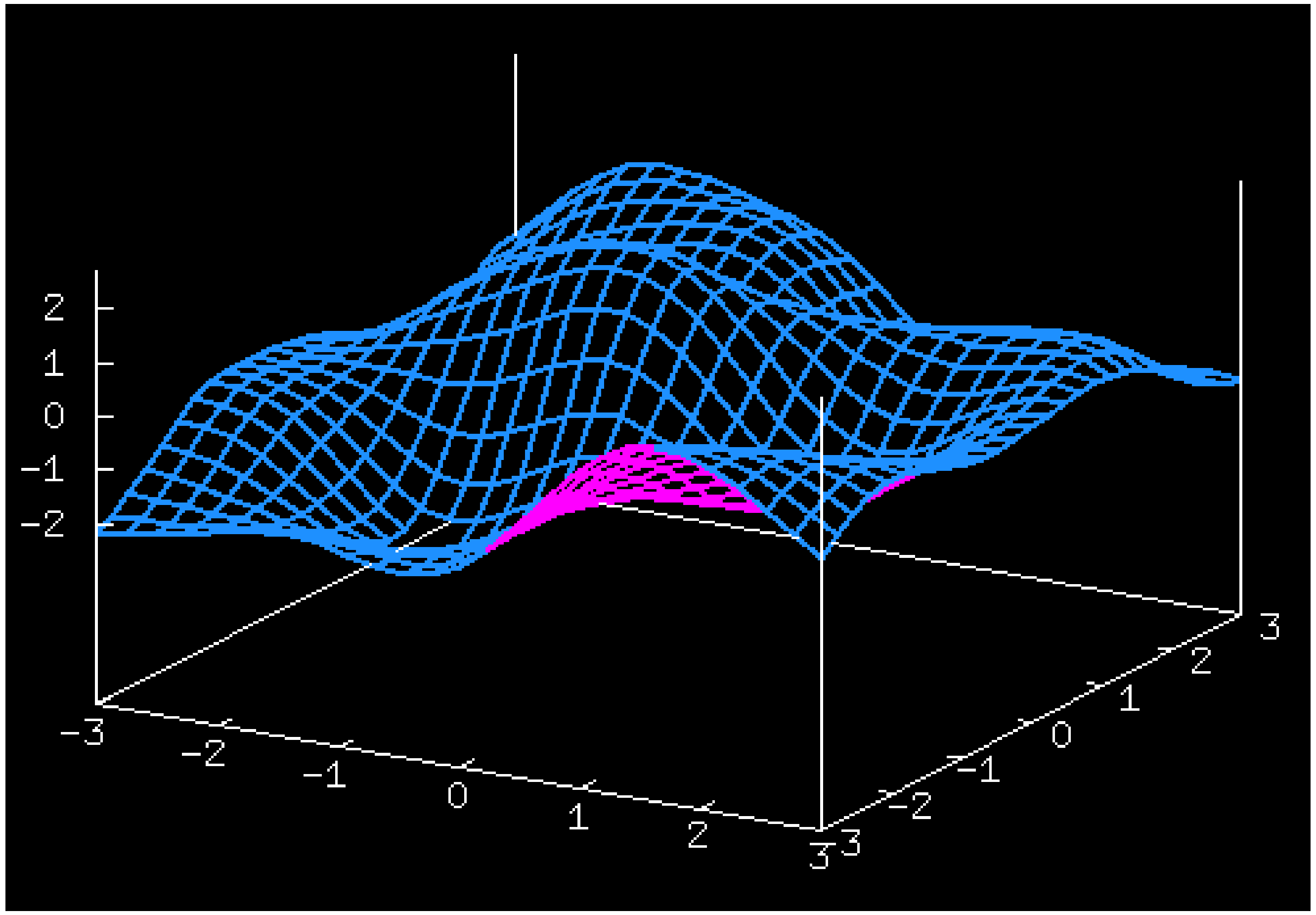


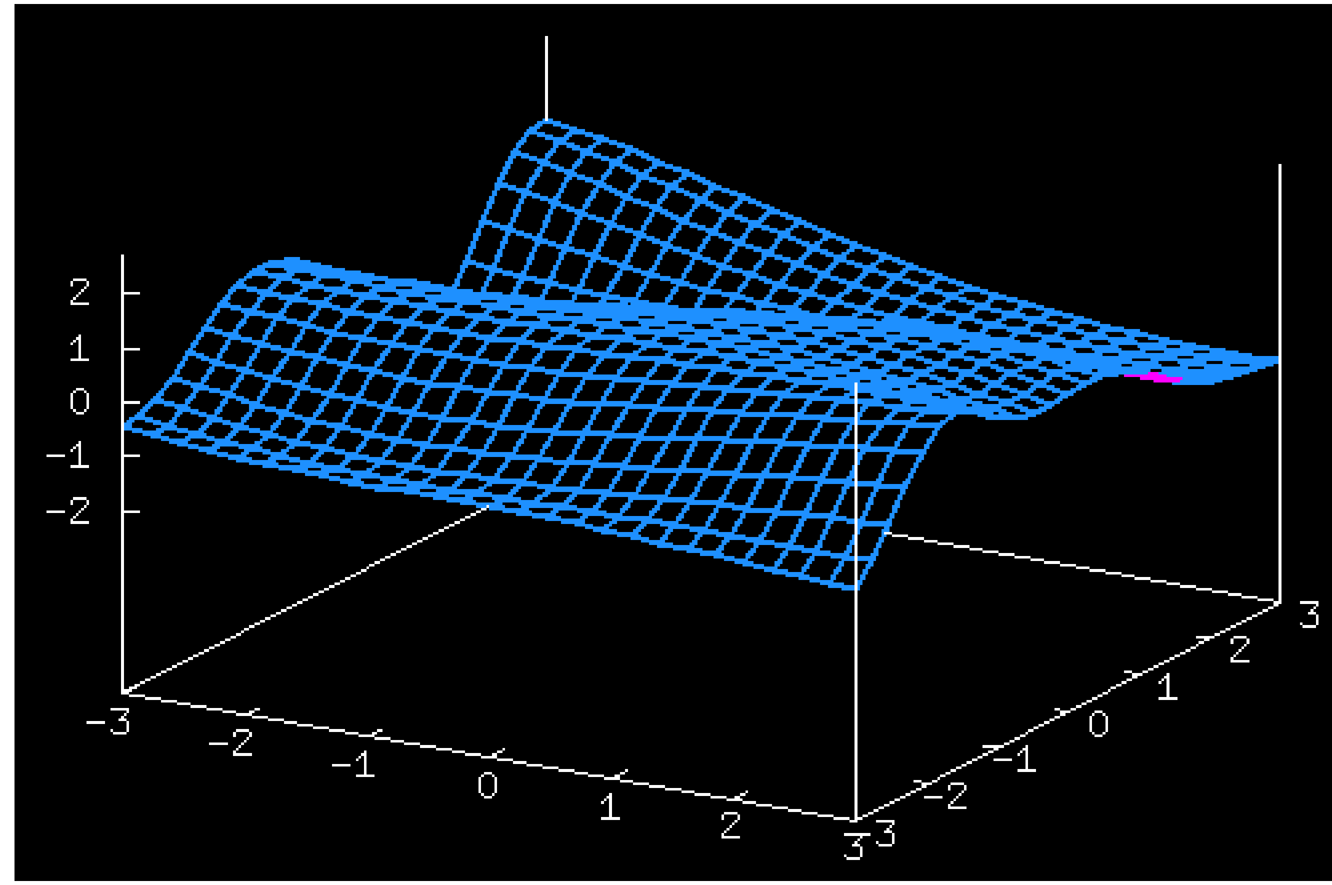
$$P(\theta \,|\, \mathbf{y}_{1:N}) \propto P(\mathbf{y}_{1:N} \,|\, \theta) P(\theta)$$

$$P(\mathbf{y} \,|\, \theta) = \frac{1}{[\det 2\pi \mathbf{K}(\theta)]^{1/2}} e^{-\frac{1}{2}\mathbf{y}^{\mathsf{T}}\mathbf{K}^{-1}(\theta)\mathbf{y}}$$

Including inferring the noise level

Two-dimensional input space

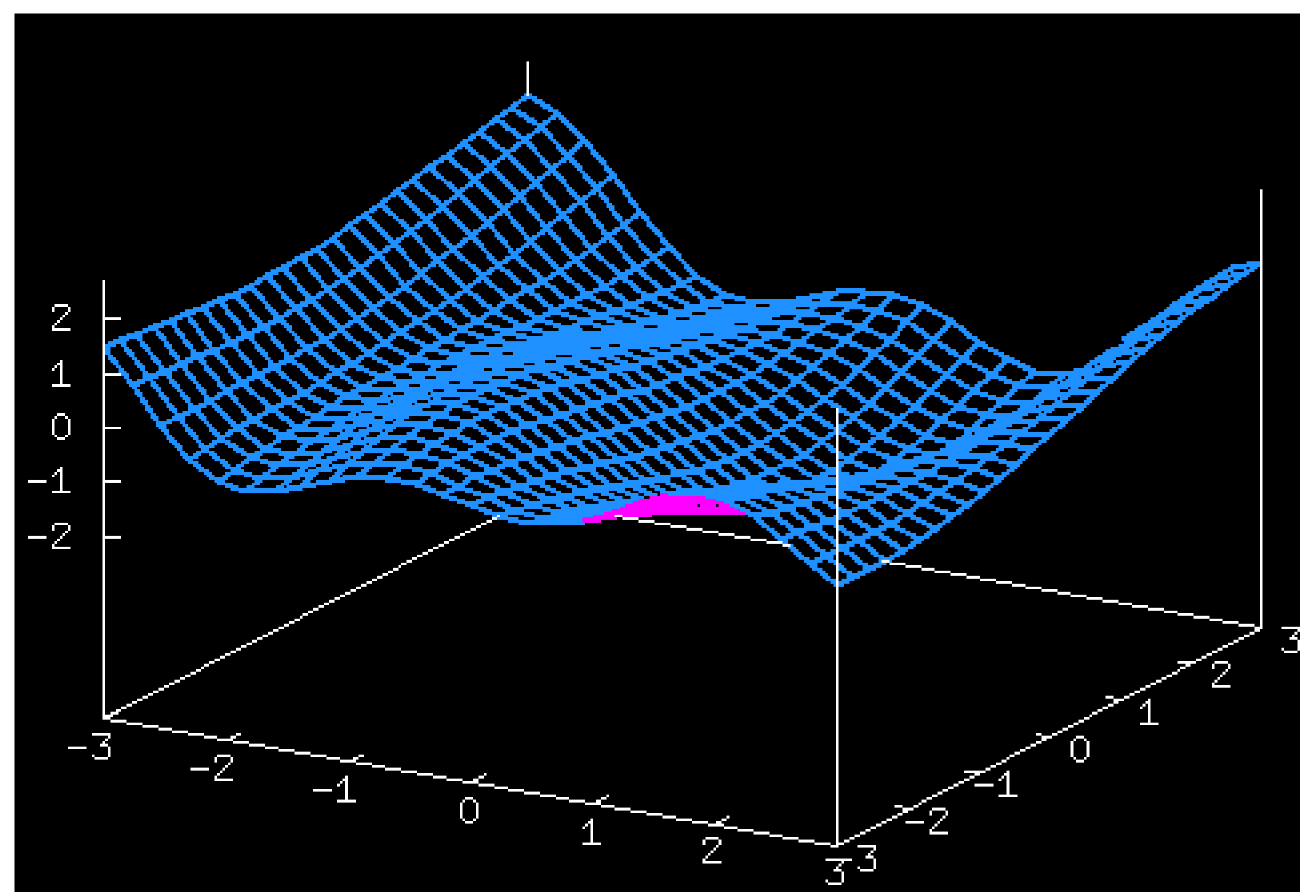




Automatic relevance determination

$$\mathrm{cov}\left(y_n, y_{n'}\right) = k(\mathbf{x}_n, \mathbf{x}_{n'}) + \sigma_\nu^2 \delta_{nn'}$$

$$k(\mathbf{x}_n, \mathbf{x}_{n'}) = \sigma_f^2 \exp\left(- \sum_{d=1}^{D} \frac{(x_{dn} - x_{dn'})^2}{2 l_d^2} \right)$$



Efficient computation (... well, modestly efficient)

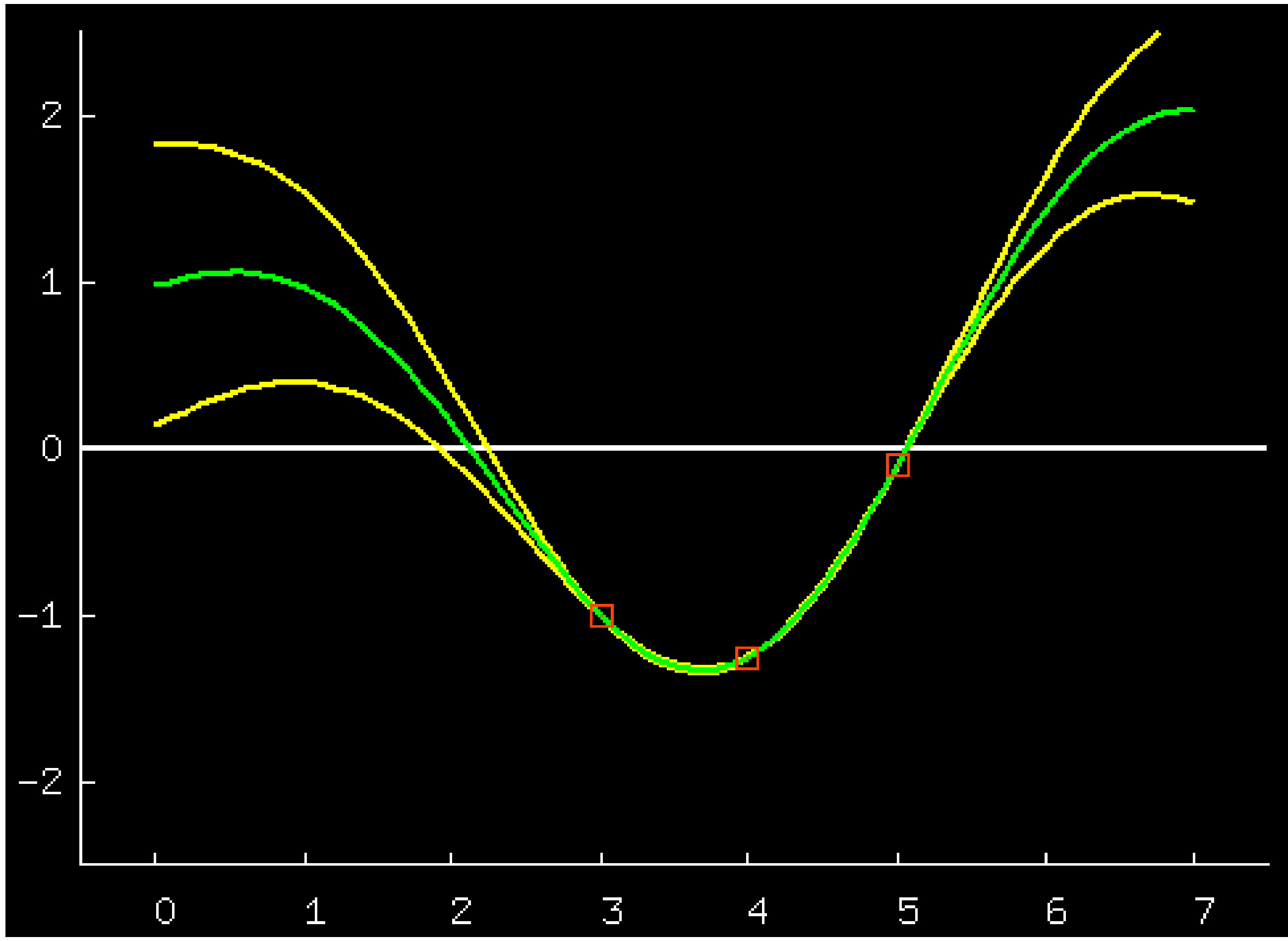
If $\mathbf{K}_{N+1} = \begin{bmatrix} \uparrow & \uparrow \\ \leftarrow \mathbf{K}_N \rightarrow & \mathbf{k} \\ \downarrow & \downarrow \\ \leftarrow \mathbf{k}^\top \rightarrow & \kappa \end{bmatrix}$

then $P(y_{N+1} \,|\, \mathbf{y}_{1:N}, \mathbf{K}_{N+1})$ is Gaussian,

$$\text{mean} = \mathbf{k}^\top \mathbf{K}_N^{-1} \mathbf{y}_{1:N}$$

$$\text{variance} = \kappa - \mathbf{k}^\top \mathbf{K}_N^{-1} \mathbf{k}$$

(p.16 in Rasmussen and Williams)



Can compute predictions (mean, variance)
at N^* new points with cost $(N + N^2)N^*$
instead of $(N + N^*)^3$

Key computational requirements

$$\text{for prediction} \begin{cases} \mathbf{k}^\top \mathbf{K}^{-1} \mathbf{y} \ (\text{where } \mathbf{K} \text{ is } N \times N) \\ \\ \mathbf{k}^\top \mathbf{K}^{-1} \mathbf{k}^\top \end{cases}$$

$$\begin{array}{c}\text{for hyperparameter optimization} \\ \text{or sampling}\end{array} \begin{cases} \mathrm{Trace}\left[\mathbf{K}^{-1}\mathbf{M}\right] \\ \\ \det\left[\mathbf{K}\right] \ (\text{perhaps}) \end{cases}$$

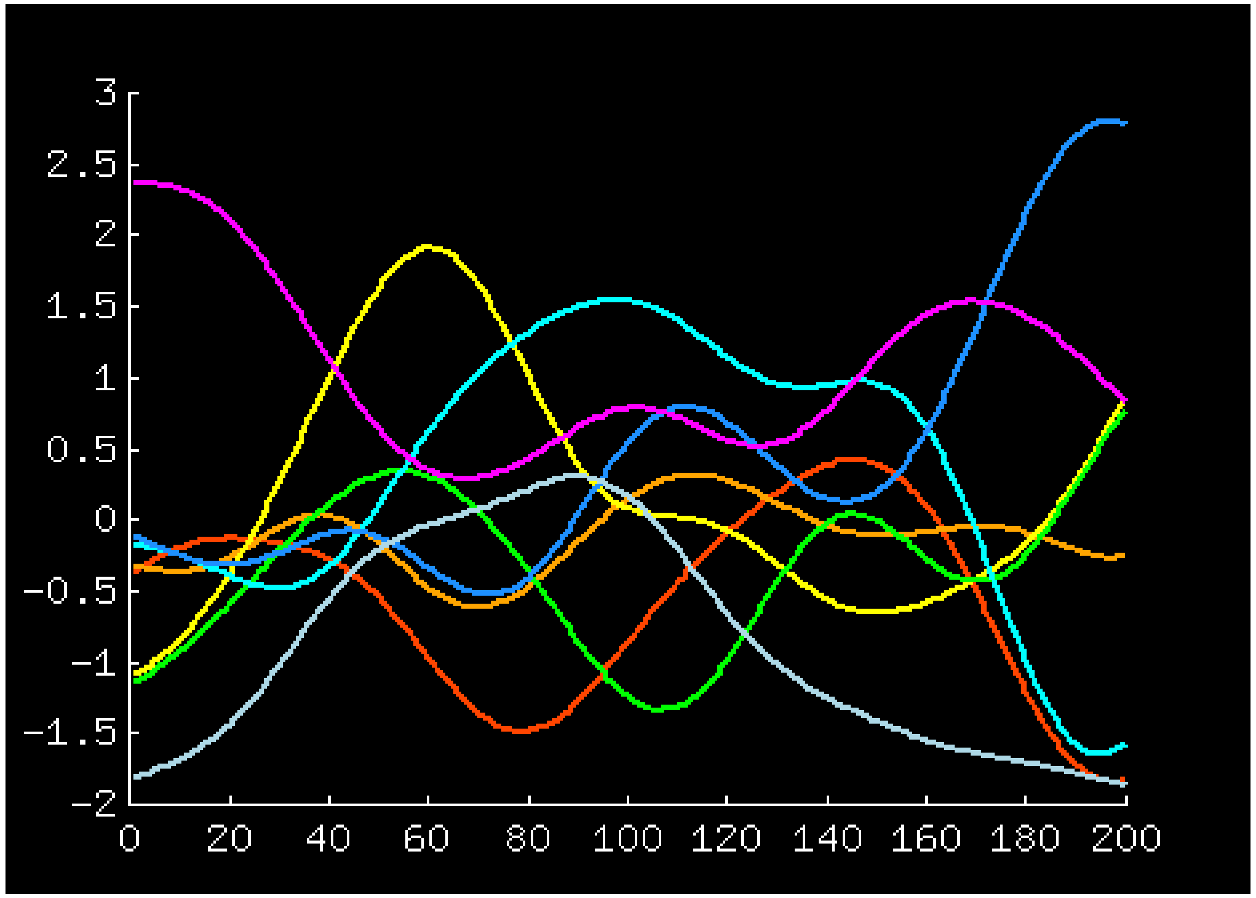
Choosing covariance functions

- Can think about your prior beliefs
  - Example: linear regression

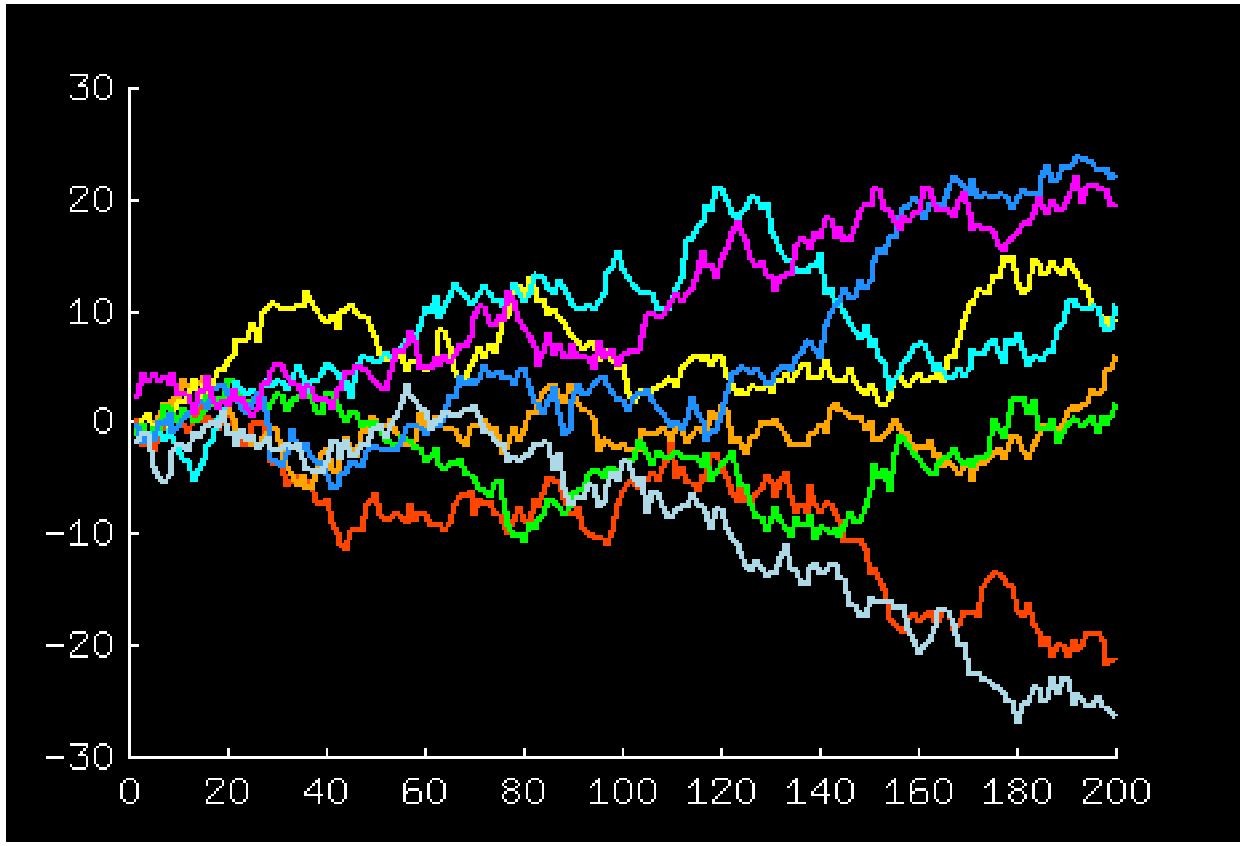
$$
\begin{aligned}
y_n &= f(x_n) + \nu_n \\
&= m x_n + c + \nu_n
\end{aligned}
$$

$$
\begin{aligned}
\langle y_n y_{n'} \rangle &= \langle (m x_n + c + \nu_n)(m x_{n'} + c + \nu_{n'}) \rangle \\
&= \underbrace{\overline{m^2} x_n x_{n'} + \overline{c^2}}_{\substack{\text{covariance function} \\ k(x_n, x_{n'})}} + \delta_{nn'} \sigma_\nu^2
\end{aligned}
$$

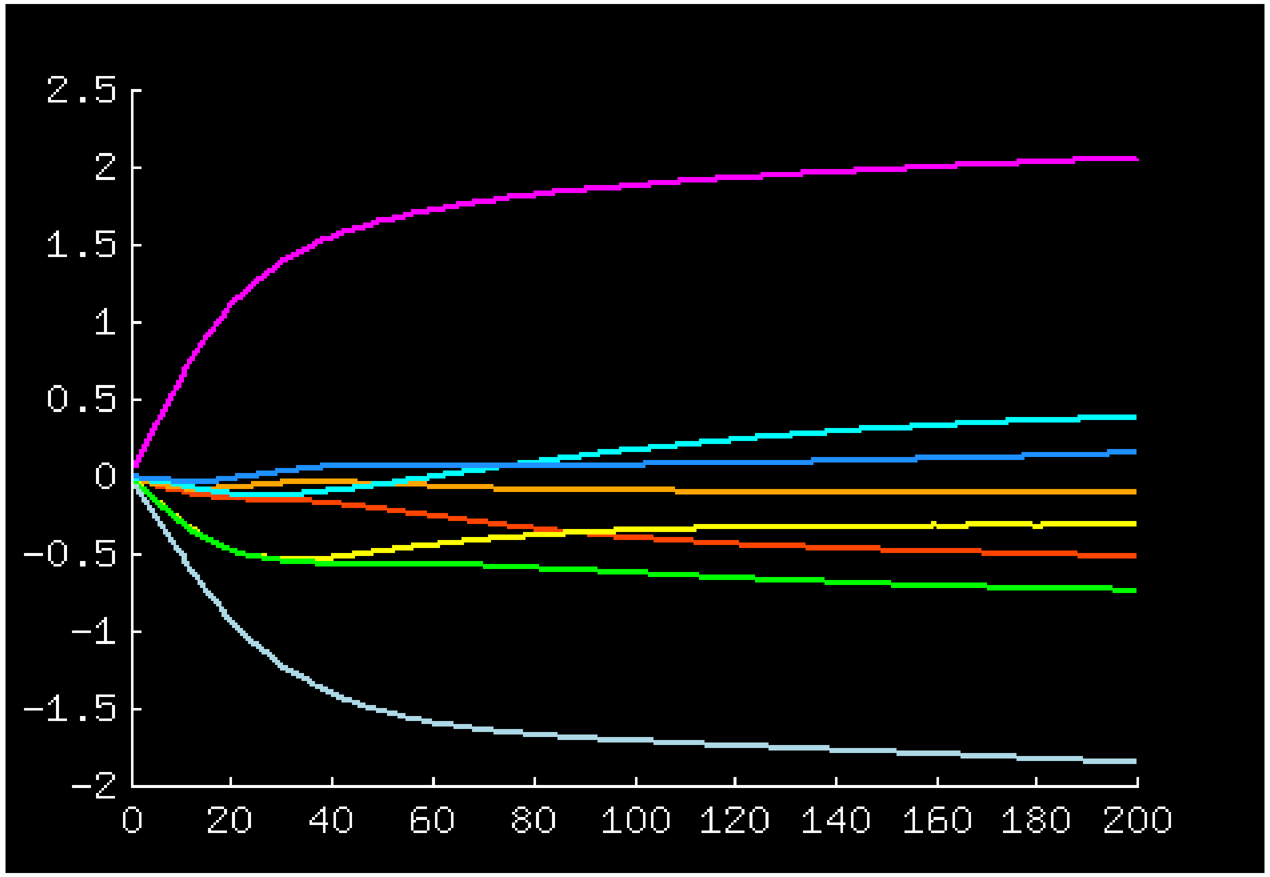
'Squared exponential'



Brownian

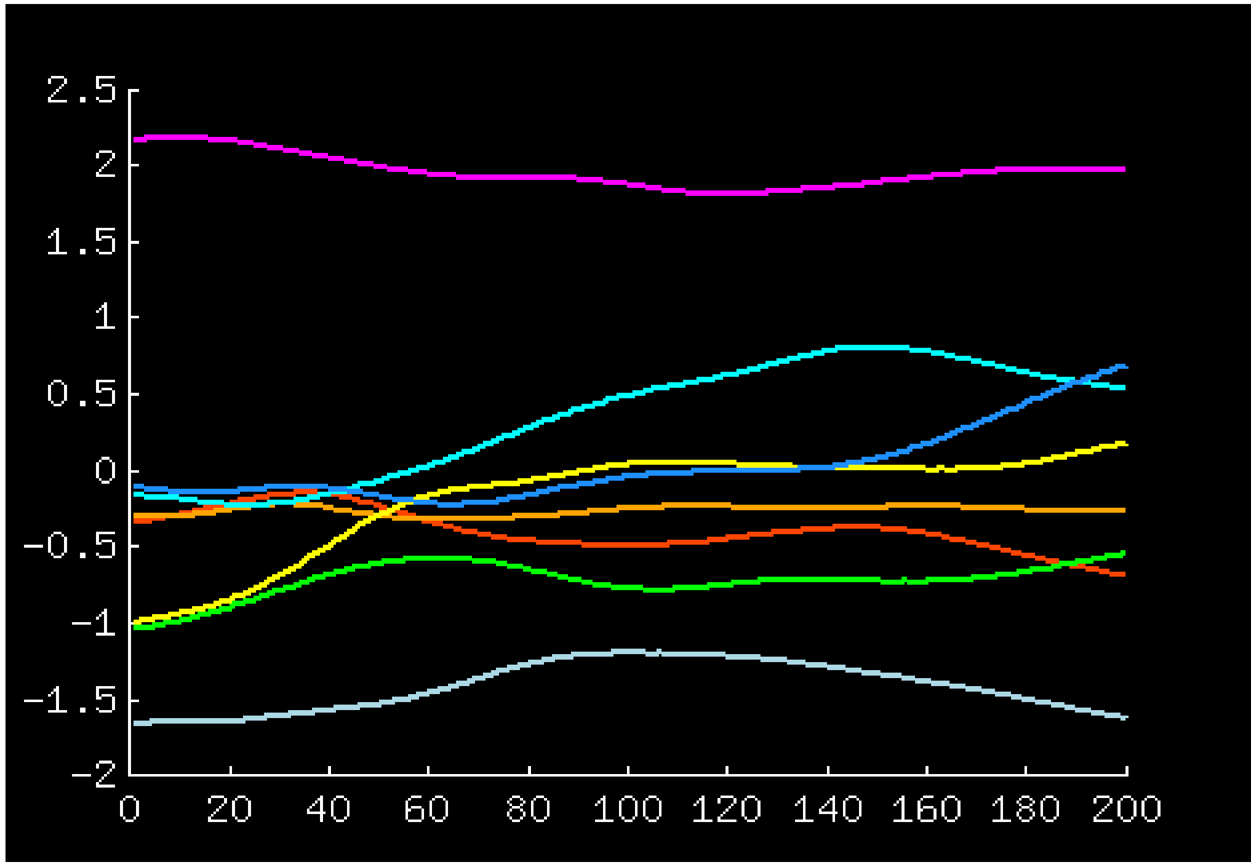


Emulate infinite neural networks



$$k_{\mathrm{NN}}(\mathbf{x}, \mathbf{x}') = \frac{2}{\pi} \sin^{-1}\left(\frac{2\mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x}}{\sqrt{(1 + 2\mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x})(1 + 2\mathbf{x}'^\top \boldsymbol{\Sigma} \mathbf{x}')}}\right)$$

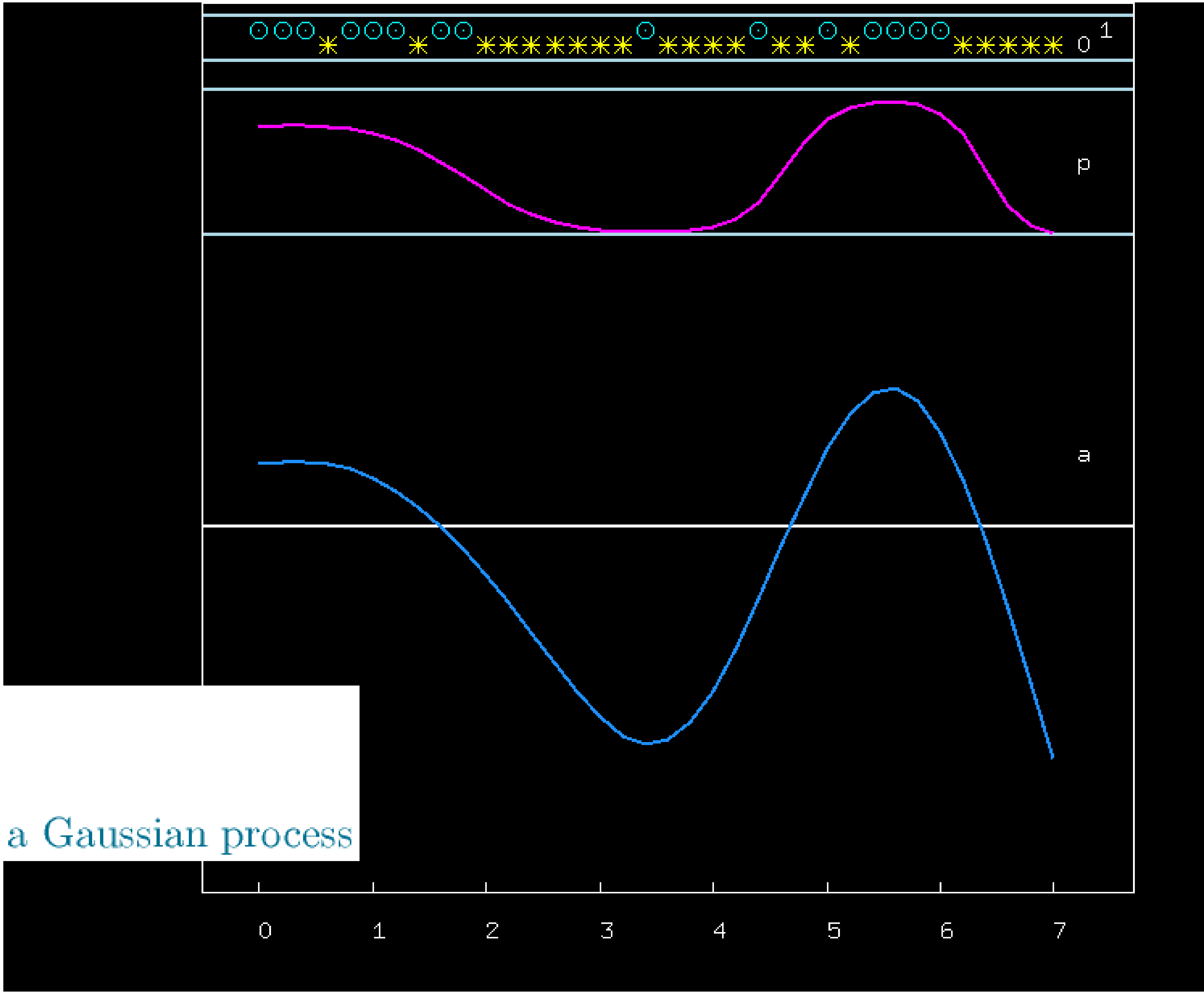
without biases



$$k_{\mathrm{NN}}(\mathbf{x}, \mathbf{x}') = \frac{2}{\pi} \sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \boldsymbol{\Sigma} \tilde{\mathbf{x}}}{\sqrt{(1 + 2\tilde{\mathbf{x}}^\top \boldsymbol{\Sigma} \tilde{\mathbf{x}})(1 + 2\tilde{\mathbf{x}}'^\top \boldsymbol{\Sigma} \tilde{\mathbf{x}}')}}\right)$$

with biases

GPs for classification

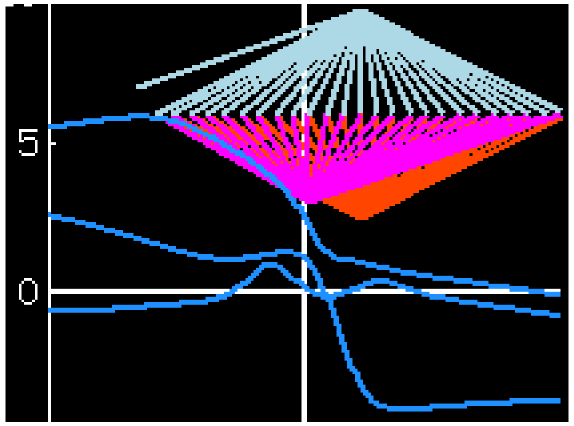


$$p(\mathbf{x}) = \frac{1}{1 + e^{-a(\mathbf{x})}}$$

where $a(\mathbf{x})$ comes from a Gaussian process

Connection to standard neural networks

Multi-layer perceptron with regularization



Gaussian process

Model	$\mathbf{x} \to y = f(\mathbf{x}; \mathbf{w}) + \epsilon$		Model	$\mathbf{x} \to y = f(\mathbf{x}) + \epsilon$
Parameters	$\mathbf{w}$			
Objective function	$M(\mathbf{w}) = \beta E_D(\mathbf{w}) + \sum_c \alpha^{(c)} E_W(\mathbf{w}^{(c)})$			
Optimization of $M(\mathbf{w})$			(matrix algebra)	
Noise level	$1/\beta$		Noise level	σ_ν^2
Weight decay rate (input weights)	$\alpha_d^{(\text{in})}$	$\sim$	Horizontal lengthscale	l_d
Weight decay rate (output weights)	$\alpha^{(\text{out})}$	$\sim$	Vertical lengthscale	σ_f

Gaussian processes compared with state-of-the-art nonlinear parametric models

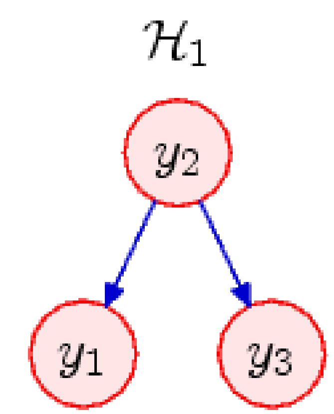
- Easy to use
  - predictions correspond to model with infinite number of parameters
- Equally good, or better, on a large range of datasets
- GPs have many standard regression methods as special cases
  - Radial basis functions
  - Splines
  - Feed-forward neural networks with one hidden layer

- Problems:
  - Ill-conditioned
  - N^3 complexity is bad news for N>1000
    - approximate methods

Gaussian Quiz solutions

$$\begin{aligned} y_2 &= \nu_2 \\ y_1 &= w_1 y_2 + \nu_1 \\ y_3 &= w_3 y_2 + \nu_3, \end{aligned}$$

$$\mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{12} & K_{22} & K_{23} \\ K_{13} & K_{23} & K_{33} \end{bmatrix} = \begin{bmatrix} w_1^2\sigma_2^2 + \sigma_1^2 & w_1\sigma_2^2 & w_1 w_3 \sigma_2^2 \\ & \sigma_2^2 & w_3\sigma_2^2 \\ & & w_3^2\sigma_2^2 + \sigma_3^2 \end{bmatrix}$$



$$P(y_1, y_2, y_3 \mid \mathcal{H}_1) = P(y_2) P(y_1 \mid y_2) P(y_3 \mid y_2) \tag{16}$$

$$= \frac{1}{Z_2} \exp\left(-\frac{y_2^2}{2\sigma_2^2}\right) \frac{1}{Z_1} \exp\left(-\frac{(y_1 - w_1 y_2)^2}{2\sigma_1^2}\right) \frac{1}{Z_3} \exp\left(-\frac{(y_3 - w_3 y_2)^2}{2\sigma_3^2}\right) \tag{17}$$

We can now collect all the terms in $y_i y_j$.

$$\begin{aligned} P(y_1, y_2, y_3) &= \frac{1}{Z'} \exp\left(-\frac{y_2^2}{2\sigma_2^2} - \frac{(y_1 - w_1 y_2)^2}{2\sigma_1^2} - \frac{(y_3 - w_3 y_2)^2}{2\sigma_3^2}\right) \\ &= \frac{1}{Z'} \exp\left(-y_2^2\left[\frac{1}{2\sigma_2^2} + \frac{w_1^2}{2\sigma_1^2} + \frac{w_3^2}{2\sigma_3^2}\right] - y_1^2 \frac{1}{2\sigma_1^2} + 2 y_1 y_2 \frac{w_1}{2\sigma_1^2} - y_3^2 \frac{1}{2\sigma_3^2} + 2 y_3 y_2 \frac{w_3}{2\sigma_3^2}\right) \\ &= \frac{1}{Z'} \exp\left(-\frac{1}{2} \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1^2} & -\frac{w_1}{\sigma_1^2} & 0 \\ -\frac{w_1}{\sigma_1^2} & \left[\frac{1}{\sigma_2^2} + \frac{w_1^2}{\sigma_1^2} + \frac{w_3^2}{\sigma_3^2}\right] & -\frac{w_3}{\sigma_3^2} \\ 0 & -\frac{w_3}{\sigma_3^2} & \frac{1}{\sigma_3^2} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}\right) \end{aligned}$$

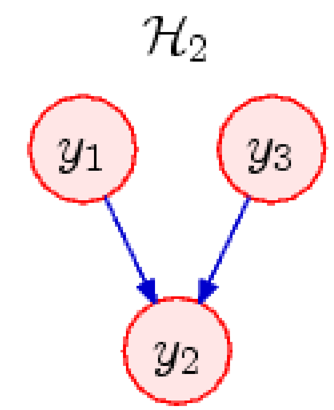
So the inverse covariance matrix is

$$\mathbf{K}^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & -\frac{w_1}{\sigma_1^2} & 0 \\ -\frac{w_1}{\sigma_1^2} & \left[\frac{1}{\sigma_2^2} + \frac{w_1^2}{\sigma_1^2} + \frac{w_3^2}{\sigma_3^2}\right] & -\frac{w_3}{\sigma_3^2} \\ 0 & -\frac{w_3}{\sigma_3^2} & \frac{1}{\sigma_3^2} \end{bmatrix}$$

Gaussian Quiz solutions

$y_2 = w_1 y_1 + w_3 y_3 + \nu_2$

$$\mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{12} & K_{22} & K_{23} \\ K_{13} & K_{23} & K_{33} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & w_1\sigma_1^2 & 0 \\ & \sigma_2^2 + w_1^2\sigma_1^2 + w_3^2\sigma_3^2 & w_3\sigma_3^2 \\ & & \sigma_3^2 \end{bmatrix}$$



$$P(y_1, y_2, y_3 \mid \mathcal{H}_2) = P(y_1)P(y_3)P(y_2 \mid y_1, y_3) \quad (23)$$

$$= \frac{1}{Z_1}\exp\left(-\frac{y_1^2}{2\sigma_1^2}\right)\frac{1}{Z_3}\exp\left(-\frac{y_3^2}{2\sigma_3^2}\right)\frac{1}{Z_2}\exp\left(-\frac{(y_2 - w_1y_1 - w_3y_3)^2}{2\sigma_2^2}\right) \quad (24)$$

We collect all the terms in $y_i y_j$.

$$P(y_1, y_2, y_3) = \frac{1}{Z'}\exp\left(-\frac{y_1^2}{2\sigma_1^2} - \frac{y_3^2}{2\sigma_3^2} - \frac{(y_2 - w_1y_1 - w_3y_3)^2}{2\sigma_2^2}\right)$$

$$= \frac{1}{Z'}\exp\left(-y_1^2\left[\frac{1}{2\sigma_1^2} + \frac{w_1^2}{2\sigma_2^2}\right] - y_2^2\frac{1}{2\sigma_2^2} + 2y_1y_2\frac{w_1}{2\sigma_1^2} - y_3^2\left[\frac{1}{2\sigma_3^2} + \frac{w_3^2}{2\sigma_2^2}\right] + 2y_3y_2\frac{w_3}{2\sigma_2^2} - 2y_3y_1\frac{w_1w_3}{2\sigma_2^2}\right)$$

$$= \frac{1}{Z'}\exp\left(-\frac{1}{2}\begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}\begin{bmatrix} \left[\frac{1}{2\sigma_1^2} + \frac{w_1^2}{\sigma_2^2}\right] & -\frac{w_1}{\sigma_2^2} & +\frac{w_1w_3}{\sigma_2^2} \\ -\frac{w_1}{\sigma_2^2} & \frac{1}{\sigma_2^2} & -\frac{w_3}{\sigma_2^2} \\ +\frac{w_1w_3}{\sigma_2^2} & -\frac{w_3}{\sigma_2^2} & \left[\frac{1}{2\sigma_3^2} + \frac{w_3^2}{\sigma_2^2}\right] \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}\right)$$

So the inverse covariance matrix is

$$\mathbf{K}^{-1} = \begin{bmatrix} \left[\frac{1}{2\sigma_1^2} + \frac{w_1^2}{\sigma_2^2}\right] & -\frac{w_1}{\sigma_2^2} & +\frac{w_1w_3}{\sigma_2^2} \\ -\frac{w_1}{\sigma_2^2} & \frac{1}{\sigma_2^2} & -\frac{w_3}{\sigma_2^2} \\ \end{bmatrix}$$

Gaussian Quiz solutions

Detailed, colourful solutions and comments are in `The Humble Gaussian distribution' (12 pages) - the top link on:

www.inference.phy.cam.ac.uk/mackay

So the inverse covariance matrix is

$$\mathbf{K}^{-1} = \begin{bmatrix} \left[\frac{1}{2\sigma_1^2} + \frac{w_1^2}{\sigma_2^2}\right] & -\frac{w_1}{\sigma_2^2} & +\frac{w_1 w_3}{\sigma_2^2} \\ -\frac{w_1}{\sigma_2^2} & \frac{1}{\sigma_2^2} & -\frac{w_3}{\sigma_2^2} \\ +\frac{w_1 w_3}{\sigma_2^2} & -\frac{w_3}{\sigma_2^2} & \left[\frac{1}{2\sigma_3^2} + \frac{w_3^2}{\sigma_2^2}\right] \end{bmatrix}$$

Presented in MagicPoint

A Free Presentation Graphic Tool

MagicPoint

- http://www.mew.org/mgp/