Introduction to Gaussian Processes

Neil D. Lawrence

MLSS, Tübingen
16-18th July 2015

Outline

Gaussian Processes

GP Non-Gaussian

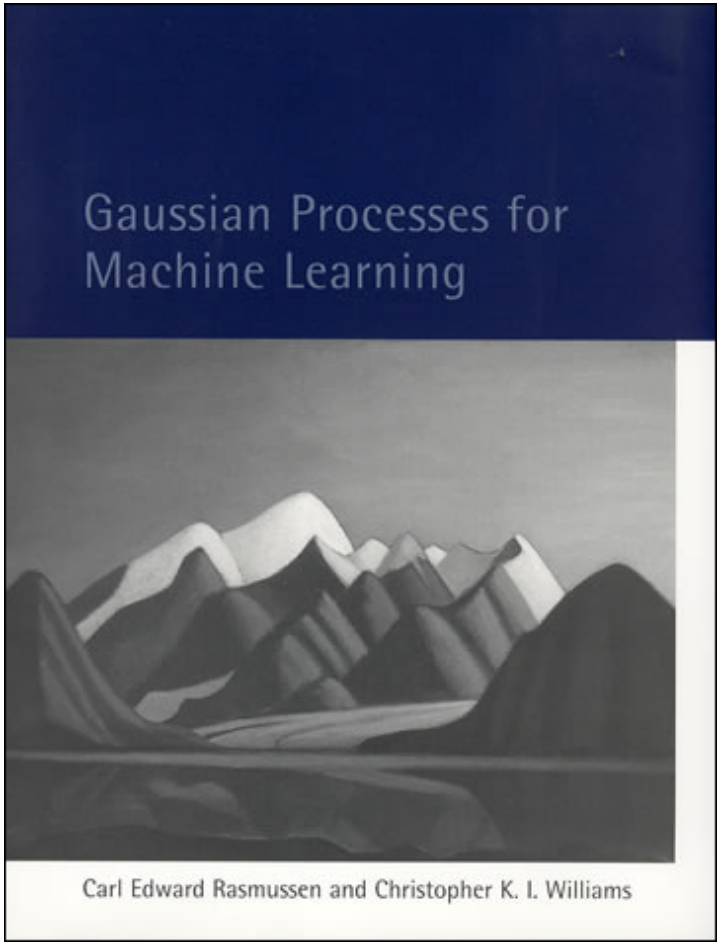
GP Limitations

Kalman Filter

Dimensionality Reduction

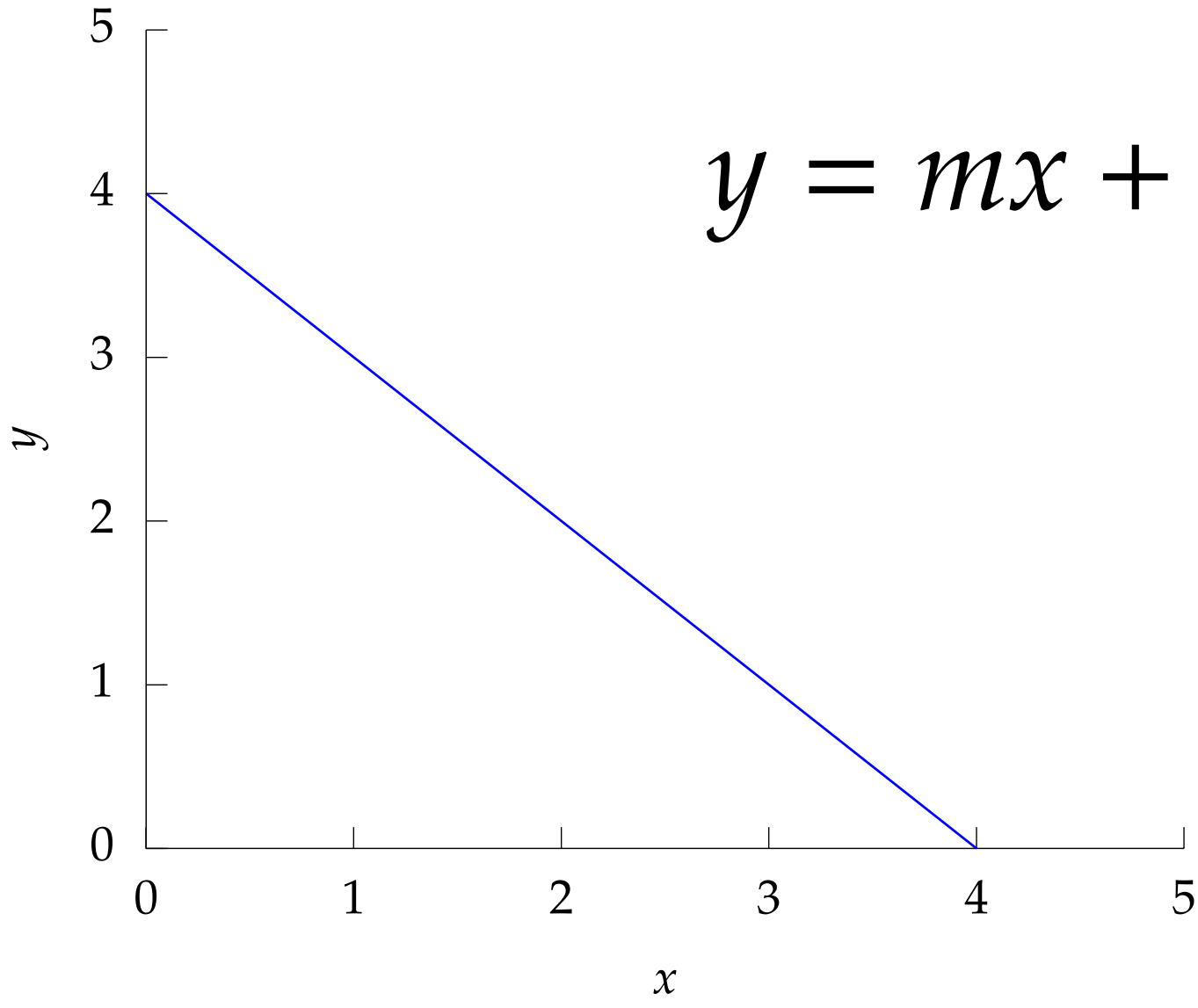
Outline

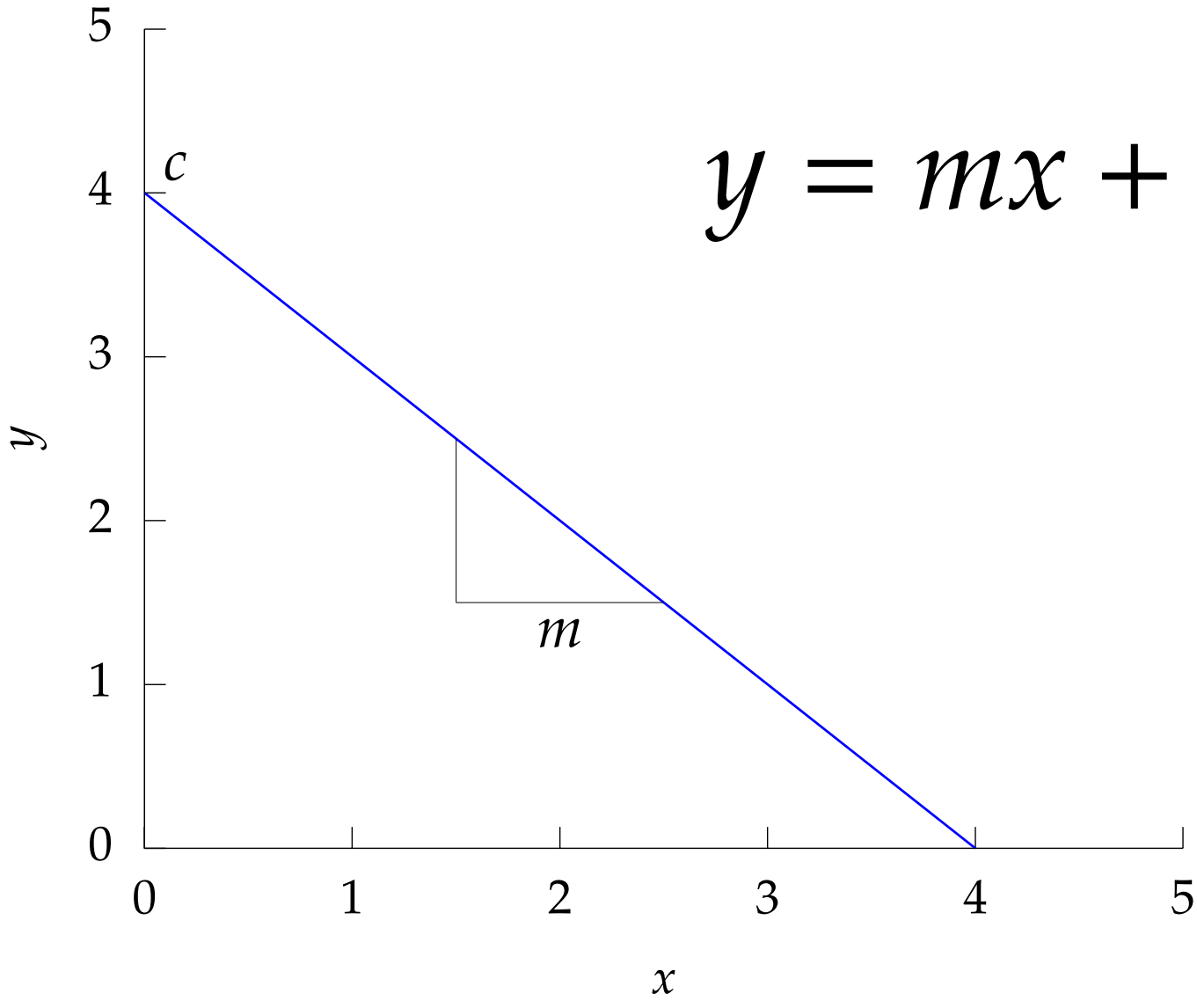
Book

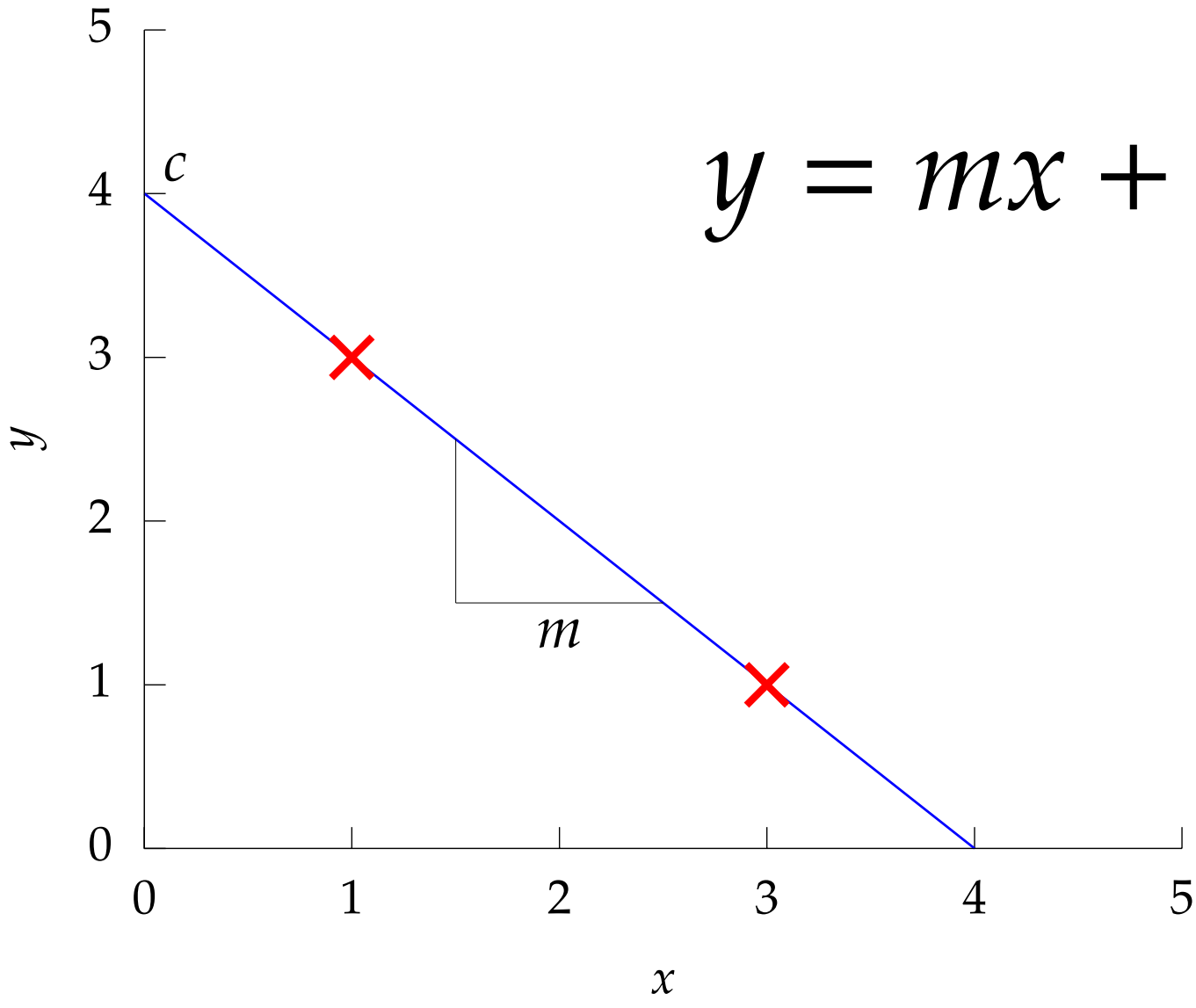


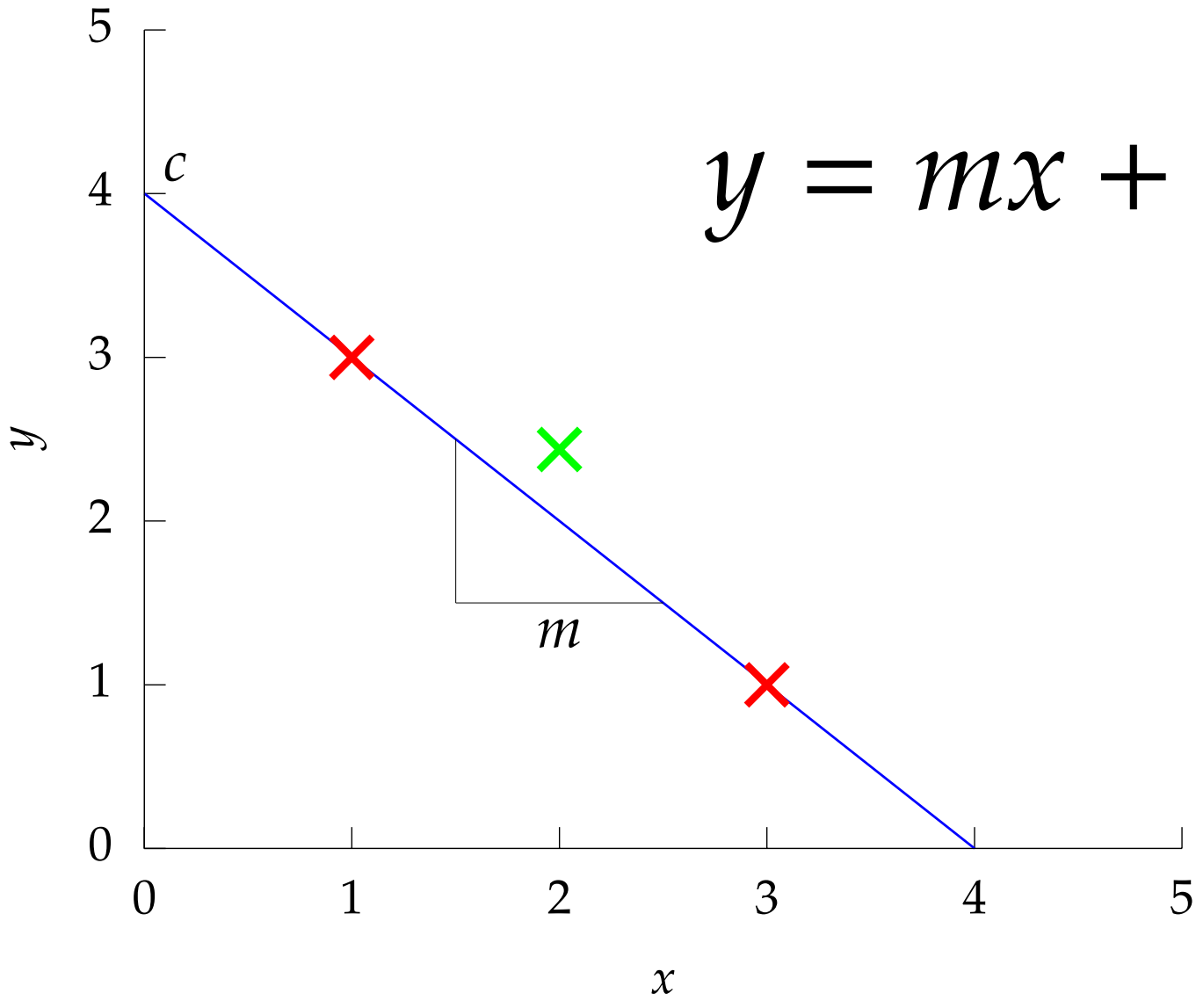
Rasmussen and Williams (2006)

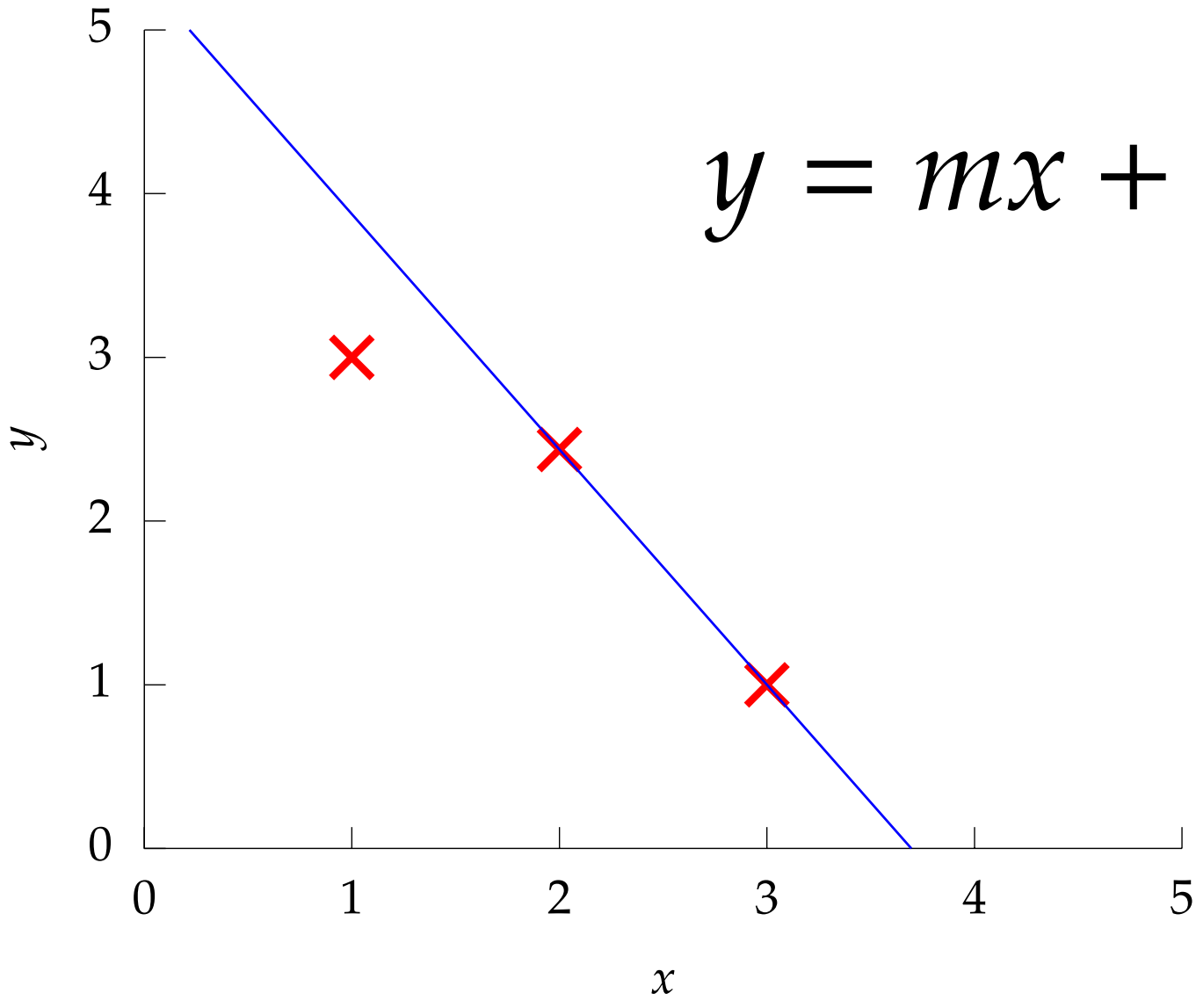
$$y = mx + c$$

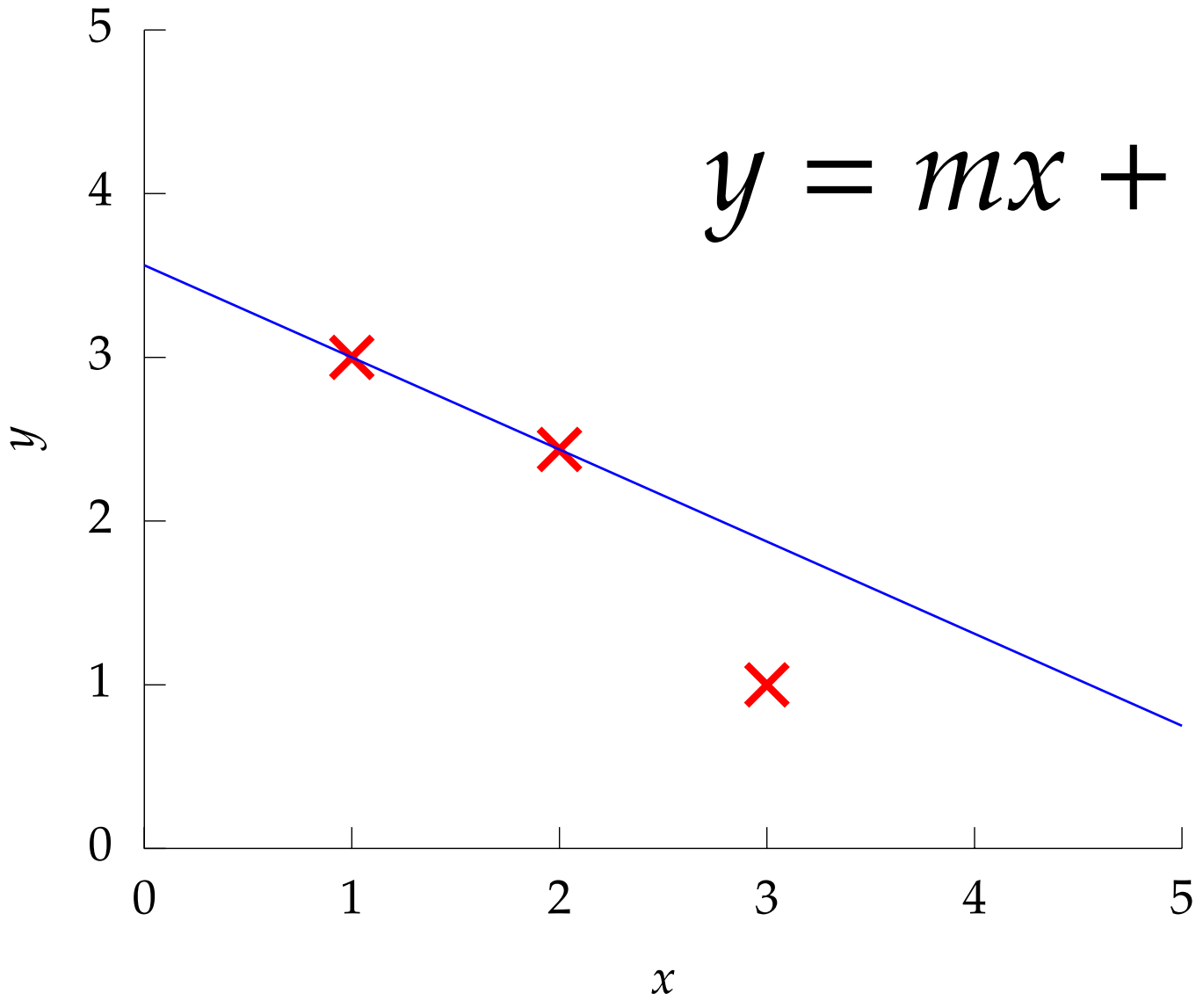

$y = mx + c$
5
4
3
2
1
0
y
0
1
2
3
4
5
x

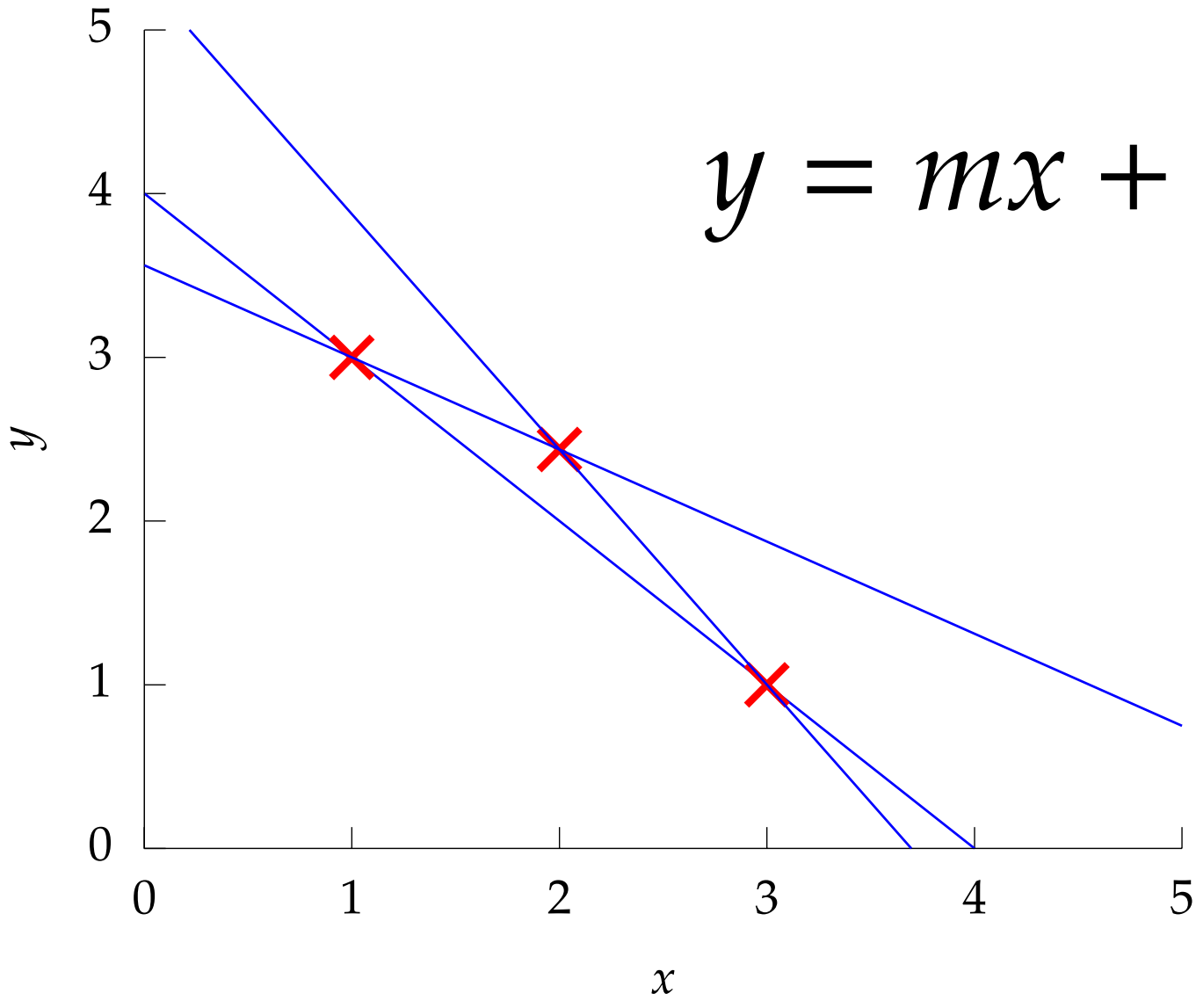

$y = mx + c$
c
m
5
4
3
2
1
0
0
1
2
3
4
5
y
x


$y = mx + c$
c
m
y
x
0
1
2
3
4
5


$y = mx + c$
c
m
y
x
0
1
2
3
4
5


$y = mx + c$
5
4
3
2
1
0
0
1
2
3
4
5
y
x


$y = mx + c$
5
4
3
2
1
0
0
1
2
3
4
5
y
x


$y = mx + c$
5
4
3
2
1
0
0
1
2
3
4
5
y
x

point 1: $x = 1, y = 3$

$$3 = m + c$$

point 2: $x = 3, y = 1$

$$1 = 3m + c$$

point 3: $x = 2, y = 2.5$

$$2.5 = 2m + c$$

riens. L'opinion contraire est une illusion de l'esprit qui, perdant de vue les raisons fugitives du choix de la volonté dans les choses indifférentes, se persuade qu'elle s'est déterminée d'elle-même et sans motifs.

Nous devons donc envisager l'état présent de l'univers, comme l'effet de son état antérieur, et comme la cause de celui qui va suivre. Une intelligence qui, pour un instant donné, connaîtrait toutes les forces dont la nature est animée, et la situation respective des êtres qui la composent, si d'ailleurs elle était assez vaste pour soumettre ces données à l'analyse, embrasserait dans la même formule les mouvemens des plus grands corps de l'univers et ceux du plus léger atome : rien ne serait incertain pour elle, et l'avenir comme le passé, serait présent à ses yeux. L'esprit humain offre, dans la perfection qu'il a su donner à l'Astronomie, une faible esquisse de cette intelligence. Ses découvertes en Mécanique et en Géométrie, jointes à celle de la pesanteur universelle, l'ont mis à portée de comprendre dans les mêmes expressions analytiques, les états passés et futurs du système du monde. En appliquant la même méthode à quelques autres objets de ses connaissances, il est parvenu à ramener à des lois générales, les phénomènes observés, et à prévoir ceux que des circonstances données doivent faire éclore. Tous ces efforts dans la recherche de la vérité, tendent à le rapprocher sans cesse de l'intelligence que nous venons de concevoir, mais dont il restera toujours infiniment éloigné. Cette tendance propre à l'espèce humaine, est ce qui la rend supérieure aux animaux; et ses progrès en ce genre, distinguent les nations et les siècles, et font leur véritable gloire.

Rappelons-nous qu'autrefois, et à une époque qui

other, we say that its choice is an effect without a cause. It is then, says Leibnitz, the blind chance of the Epicureans. The contrary opinion is an illusion of the mind, which, losing sight of the evasive reasons of the choice of the will in indifferent things, believes that choice is determined of itself and without motives.

We ought then to regard the present state of the universe as the effect of its anterior state and as the cause of the one which is to follow. Given for one instant an intelligence which could comprehend all the forces by which nature is animated and the respective situation of the beings who compose it—an intelligence sufficiently vast to submit these data to analysis—it would embrace in the same formula the movements of the greatest bodies of the universe and those of the lightest atom; for it, nothing would be uncertain and the future, as the past, would be present to its eyes. The human mind offers, in the perfection which it has been able to give to astronomy, a feeble idea of this intelligence. Its discoveries in mechanics and geometry, added to that of universal gravity, have enabled it to comprehend in the same analytical expressions the past and future states of the system of the world. Applying the same method to some other objects of its knowledge, it has succeeded in referring to general laws observed phenomena and in foreseeing those which given circumstances ought to produce. All these efforts in the search for truth tend to lead it back continually to the vast intelligence which we have just mentioned, but from which it will always remain infinitely removed. This tendency, peculiar to the human race, is that which renders it superior to animals; and their progress

height: "The day will come when, by study pursued through several ages, the things now concealed will appear with evidence; and posterity will be astonished that truths so clear had escaped us." Clairaut then undertook to submit to analysis the perturbations which the comet had experienced by the action of the two great planets, Jupiter and Saturn; after immense calculations he fixed its next passage at the perihelion toward the beginning of April, 1759, which was actually verified by observation. The regularity which astronomy shows us in the movements of the comets doubtless exists also in all phenomena.

The curve described by a simple molecule of air or vapor is regulated in a manner just as certain as the planetary orbits; the only difference between them is that which comes from our ignorance.

Probability is relative, in part to this ignorance, in part to our knowledge. We know that of three or a greater number of events a single one ought to occur; but nothing induces us to believe that one of them will occur rather than the others. In this state of indecision it is impossible for us to announce their occurrence with certainty. It is, however, probable that one of these events, chosen at will, will not occur because we see several cases equally possible which exclude its occurrence, while only a single one favors it.

The theory of chance consists in reducing all the events of the same kind to a certain number of cases equally possible, that is to say, to such as we may be equally undecided about in regard to their existence, and in determining the number of cases favorable to the event whose probability is sought. The ratio of

point 1: $x = 1, y = 3$

$$3 = m + c + \epsilon_1$$

point 2: $x = 3, y = 1$

$$1 = 3m + c + \epsilon_2$$

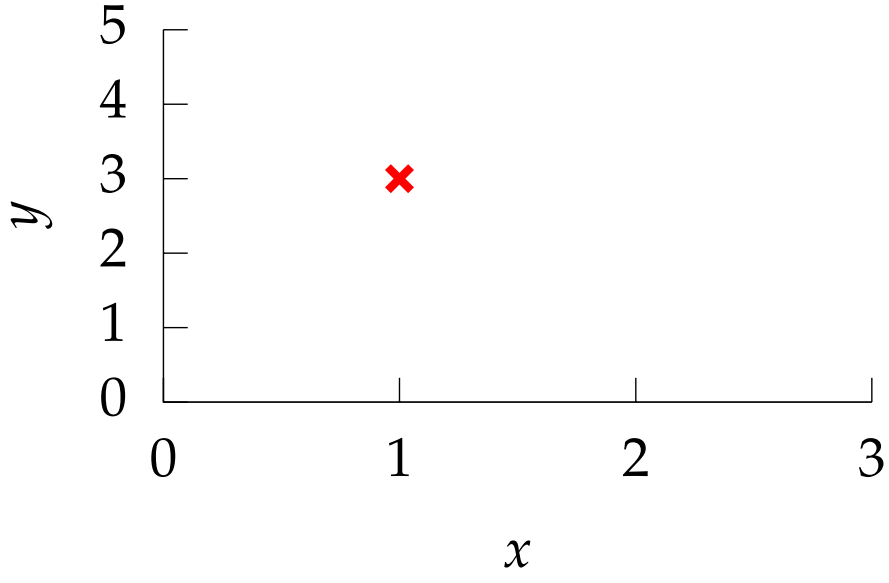
point 3: $x = 2, y = 2.5$

$$2.5 = 2m + c + \epsilon_3$$

Underdetermined System

What about two unknowns and *one* observation?

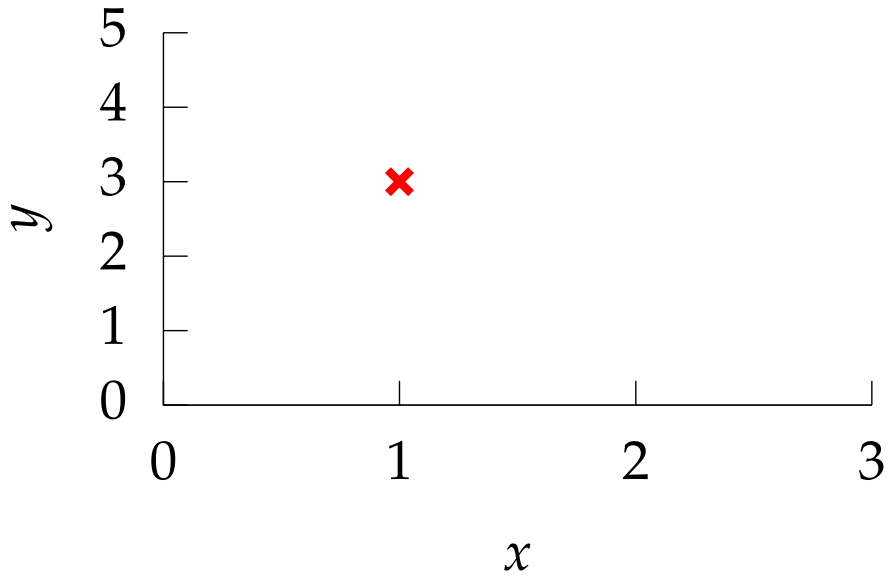
$$y_1 = mx_1 + c$$



Underdetermined System

Can compute m given c.

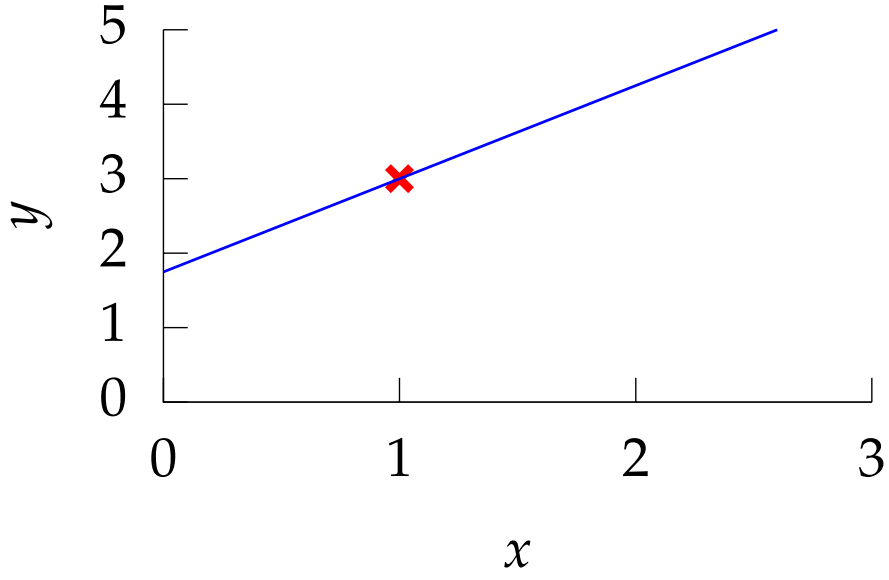
$$m = \frac{y_1 - c}{x}$$



Underdetermined System

Can compute m given c.

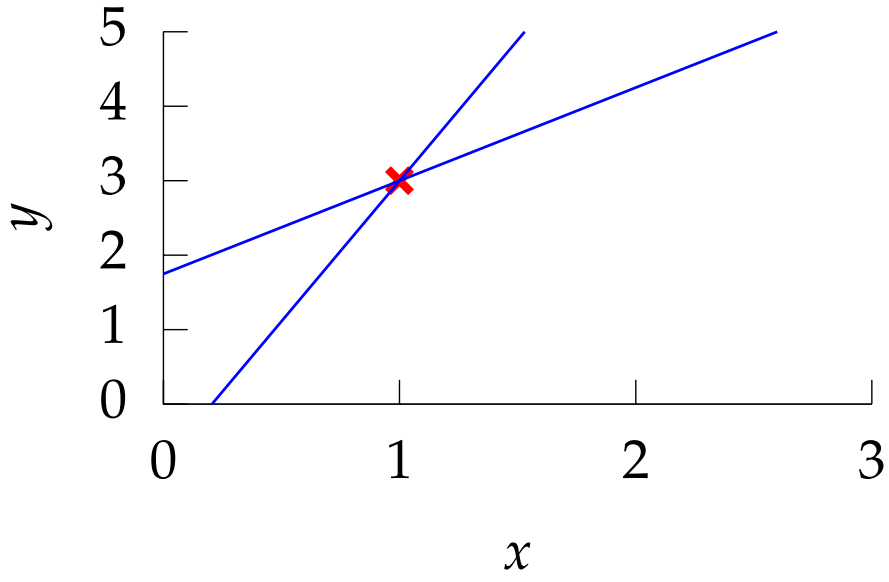
$$c = 1.75 \implies m = 1.25$$



Underdetermined System

Can compute m given c.

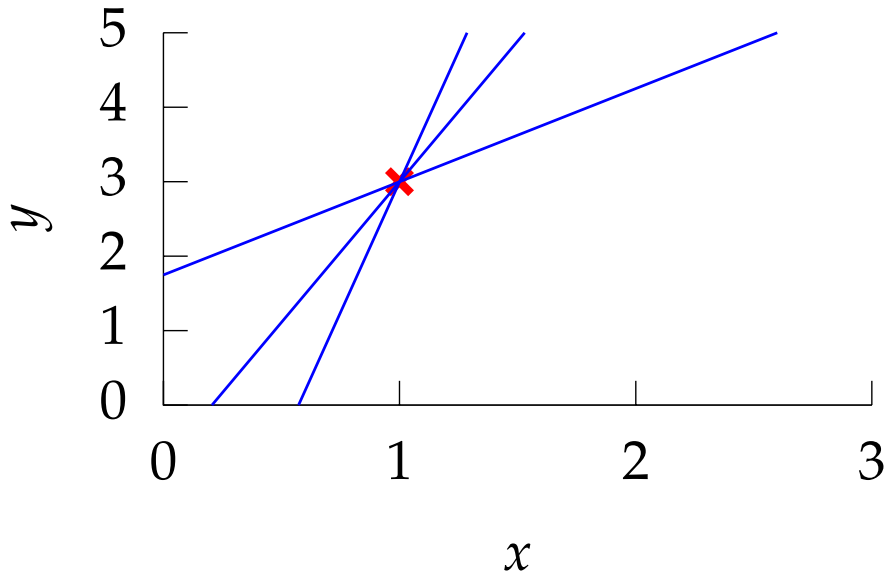
$$c = -0.777 \implies m = 3.78$$



Underdetermined System

Can compute m given c.

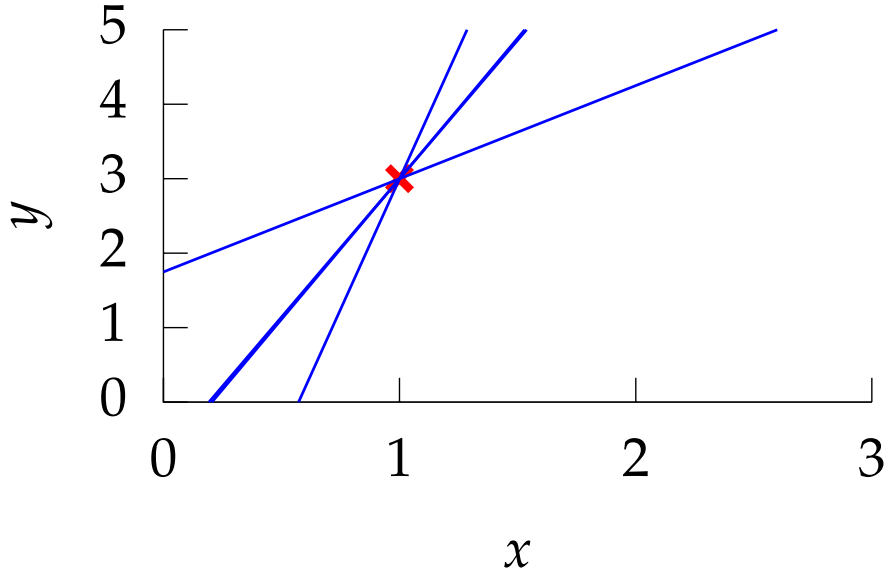
$$c = -4.01 \implies m = 7.01$$



Underdetermined System

Can compute m given c.

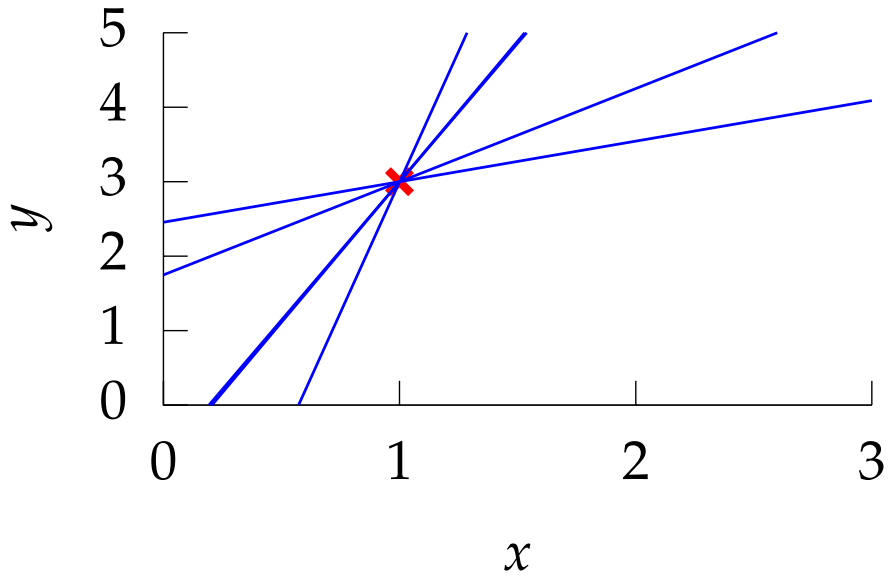
$$c = -0.718 \implies m = 3.72$$



Underdetermined System

Can compute m given c.

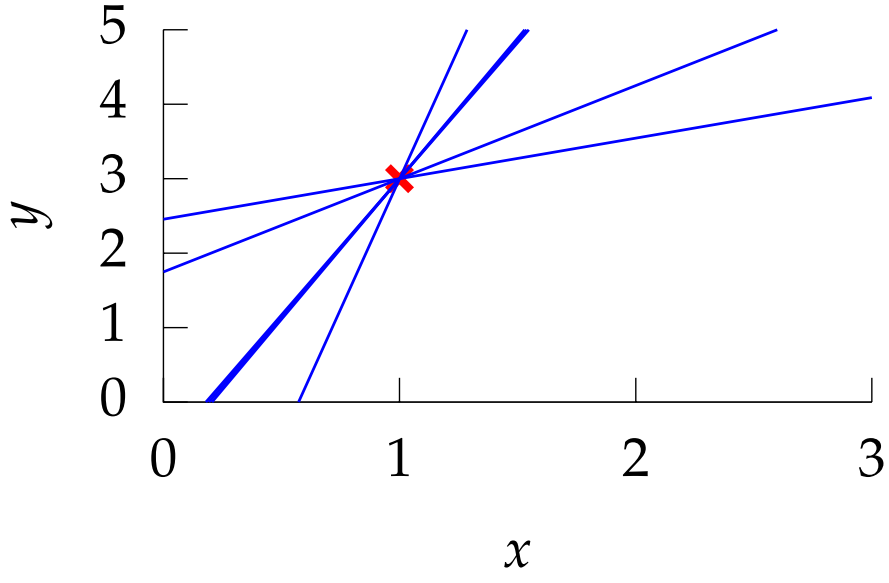
$$c = 2.45 \implies m = 0.545$$



Underdetermined System

Can compute m given c.

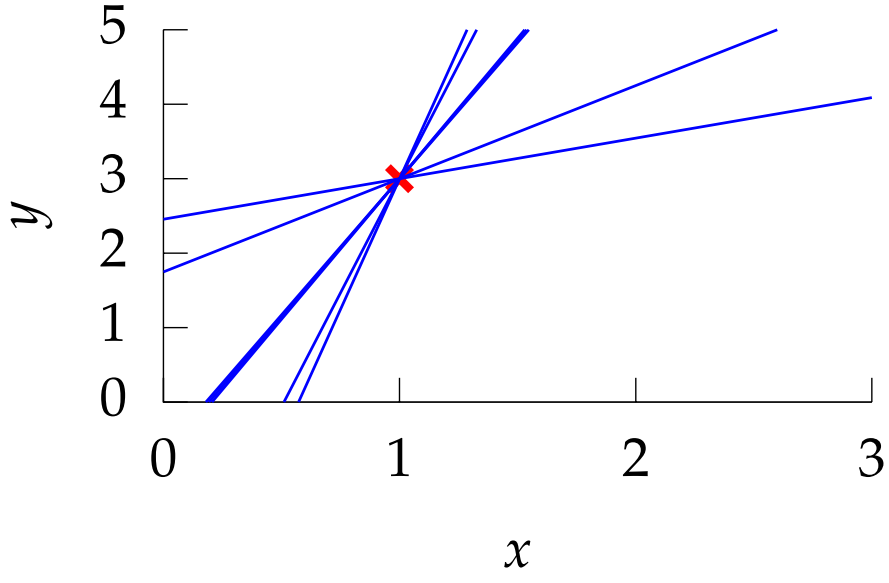
$$c = -0.657 \implies m = 3.66$$



Underdetermined System

Can compute m given c.

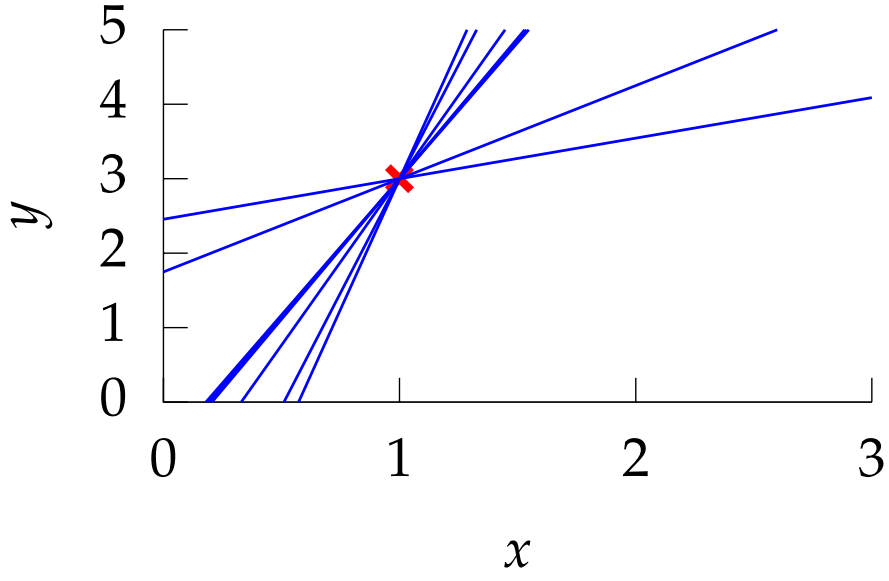
$$c = -3.13 \implies m = 6.13$$



Underdetermined System

Can compute m given c.

$$c = -1.47 \implies m = 4.47$$

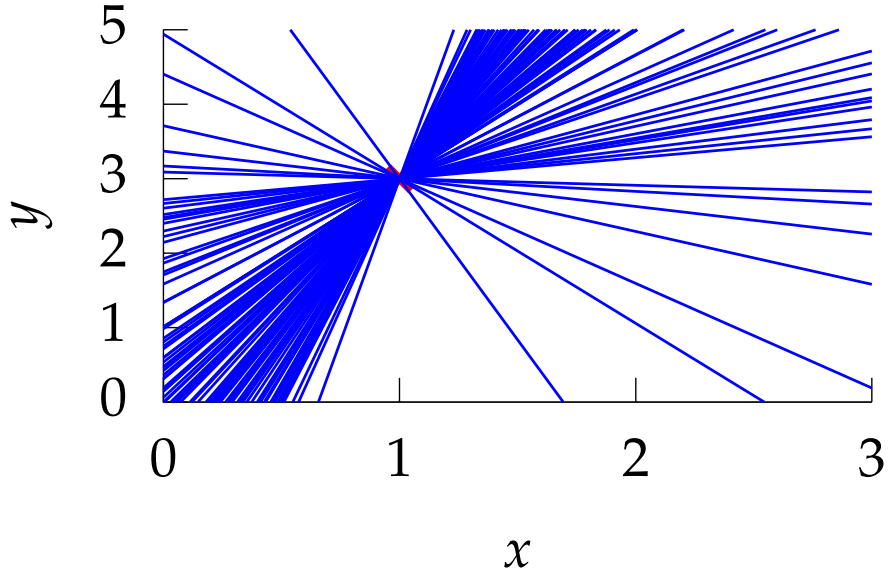


Underdetermined System

Can compute m given c.
Assume

$$c \sim \mathcal{N}(0, 4),$$

we find a distribution of solutions.



Gaussian Process

$$y_i(\mathbf{x}_i) = f(\mathbf{x}_i) + \epsilon_i$$

- Place a prior over the process as well as the noise.
- Leads to models that are not i.i.d.
- Contrast with classical model's objective function:

$$\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b))_+ + \lambda \mathbf{w}^\top \mathbf{w}$$

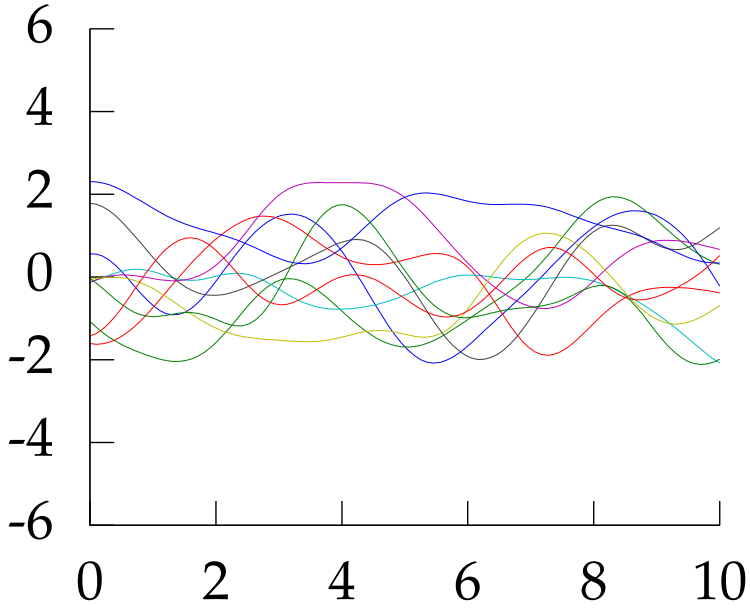
Model and Algorithm

- I'm keen on the idea of a conceptual separation model and algorithm.
- Model is how you encode the regularities of the universe.
- Algorithm is how you combine that model with data.

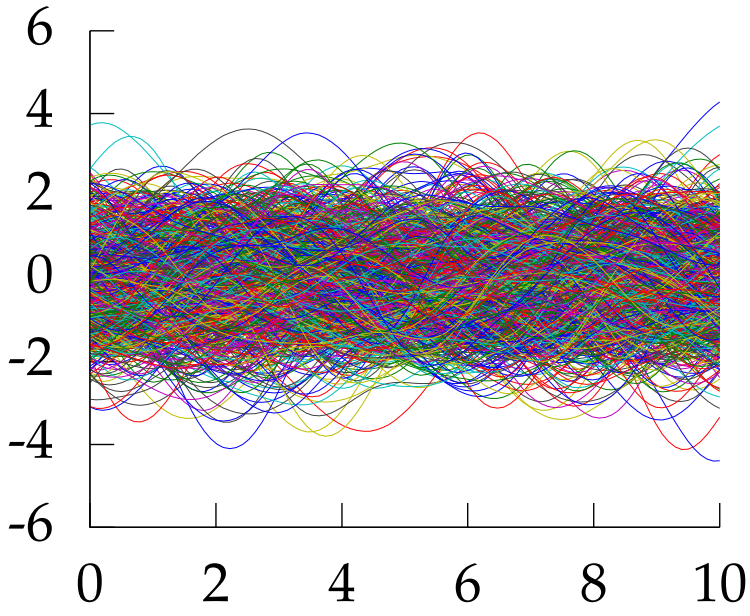
$$\text{data} + \text{model} \rightarrow \text{prediction}$$

- Of course often we are restricted in modeling choice due to lack of algorithms.

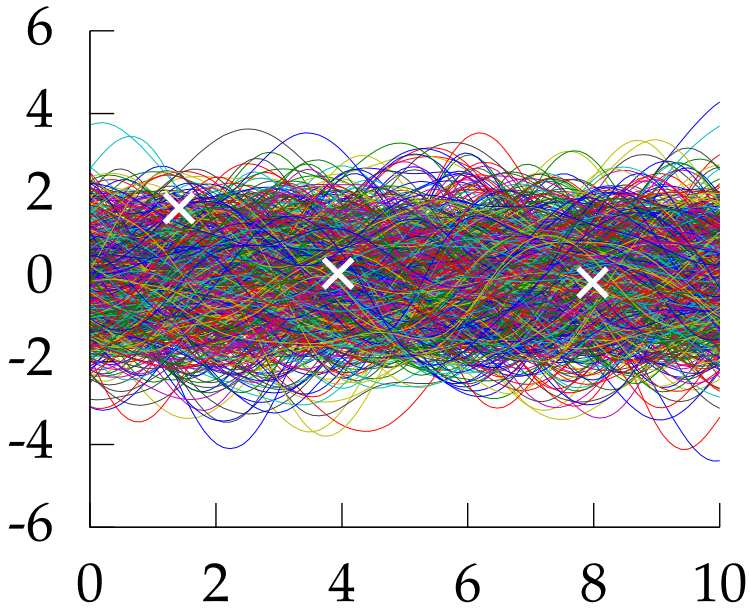
Gaussian Processes: Extremely Short Overview



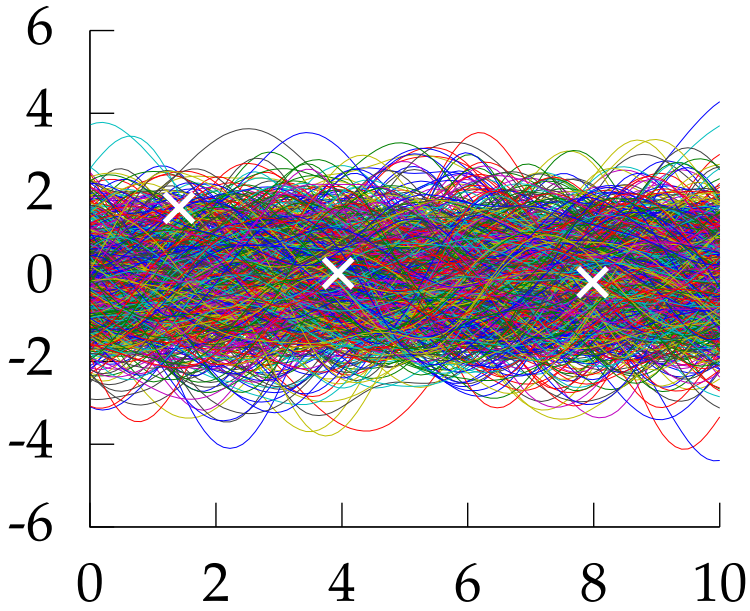
Gaussian Processes: Extremely Short Overview

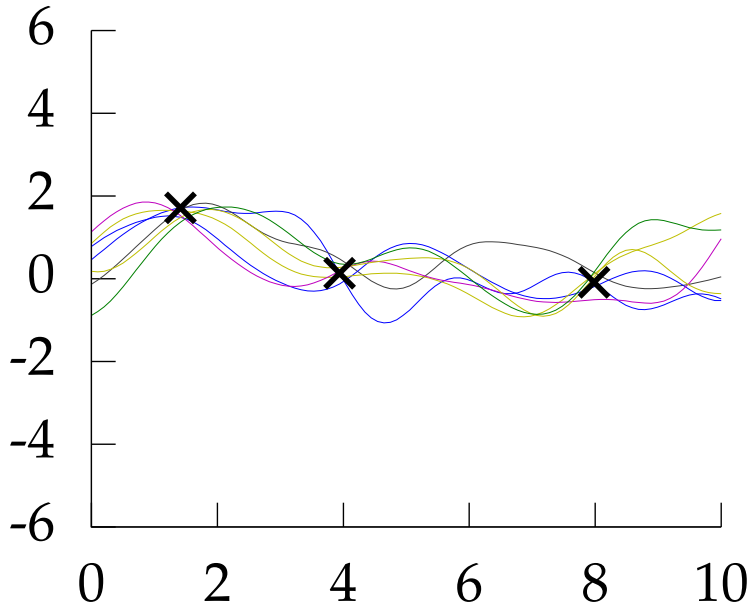


Gaussian Processes: Extremely Short Overview



Gaussian Processes: Extremely Short Overview



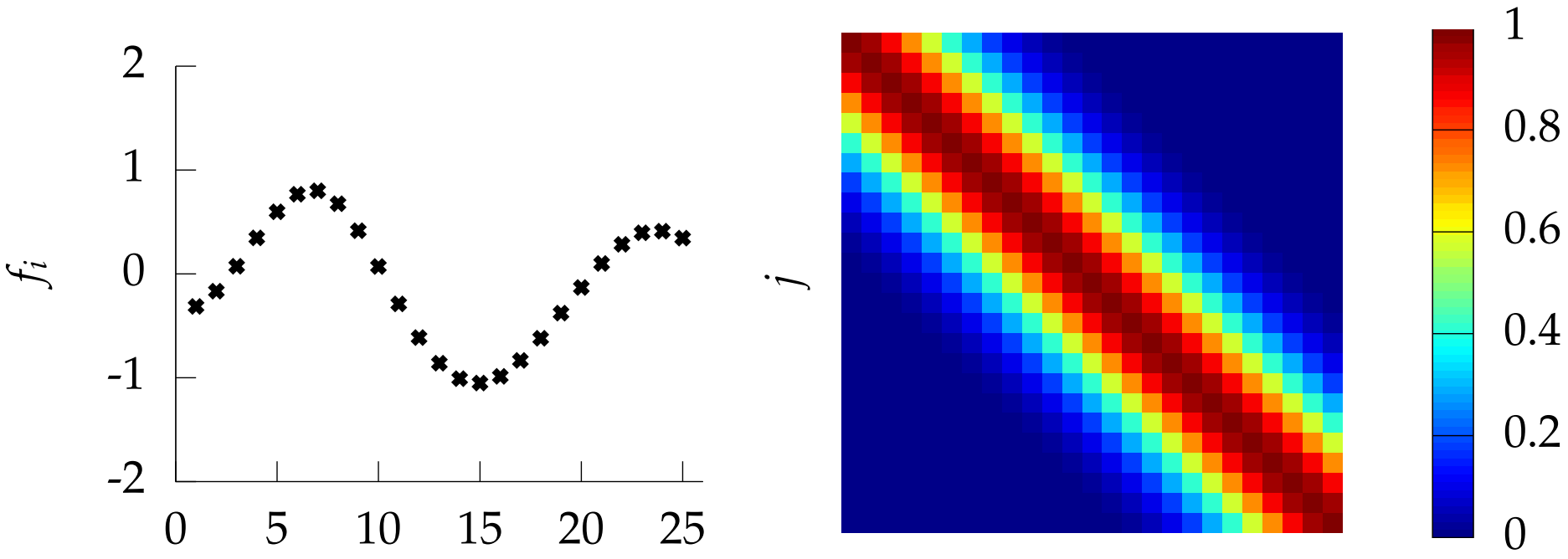


Sampling a Function

Multi-variate Gaussians

- We will consider a Gaussian with a particular structure of covariance matrix.
- Generate a single sample from this 25 dimensional Gaussian distribution, $\mathbf{f} = [f_1, f_2 \dots f_{25}]$.
- We will plot these points against their index.

Gaussian Distribution Sample

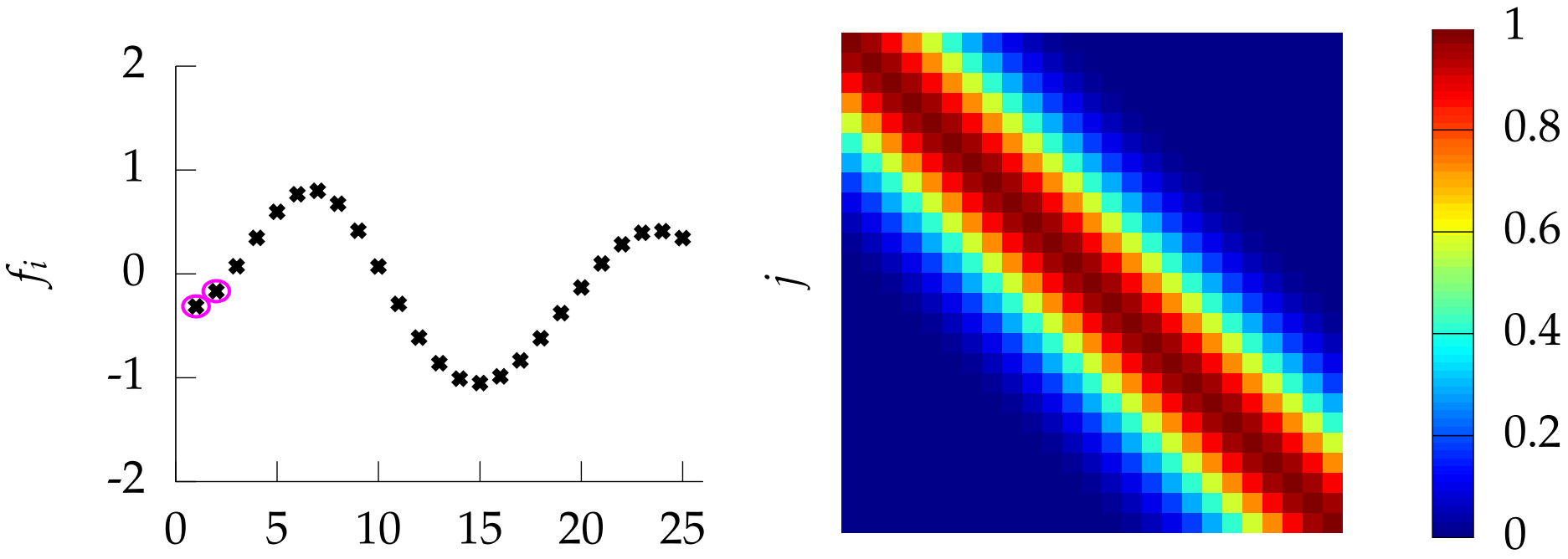


(a) A 25 dimensional correlated random variable (values ploted against index)

(b) colormap showing correlations between dimensions.

Figure: A sample from a 25 dimensional Gaussian distribution.

Gaussian Distribution Sample

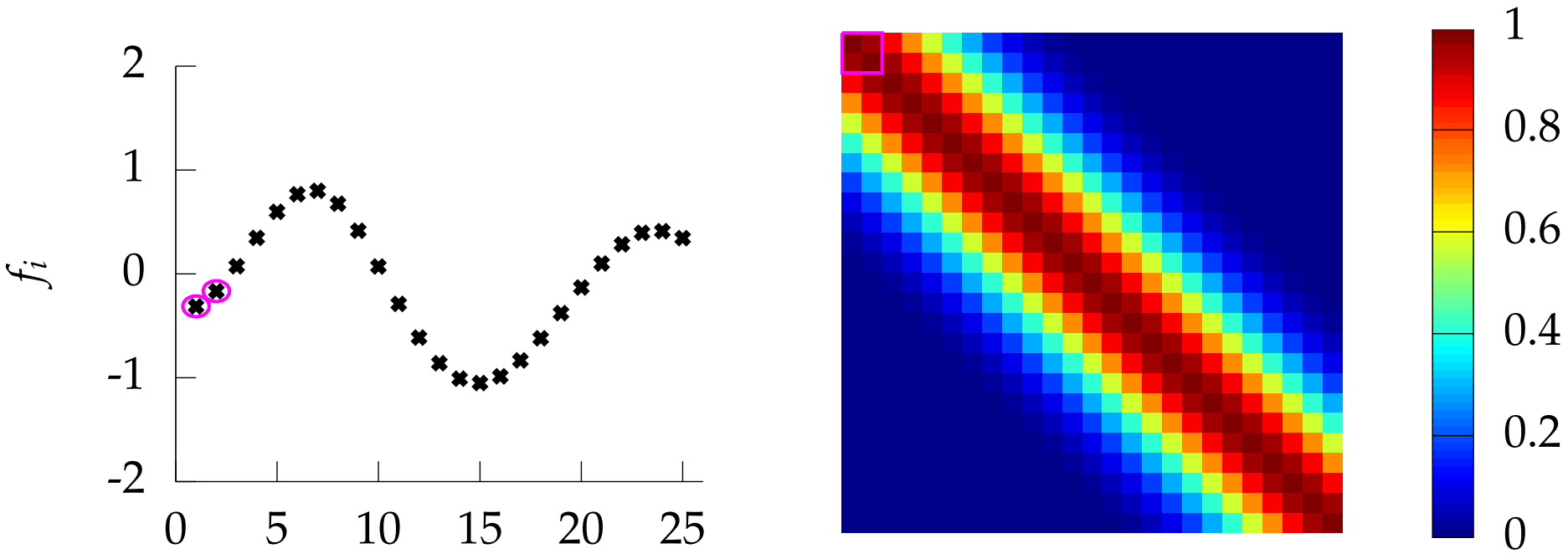


(a) A 25 dimensional correlated random variable (values ploted against index)

(b) colormap showing correlations between dimensions.

Figure: A sample from a 25 dimensional Gaussian distribution.

Gaussian Distribution Sample

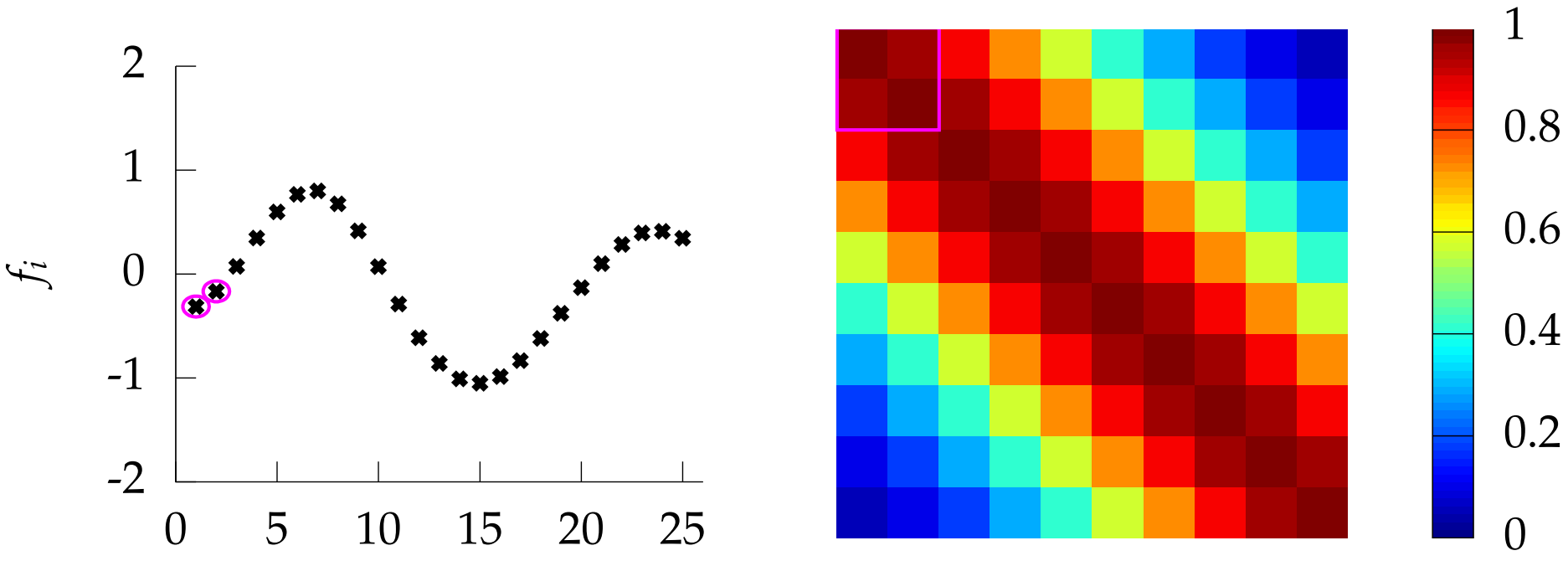


(a) A 25 dimensional correlated random variable (values ploted against index)

(b) colormap showing correlations between dimensions.

Figure: A sample from a 25 dimensional Gaussian distribution.

Gaussian Distribution Sample

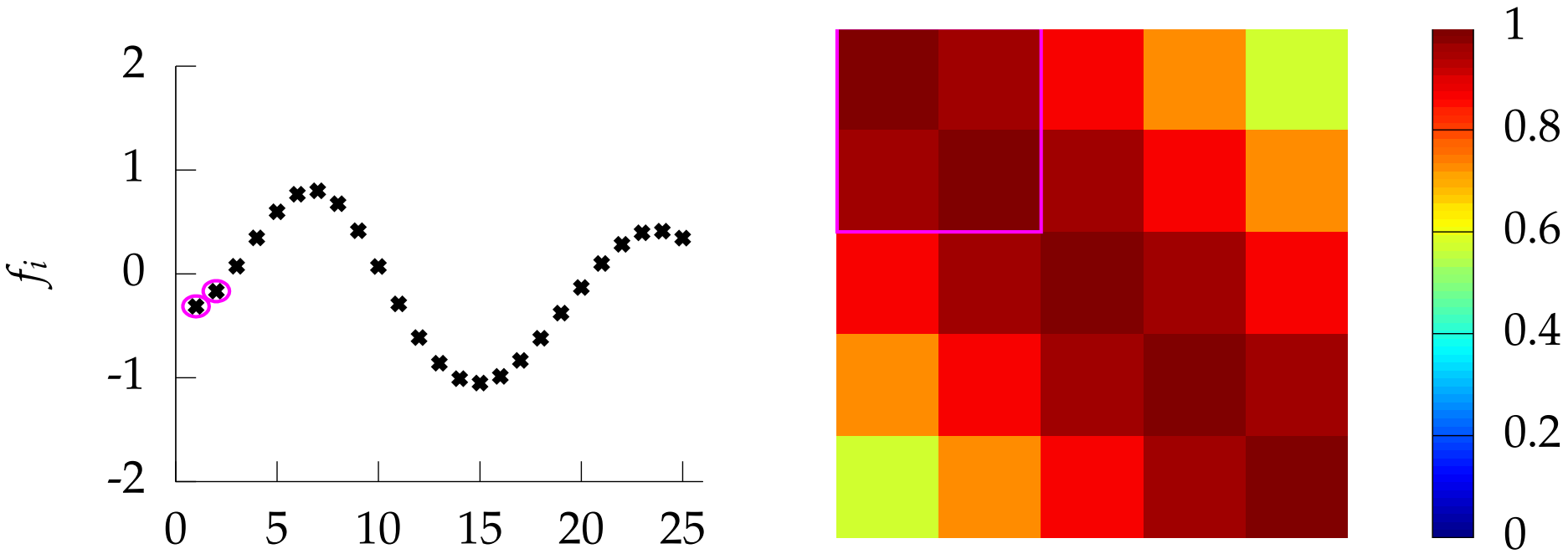


(a) A 25 dimensional correlated random variable (values ploted against index)

(b) colormap showing correlations between dimensions.

Figure: A sample from a 25 dimensional Gaussian distribution.

Gaussian Distribution Sample

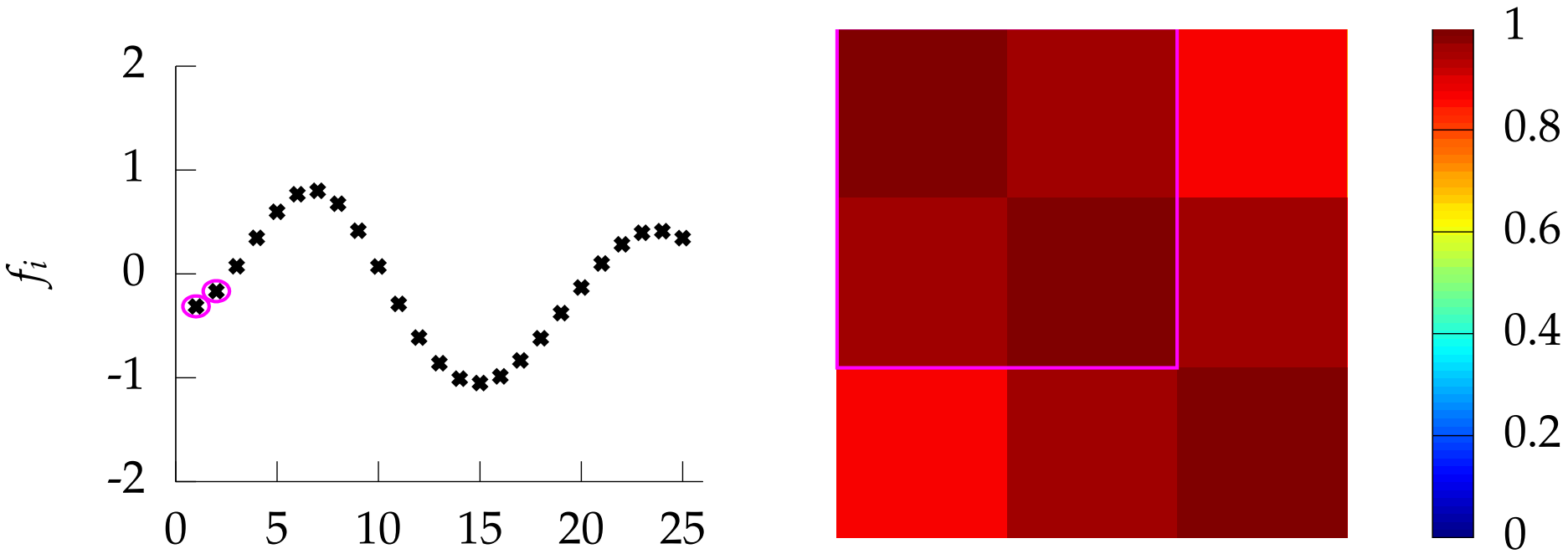


(a) A 25 dimensional correlated random variable (values ploted against index)

(b) colormap showing correlations between dimensions.

Figure: A sample from a 25 dimensional Gaussian distribution.

Gaussian Distribution Sample

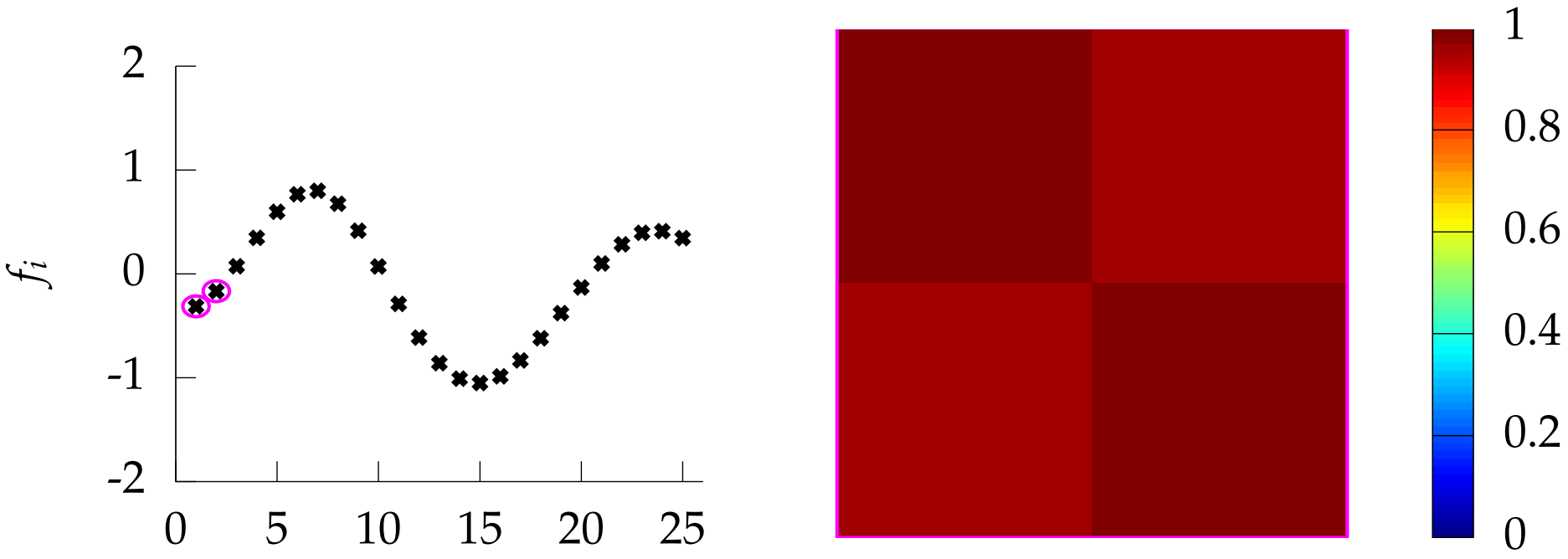


(a) A 25 dimensional correlated random variable (values ploted against index)

(b) colormap showing correlations between dimensions.

Figure: A sample from a 25 dimensional Gaussian distribution.

Gaussian Distribution Sample

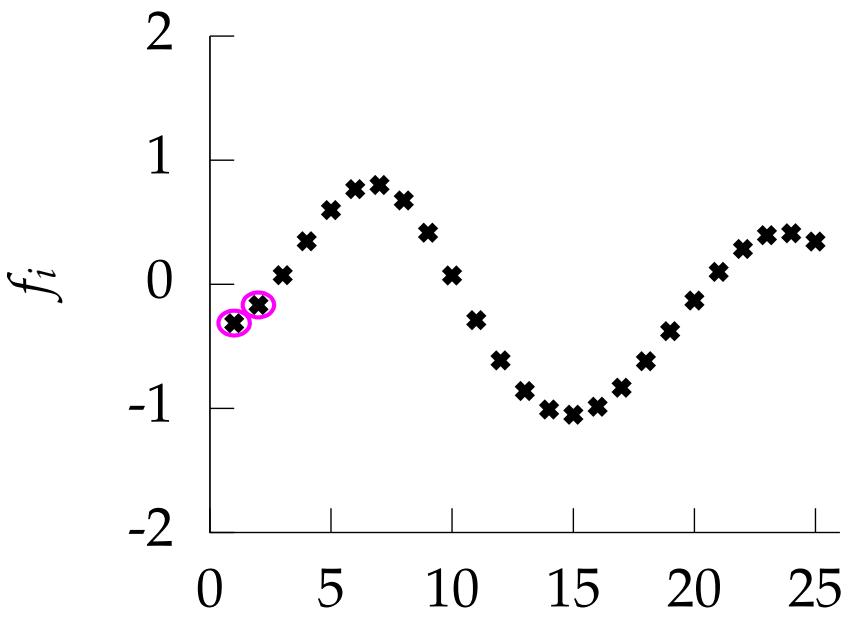


(a) A 25 dimensional correlated random variable (values ploted against index)

(b) colormap showing correlations between dimensions.

Figure: A sample from a 25 dimensional Gaussian distribution.

Gaussian Distribution Sample



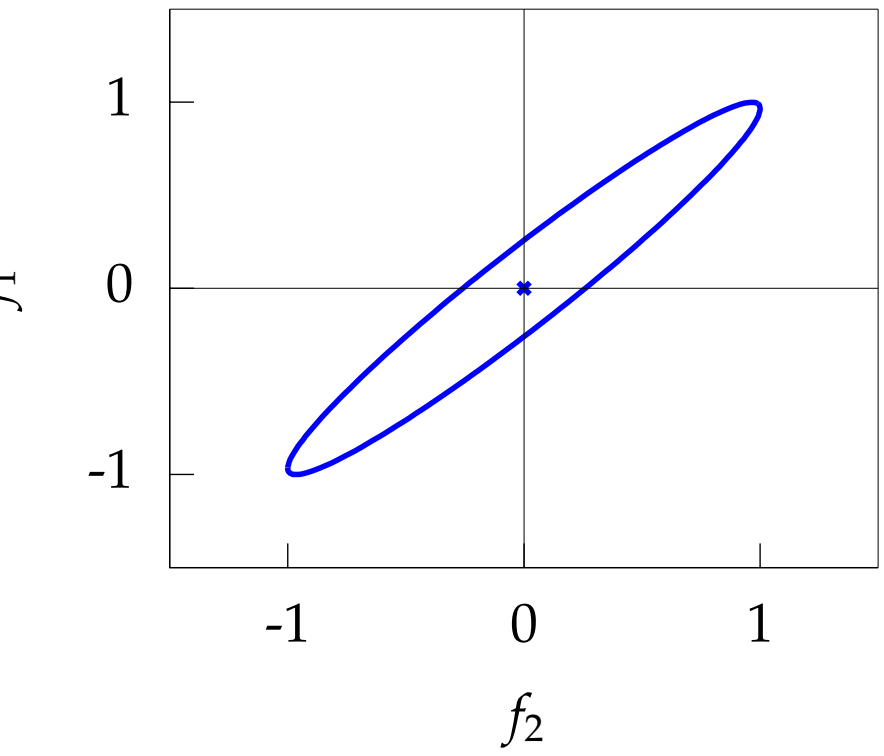
(a) A 25 dimensional correlated random variable (values ploted against index)

$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

(b) correlation between f_1 and f_2.

Figure: A sample from a 25 dimensional Gaussian distribution.

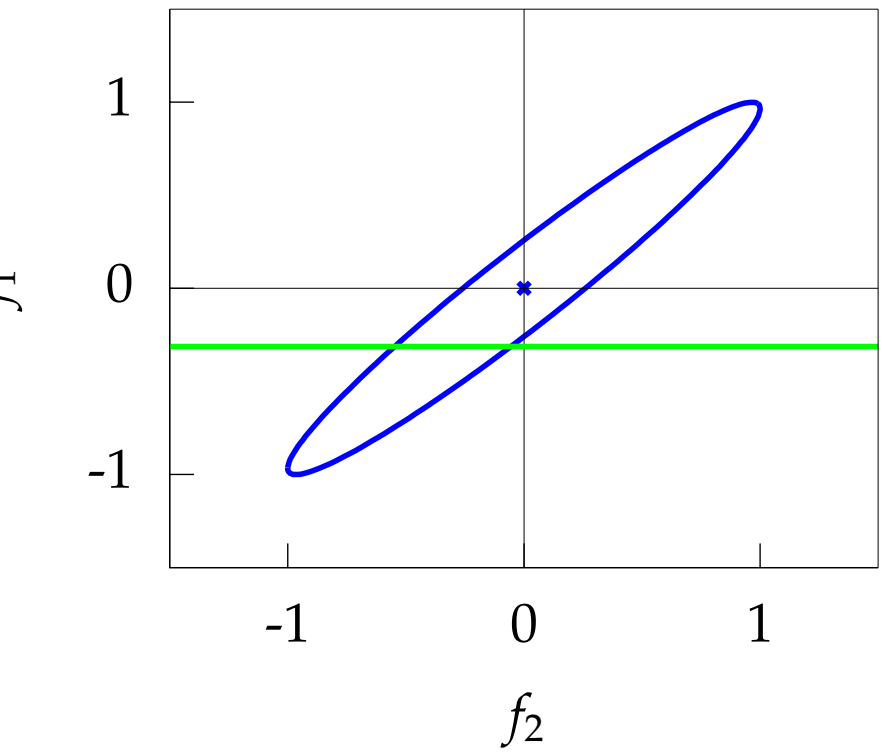
Prediction of f_2 from f_1



$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution, $p(f_1, f_2)$.

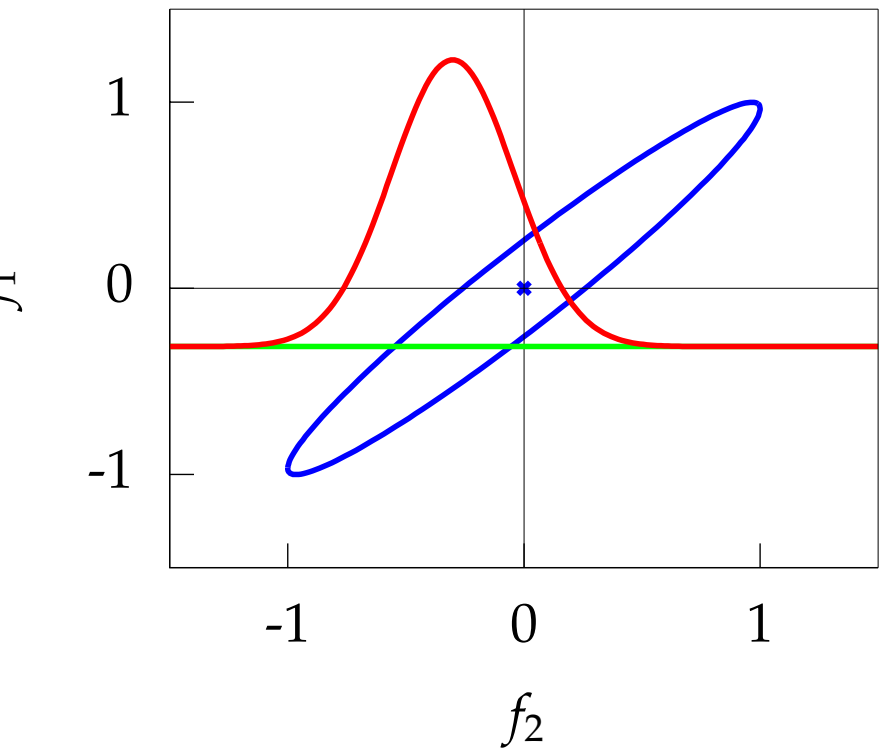
Prediction of f_2 from f_1



$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution, $p(f_1, f_2)$.
- We observe that $f_1 = -0.313$.

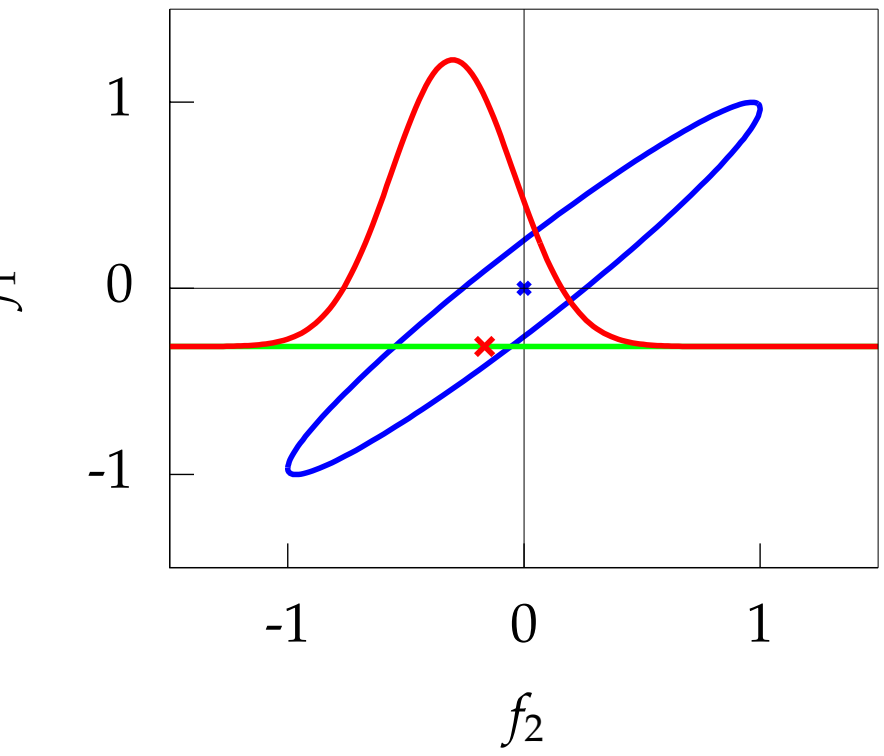
Prediction of f_2 from f_1



$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution, $p(f_1, f_2)$.
- We observe that $f_1 = -0.313$.
- Conditional density: $p(f_2|f_1 = -0.313)$.

Prediction of f_2 from f_1



$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution, $p(f_1, f_2)$.
- We observe that $f_1 = -0.313$.
- Conditional density: $p(f_2|f_1 = -0.313)$.

Prediction with Correlated Gaussians

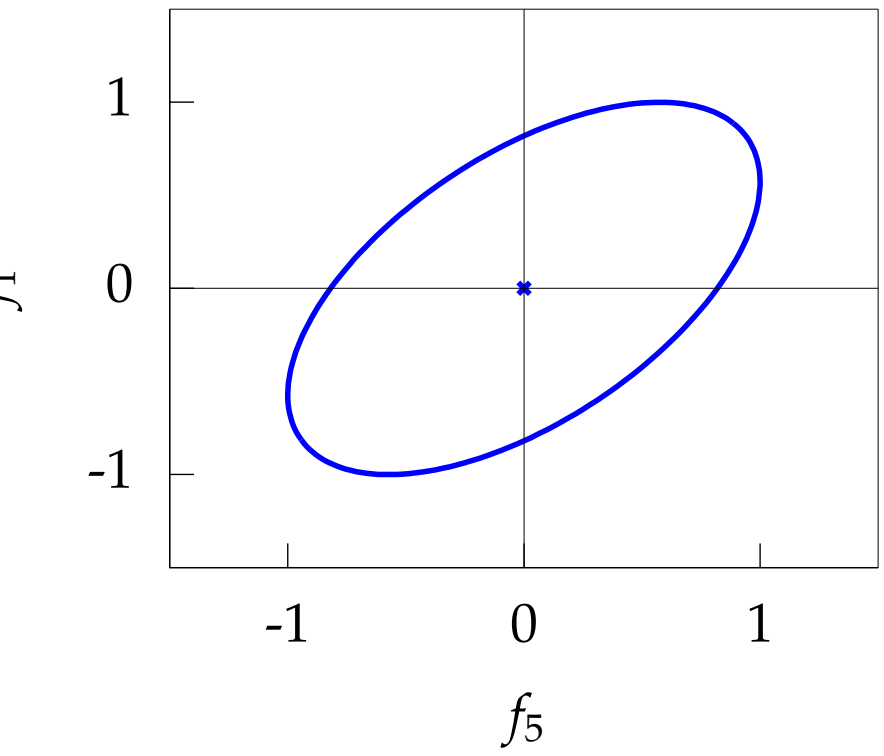
- Prediction of f_2 from f_1 requires *conditional density*.
- Conditional density is *also* Gaussian.

$$p(f_2|f_1) = \mathcal{N}\left(f_2|\frac{k_{1,2}}{k_{1,1}}f_1, k_{2,2} - \frac{k_{1,2}^2}{k_{1,1}}\right)$$

where covariance of joint density is given by

$$\mathbf{K} = \begin{bmatrix} k_{1,1} & k_{1,2} \\ k_{2,1} & k_{2,2} \end{bmatrix}$$

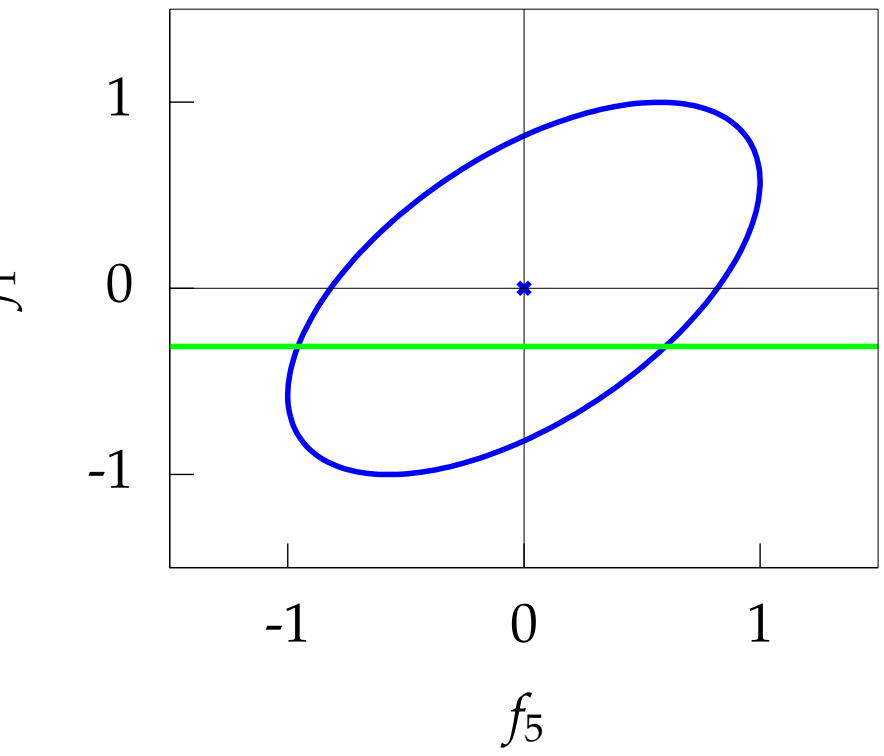
Prediction of f_5 from f_1



$$\begin{bmatrix} 1 & 0.57375 \\ 0.57375 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution, $p(f_1, f_5)$.

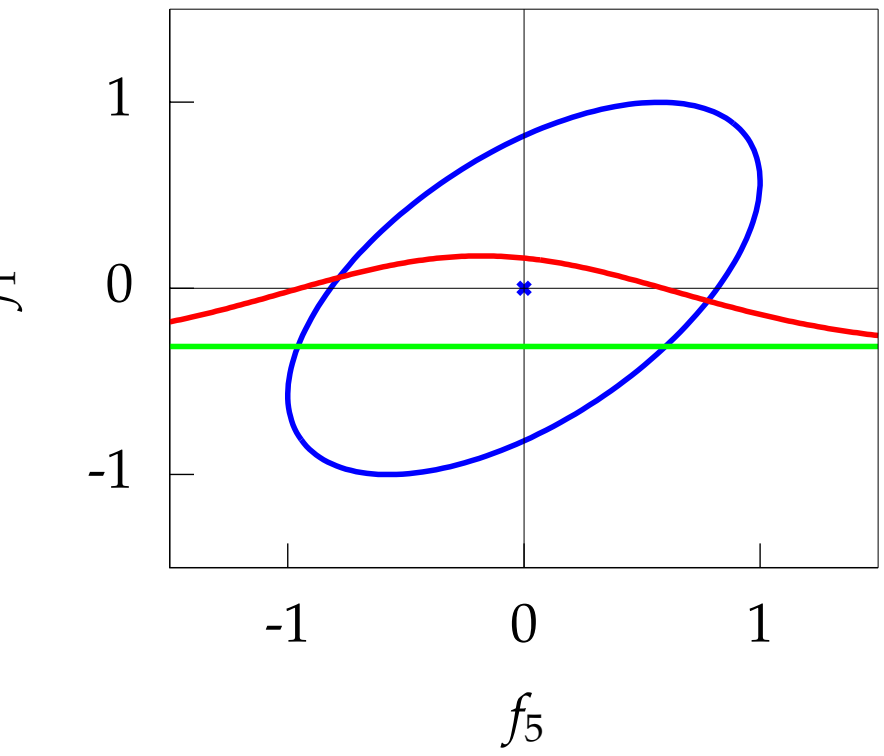
Prediction of f_5 from f_1



$$\begin{bmatrix} 1 & 0.57375 \\ 0.57375 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution, $p(f_1, f_5)$.
- We observe that $f_1 = -0.313$.

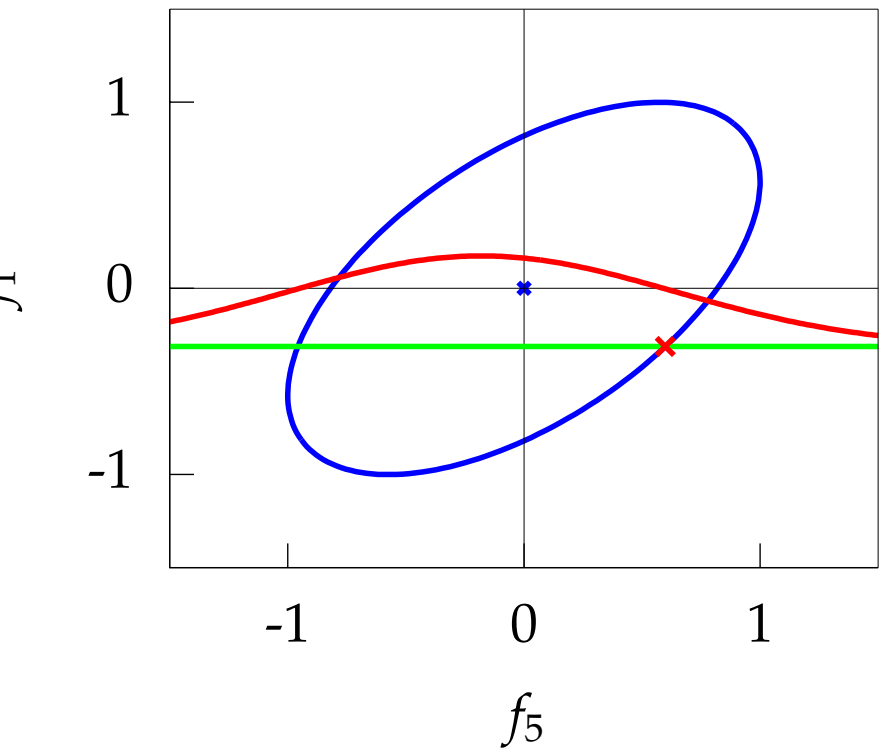
Prediction of f_5 from f_1



$$\begin{bmatrix} 1 & 0.57375 \\ 0.57375 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution, $p(f_1, f_5)$.
- We observe that $f_1 = -0.313$.
- Conditional density: $p(f_5|f_1 = -0.313)$.

Prediction of f_5 from f_1



$$\begin{bmatrix} 1 & 0.57375 \\ 0.57375 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution, $p(f_1, f_5)$.
- We observe that $f_1 = -0.313$.
- Conditional density: $p(f_5|f_1 = -0.313)$.

Prediction with Correlated Gaussians

- Prediction of $\mathbf{f}_*$ from $\mathbf{f}$ requires multivariate *conditional density*.
- Multivariate conditional density is *also* Gaussian.

$$p(\mathbf{f}_*|\mathbf{f}) = \mathcal{N}\left(\mathbf{f}_*|\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}, \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\right)$$

- Here covariance of joint density is given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix}$$

Prediction with Correlated Gaussians

- Prediction of $\mathbf{f}_*$ from $\mathbf{f}$ requires multivariate *conditional density*.
- Multivariate conditional density is *also* Gaussian.

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

$$\boldsymbol{\mu} = \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}$$

$$\boldsymbol{\Sigma} = \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}$$

- Here covariance of joint density is given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix}$$

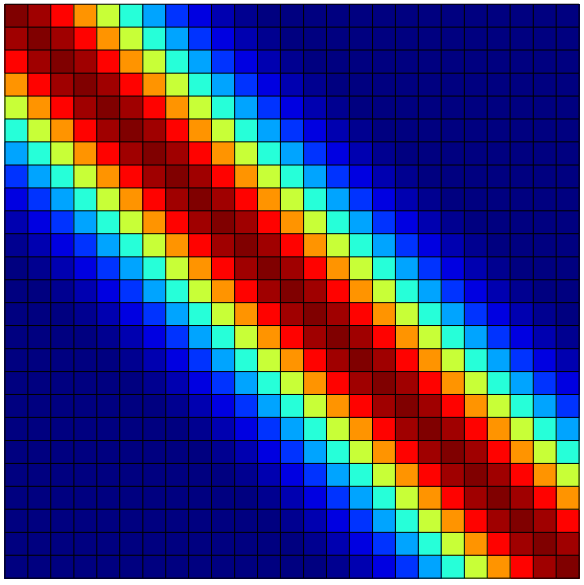
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



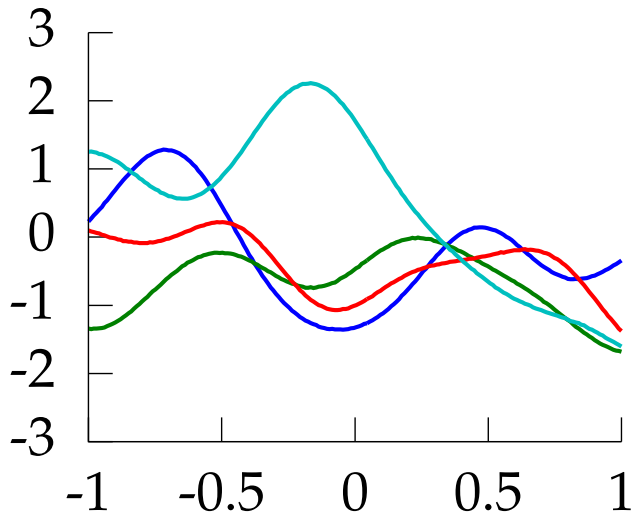
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_1 = -3.0, x_1 = -3.0$$

$$k_{1,1} = 1.00 \times \exp\left(-\frac{(-3.0 - -3.0)^2}{2 \times 2.00^2}\right)$$

$$x_1 = -3.0, x_2 = 1.20, \text{ and } x_3 = 1.40 \text{ with } \ell = 2.00 \text{ and } \alpha = 1.00.$$

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_1 = -3.0,\ x_1 = -3.0$$

$$k_{1,1} = 1.00 \times \exp\left(-\frac{(-3.0 - -3.0)^2}{2 \times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & & \\ & & \\ & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.20, x_1 = -3.0$$

$$k_{2,1} = 1.00 \times \exp\left(-\frac{(1.20 - -3.0)^2}{2 \times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & & \\ & & \\ & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.20,\ x_1 = -3.0$$

$$k_{2,1} = 1.00 \times \exp\left(-\frac{(1.20 - -3.0)^2}{2 \times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & & \\ & \boxed{0.110} & \\ & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.20,\ x_1 = -3.0$$

$$k_{2,1} = 1.00 \times \exp\left(-\frac{(1.20 - -3.0)^2}{2 \times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & 0.110 \\ 0.110 & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.20,\ x_2 = 1.20$$

$$k_{2,2} = 1.00 \times \exp\left(-\frac{(1.20-1.20)^2}{2\times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & 0.110 \\ 0.110 & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$\begin{bmatrix} 1.00 & 0.110 \\ 0.110 & 1.00 \end{bmatrix}$$

$x_2 = 1.20, x_2 = 1.20$

$$k_{2,2} = 1.00 \times \exp\left(-\frac{(1.20-1.20)^2}{2\times 2.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40, x_1 = -3.0$$

$$\begin{bmatrix} 1.00 & 0.110 \\ 0.110 & 1.00 \end{bmatrix}$$

$$k_{3,1} = 1.00 \times \exp\left(-\frac{(1.40 - -3.0)^2}{2 \times 2.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\|x_i - x_j\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_1 = -3.0$$

$$\begin{bmatrix} 1.00 & 0.110 \\ 0.110 & 1.00 \\ 0.0889 & \end{bmatrix}$$

$$k_{3,1} = 1.00 \times \exp\left(-\frac{(1.40 - -3.0)^2}{2 \times 2.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_1 = -3.0$$

$$k_{3,1} = 1.00 \times \exp\left(-\frac{(1.40 - -3.0)^2}{2 \times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & 0.110 & 0.0889 \\ 0.110 & 1.00 & \\ 0.0889 & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_2 = 1.20$$

$$k_{3,2} = 1.00 \times \exp\left(-\frac{(1.40-1.20)^2}{2\times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & 0.110 & 0.0889 \\ 0.110 & 1.00 & \\ 0.0889 & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_2 = 1.20$$

$$\begin{bmatrix} 1.00 & 0.110 & 0.0889 \\ 0.110 & 1.00 & \\ 0.0889 & 0.995 & \end{bmatrix}$$

$$k_{3,2} = 1.00 \times \exp\left(-\frac{(1.40-1.20)^2}{2\times 2.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_2 = 1.20$$

$$k_{3,2} = 1.00 \times \exp\left(-\frac{(1.40-1.20)^2}{2\times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & 0.110 & 0.0889 \\ 0.110 & 1.00 & 0.995 \\ 0.0889 & 0.995 & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_3 = 1.40$$

$$k_{3,3} = 1.00 \times \exp\left(-\frac{(1.40-1.40)^2}{2\times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & 0.110 & 0.0889 \\ 0.110 & 1.00 & 0.995 \\ 0.0889 & 0.995 & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_3 = 1.40$$

$$k_{3,3} = 1.00 \times \exp\left(-\frac{(1.40-1.40)^2}{2\times 2.00^2}\right)$$

$$\begin{bmatrix} 1.00 & 0.110 & 0.0889 \\ 0.110 & 1.00 & 0.995 \\ 0.0889 & 0.995 & 1.00 \end{bmatrix}$$

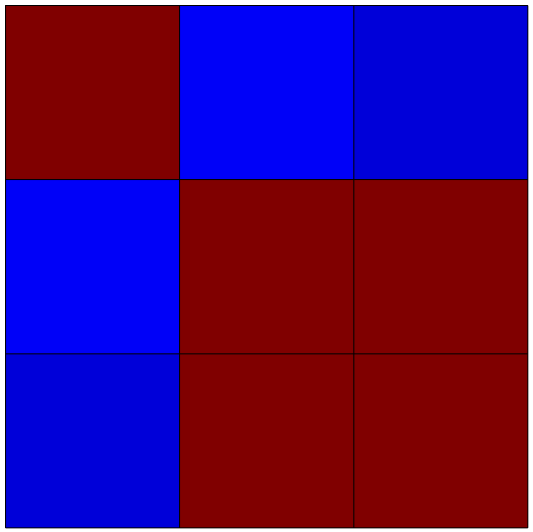
$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$x_3 = 1.40, x_3 = 1.40$



$$k_{3,3} = 1.00 \times \exp\left(-\frac{(1.40-1.40)^2}{2\times 2.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 2.00$ and $\alpha = 1.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_1 = -3, x_1 = -3$$

$$k_{1,1} = 1.0 \times \exp\left(-\frac{(-3--3)^2}{2\times 2.0^2}\right)$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_1 = -3, x_1 = -3$$

$$k_{1,1} = 1.0 \times \exp\left(-\frac{(-3--3)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & & \\ & & \\ & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.2, x_1 = -3$$

$$k_{2,1} = 1.0 \times \exp\left(-\frac{(1.2 - -3)^2}{2 \times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & & \\ & & \\ & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.2, x_1 = -3$$

$$k_{2,1} = 1.0 \times \exp\left(-\frac{(1.2 - -3)^2}{2 \times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & & \\ 0.11 & & \\ & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.2, x_1 = -3$$

$$k_{2,1} = 1.0 \times \exp\left(-\frac{(1.2 - -3)^2}{2 \times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 \\ 0.11 & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.2,\ x_2 = 1.2$$

$$\begin{bmatrix} 1.0 & 0.11 \\ 0.11 & \\ & \end{bmatrix}$$

$$k_{2,2} = 1.0 \times \exp\left(-\frac{(1.2-1.2)^2}{2\times 2.0^2}\right)$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.2,\ x_2 = 1.2$$

$$k_{2,2} = 1.0 \times \exp\left(-\frac{(1.2-1.2)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 \\ 0.11 & 1.0 \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.4,\ x_1 = -3$$

$$k_{3,1} = 1.0 \times \exp\left(-\frac{(1.4 - -3)^2}{2 \times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 \\ 0.11 & 1.0 \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.4,\ x_1 = -3$$

$$k_{3,1} = 1.0 \times \exp\left(-\frac{(1.4 - -3)^2}{2 \times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & \\ 0.11 & 1.0 & \\ \boxed{0.089} & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.4,\ x_1 = -3$$

$$k_{3,1} = 1.0 \times \exp\left(-\frac{(1.4 - -3)^2}{2 \times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 \\ 0.11 & 1.0 & \\ 0.089 & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.4,\ x_2 = 1.2$$

$$k_{3,2} = 1.0 \times \exp\left(-\frac{(1.4-1.2)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 \\ 0.11 & 1.0 & \\ 0.089 & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.4,\ x_2 = 1.2$$

$$k_{3,2} = 1.0 \times \exp\left(-\frac{(1.4-1.2)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 \\ 0.11 & 1.0 & \\ 0.089 & 1.0 & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.4,\ x_2 = 1.2$$

$$k_{3,2} = 1.0 \times \exp\left(-\frac{(1.4-1.2)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 \\ 0.11 & 1.0 & 1.0 \\ 0.089 & 1.0 & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.4, x_3 = 1.4$$

$$k_{3,3} = 1.0 \times \exp\left(-\frac{(1.4-1.4)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 \\ 0.11 & 1.0 & 1.0 \\ 0.089 & 1.0 & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.4,\ x_3 = 1.4$$

$$k_{3,3} = 1.0 \times \exp\left(-\frac{(1.4-1.4)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 \\ 0.11 & 1.0 & 1.0 \\ 0.089 & 1.0 & 1.0 \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0, x_1 = -3$$

$$k_{4,1} = 1.0 \times \exp\left(-\frac{(2.0 - -3)^2}{2 \times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 \\ 0.11 & 1.0 & 1.0 \\ 0.089 & 1.0 & 1.0 \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0,\ x_1 = -3$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 \\ 0.11 & 1.0 & 1.0 \\ 0.089 & 1.0 & 1.0 \\ 0.044 & & \end{bmatrix}$$

$$k_{4,1} = 1.0 \times \exp\left(-\frac{(2.0 - -3)^2}{2 \times 2.0^2}\right)$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0,\ x_1 = -3$$

$$k_{4,1} = 1.0 \times \exp\left(-\frac{(2.0 - -3)^2}{2 \times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 & 0.044 \\ 0.11 & 1.0 & 1.0 & \\ 0.089 & 1.0 & 1.0 & \\ 0.044 & & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0,\ x_2 = 1.2$$

$$k_{4,2} = 1.0 \times \exp\left(-\frac{(2.0-1.2)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 & 0.044 \\ 0.11 & 1.0 & 1.0 & \\ 0.089 & 1.0 & 1.0 & \\ 0.044 & & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0,\ x_2 = 1.2$$

$$k_{4,2} = 1.0 \times \exp\left(-\frac{(2.0-1.2)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 & 0.044 \\ 0.11 & 1.0 & 1.0 & \\ 0.089 & 1.0 & 1.0 & \\ 0.044 & 0.92 & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0,\ x_2 = 1.2$$

$$k_{4,2} = 1.0 \times \exp\left(-\frac{(2.0-1.2)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 & 0.044 \\ 0.11 & 1.0 & 1.0 & 0.92 \\ 0.089 & 1.0 & 1.0 & \\ 0.044 & 0.92 & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0,\ x_3 = 1.4$$

$$k_{4,3} = 1.0 \times \exp\left(-\frac{(2.0-1.4)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 & 0.044 \\ 0.11 & 1.0 & 1.0 & 0.92 \\ 0.089 & 1.0 & 1.0 & \\ 0.044 & 0.92 & & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0,\ x_3 = 1.4$$

$$k_{4,3} = 1.0 \times \exp\left(-\frac{(2.0-1.4)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 & 0.044 \\ 0.11 & 1.0 & 1.0 & 0.92 \\ 0.089 & 1.0 & 1.0 & \\ 0.044 & 0.92 & 0.96 & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0,\ x_3 = 1.4$$

$$k_{4,3} = 1.0 \times \exp\left(-\frac{(2.0-1.4)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 & 0.044 \\ 0.11 & 1.0 & 1.0 & 0.92 \\ 0.089 & 1.0 & 1.0 & 0.96 \\ 0.044 & 0.92 & 0.96 & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0, x_4 = 2.0$$

$$k_{4,4} = 1.0 \times \exp\left(-\frac{(2.0-2.0)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 & 0.044 \\ 0.11 & 1.0 & 1.0 & 0.92 \\ 0.089 & 1.0 & 1.0 & 0.96 \\ 0.044 & 0.92 & 0.96 & \end{bmatrix}$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_4 = 2.0,\ x_4 = 2.0$$

$$k_{4,4} = 1.0 \times \exp\left(-\frac{(2.0-2.0)^2}{2\times 2.0^2}\right)$$

$$\begin{bmatrix} 1.0 & 0.11 & 0.089 & 0.044 \\ 0.11 & 1.0 & 1.0 & 0.92 \\ 0.089 & 1.0 & 1.0 & 0.96 \\ 0.044 & 0.92 & 0.96 & 1.0 \end{bmatrix}$$

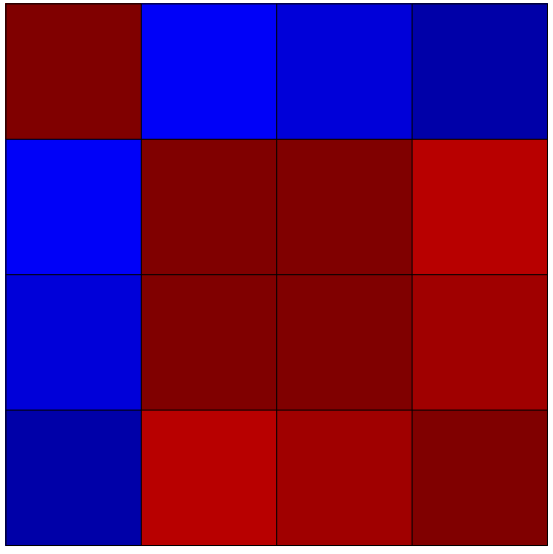
$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$x_4 = 2.0, x_4 = 2.0$



$$k_{4,4} = 1.0 \times \exp\left(-\frac{(2.0-2.0)^2}{2\times 2.0^2}\right)$$

$x_1 = -3$, $x_2 = 1.2$, $x_3 = 1.4$, and $x_4 = 2.0$ with $\ell = 2.0$ and $\alpha = 1.0$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_1 = -3.0, x_1 = -3.0$$

$$k_{1,1} = 4.00 \times \exp\left(-\frac{(-3.0 - -3.0)^2}{2 \times 5.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_1 = -3.0,\ x_1 = -3.0$$

$$k_{1,1} = 4.00 \times \exp\left(-\frac{(-3.0 - -3.0)^2}{2 \times 5.00^2}\right)$$

$$\begin{bmatrix} 4.00 & & \\ & & \\ & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.20,\ x_1 = -3.0$$

$$k_{2,1} = 4.00 \times \exp\left(-\frac{(1.20 - -3.0)^2}{2 \times 5.00^2}\right)$$

$$\begin{bmatrix} 4.00 & & \\ & & \\ & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.20,\ x_1 = -3.0$$

$$k_{2,1} = 4.00 \times \exp\left(-\frac{(1.20 - -3.0)^2}{2 \times 5.00^2}\right)$$

$$\begin{bmatrix} 4.00 & & \\ & 2.81 & \\ & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.20,\ x_1 = -3.0$$

$$\begin{bmatrix} 4.00 & 2.81 \\ 2.81 & \end{bmatrix}$$

$$k_{2,1} = 4.00 \times \exp\left(-\frac{(1.20 - -3.0)^2}{2 \times 5.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.20,\ x_2 = 1.20$$

$$k_{2,2} = 4.00 \times \exp\left(-\frac{(1.20-1.20)^2}{2\times 5.00^2}\right)$$

$$\begin{bmatrix} & & \\ & 4.00 & 2.81 \\ & & \\ & 2.81 & \\ & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_2 = 1.20, x_2 = 1.20$$

$$k_{2,2} = 4.00 \times \exp\left(-\frac{(1.20 - 1.20)^2}{2 \times 5.00^2}\right)$$

$$\begin{bmatrix} 4.00 & 2.81 & \\ 2.81 & \boxed{4.00} & \\ & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40, x_1 = -3.0$$

$$\begin{bmatrix} & \\ & \end{bmatrix}$$

$$k_{3,1} = 4.00 \times \exp\left(-\frac{(1.40 - -3.0)^2}{2 \times 5.00^2}\right)$$

$$\begin{bmatrix} 4.00 & 2.81 \\ 2.81 & 4.00 \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_1 = -3.0$$

$$\begin{bmatrix} 4.00 & 2.81 \\ 2.81 & 4.00 \end{bmatrix}$$

$$k_{3,1} = 4.00 \times \exp\left(-\frac{(1.40 - -3.0)^2}{2 \times 5.00^2}\right)$$

$$\begin{bmatrix} 2.72 \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40, x_1 = -3.0$$

$$\begin{bmatrix} 4.00 & 2.81 & 2.72 \\ 2.81 & 4.00 & \\ 2.72 & & \end{bmatrix}$$

$$k_{3,1} = 4.00 \times \exp\left(-\frac{(1.40 - -3.0)^2}{2 \times 5.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_2 = 1.20$$

$$k_{3,2} = 4.00 \times \exp\left(-\frac{(1.40-1.20)^2}{2\times 5.00^2}\right)$$

$$\begin{bmatrix} 4.00 & 2.81 & 2.72 \\ 2.81 & 4.00 & \\ 2.72 & & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_2 = 1.20$$

$$\begin{bmatrix} 4.00 & 2.81 & 2.72 \\ 2.81 & 4.00 & \\ 2.72 & 4.00 & \end{bmatrix}$$

$$k_{3,2} = 4.00 \times \exp\left(-\frac{(1.40-1.20)^2}{2\times 5.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_2 = 1.20$$

$$k_{3,2} = 4.00 \times \exp\left(-\frac{(1.40-1.20)^2}{2\times 5.00^2}\right)$$

$$\begin{bmatrix} 4.00 & 2.81 & 2.72 \\ 2.81 & 4.00 & 4.00 \\ 2.72 & 4.00 & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40, x_3 = 1.40$$

$$k_{3,3} = 4.00 \times \exp\left(-\frac{(1.40-1.40)^2}{2\times 5.00^2}\right)$$

$$\begin{bmatrix} 4.00 & 2.81 & 2.72 \\ 2.81 & 4.00 & 4.00 \\ 2.72 & 4.00 & \end{bmatrix}$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$$x_3 = 1.40,\ x_3 = 1.40$$

$$k_{3,3} = 4.00 \times \exp\left(-\frac{(1.40-1.40)^2}{2\times 5.00^2}\right)$$

$$\begin{bmatrix} 4.00 & 2.81 & 2.72 \\ 2.81 & 4.00 & 4.00 \\ 2.72 & 4.00 & 4.00 \end{bmatrix}$$

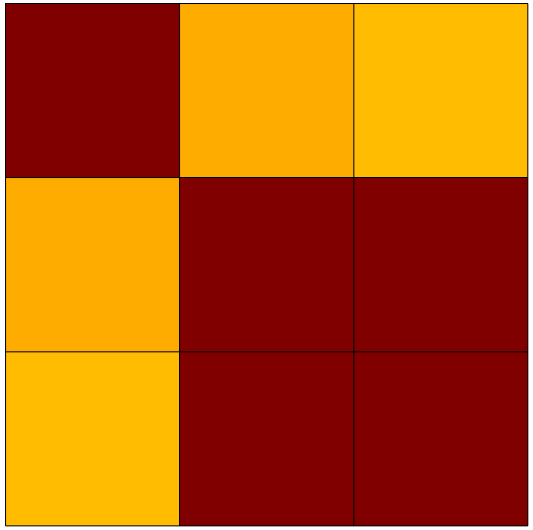
$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Covariance Functions

Where did this covariance matrix come from?

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\ell^2}\right)$$

$x_3 = 1.40, x_3 = 1.40$



$$k_{3,3} = 4.00 \times \exp\left(-\frac{(1.40-1.40)^2}{2\times 5.00^2}\right)$$

$x_1 = -3.0$, $x_2 = 1.20$, and $x_3 = 1.40$ with $\ell = 5.00$ and $\alpha = 4.00$.

Gaussian Process Interpolation

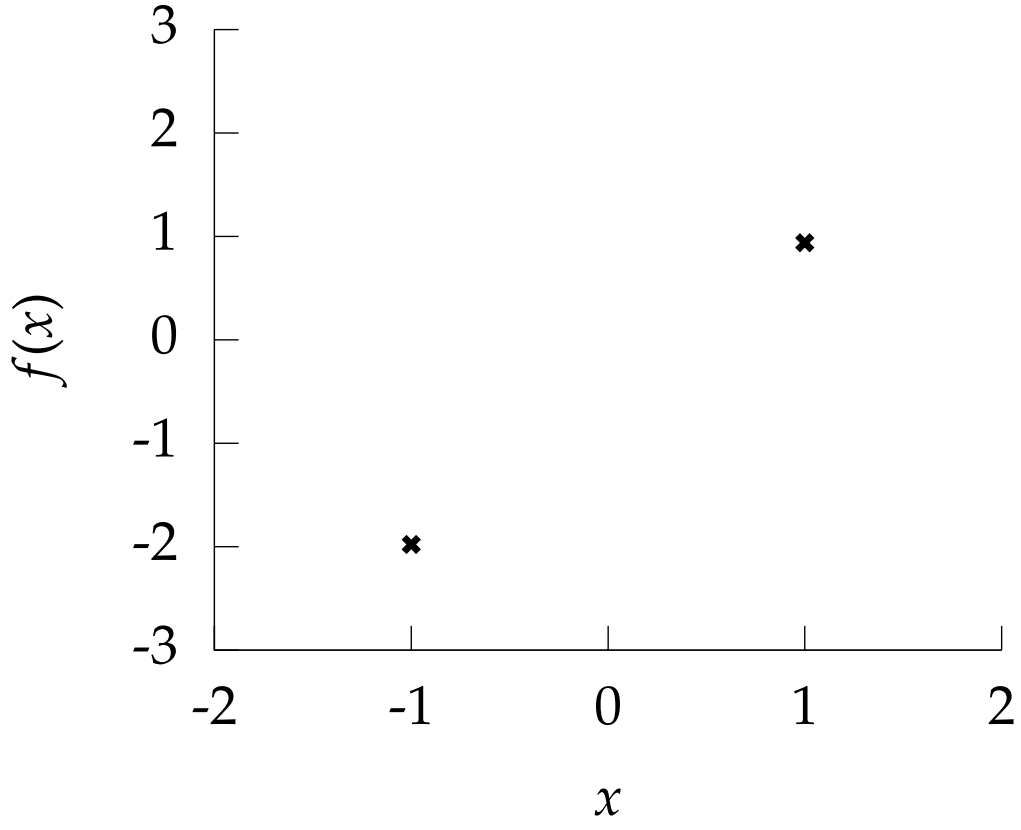


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

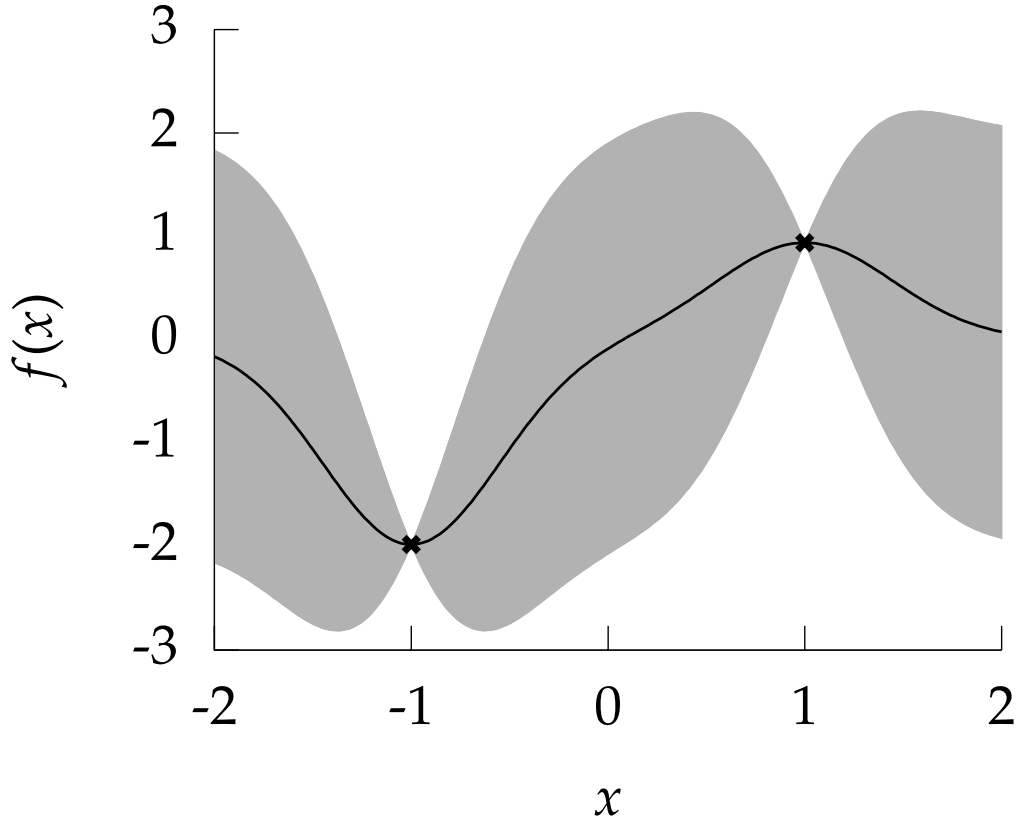


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

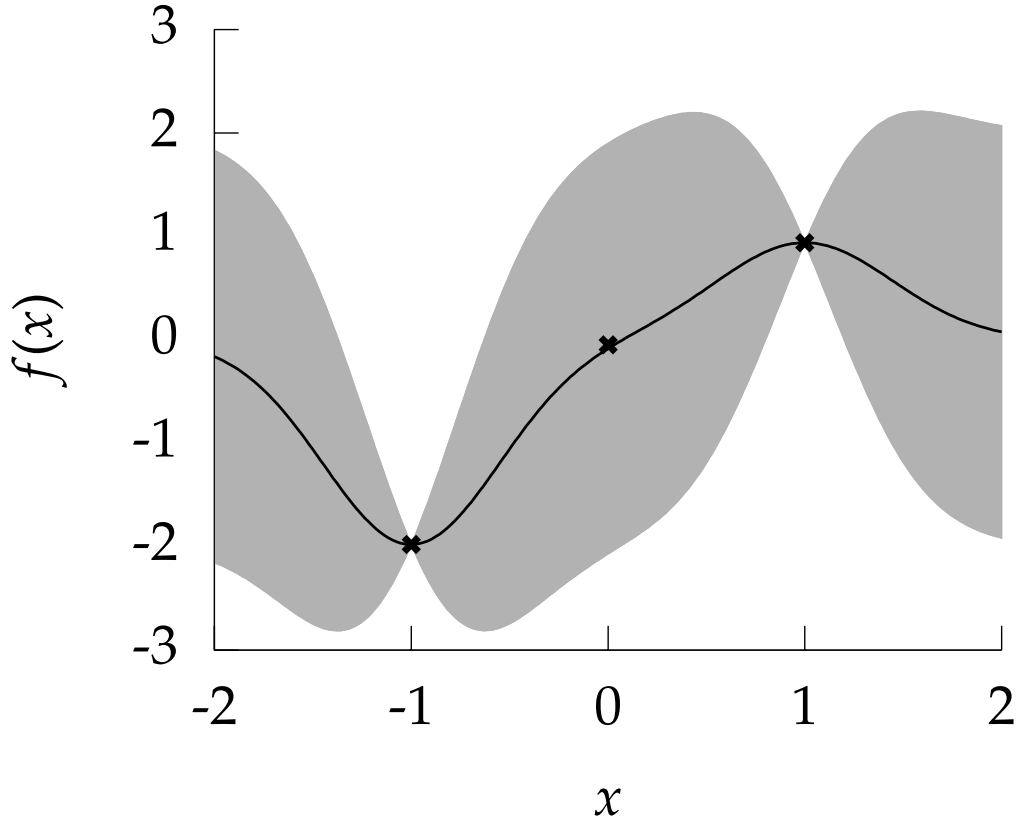


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

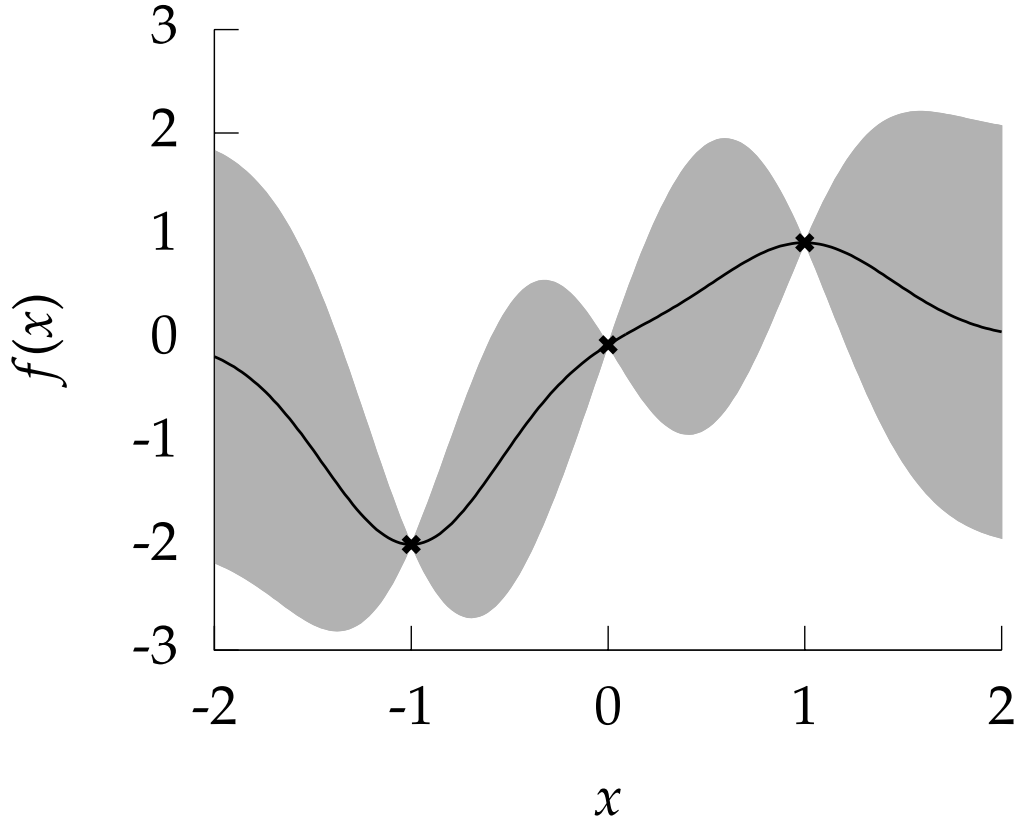


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

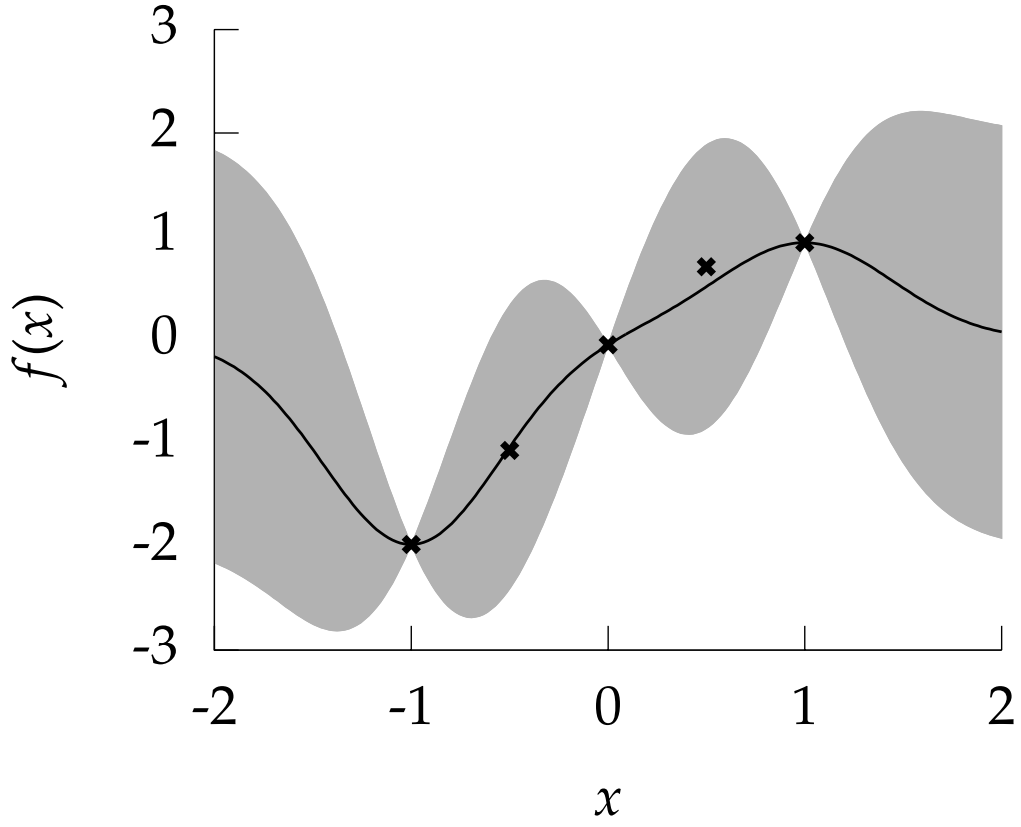


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

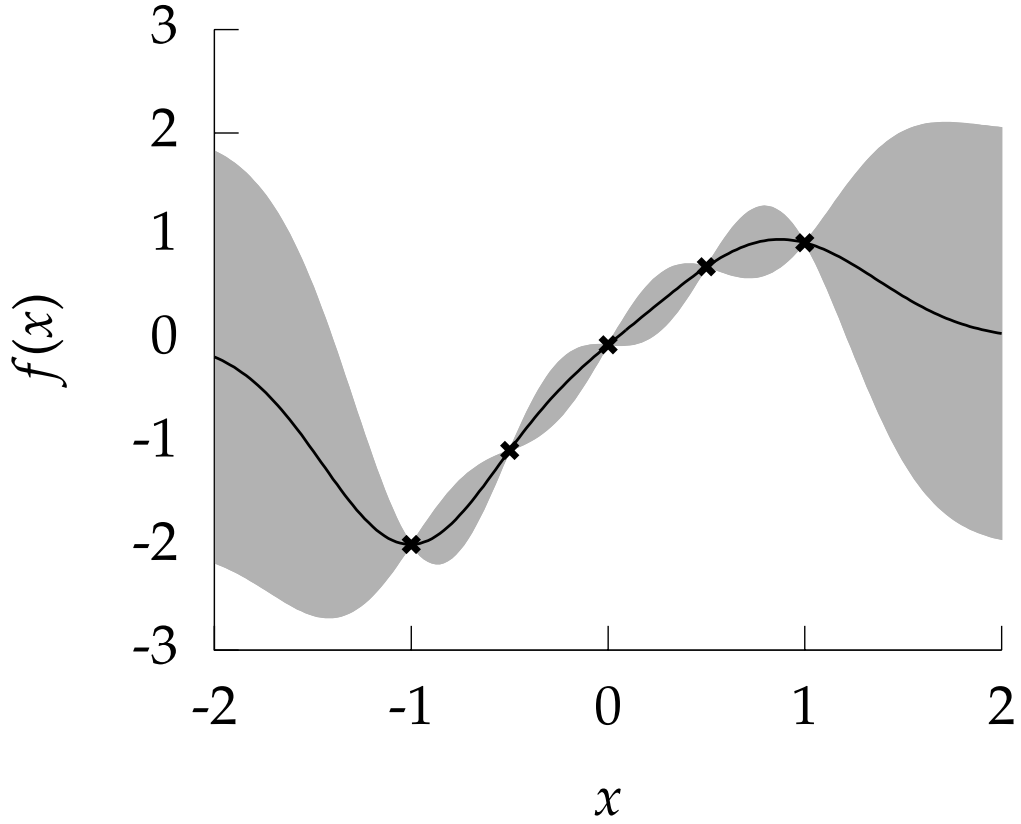


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

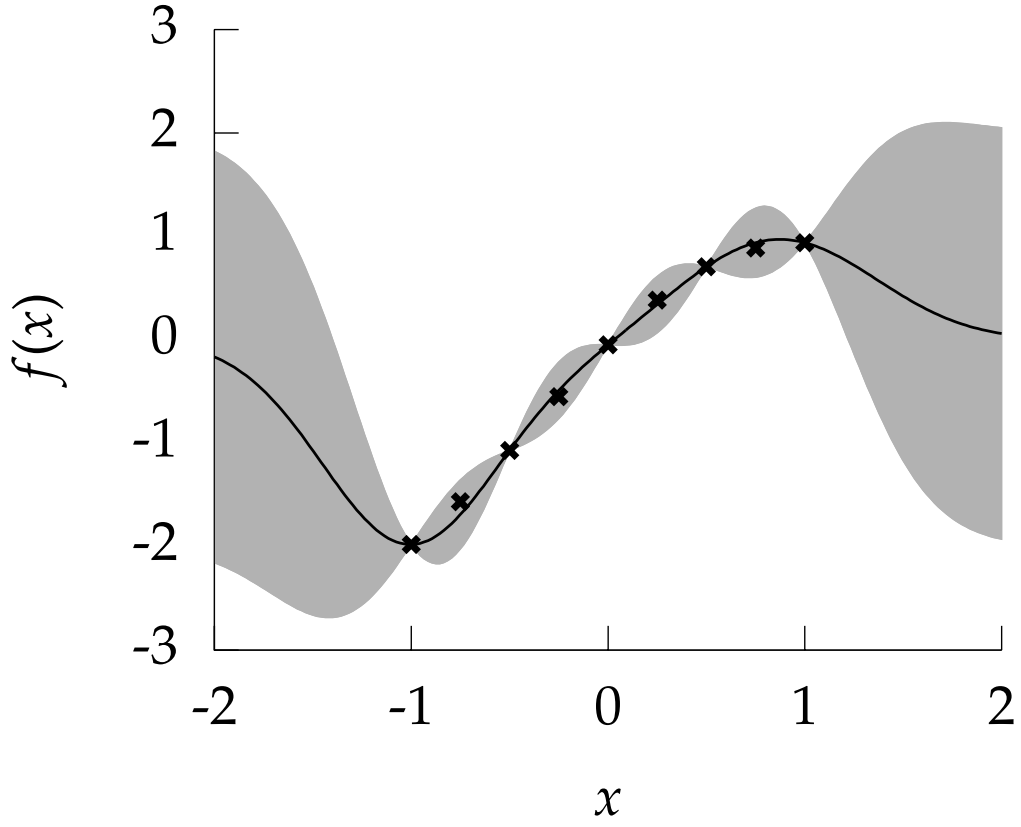


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

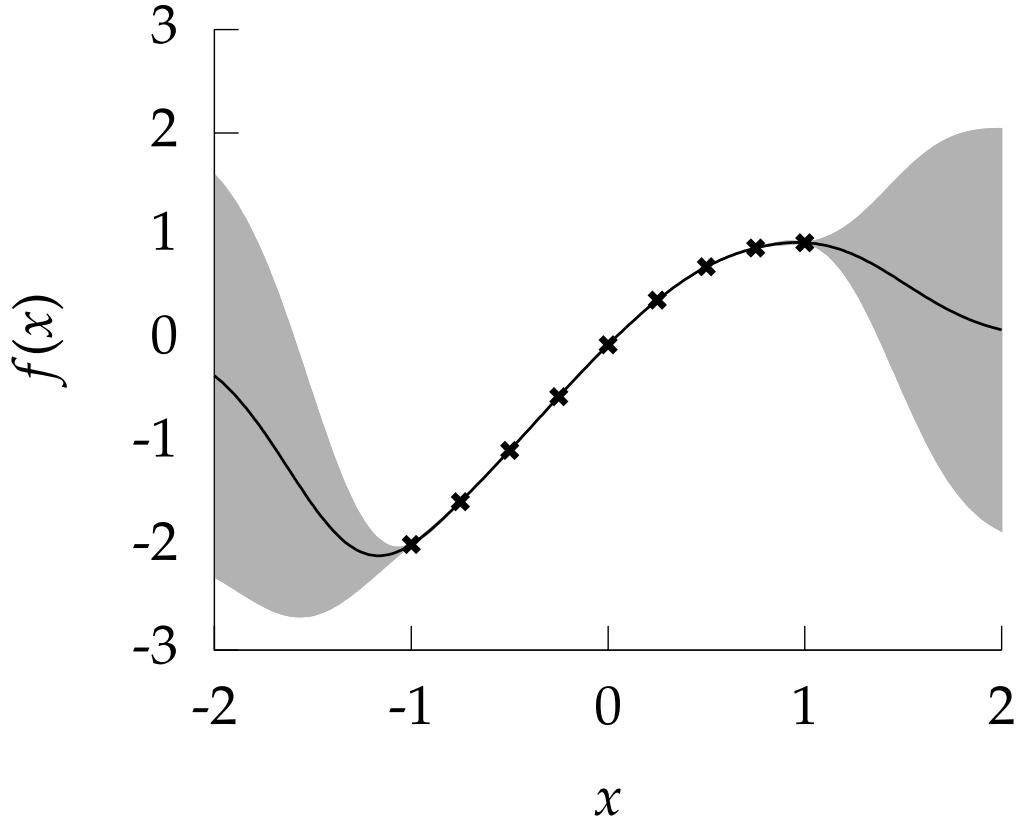


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Noise

- Gaussian noise model,

$$p\left(y_i|f_i\right) = \mathcal{N}\left(y_i|f_i, \sigma^2\right)$$

  where σ^2 is the variance of the noise.
- Equivalent to a covariance function of the form

$$k(\mathbf{x}_i, \mathbf{x}_j) = \delta_{i,j}\sigma^2$$

  where $\delta_{i,j}$ is the Kronecker delta function.
- Additive nature of Gaussians means we can simply add this term to existing covariance matrices.

Gaussian Process Regression

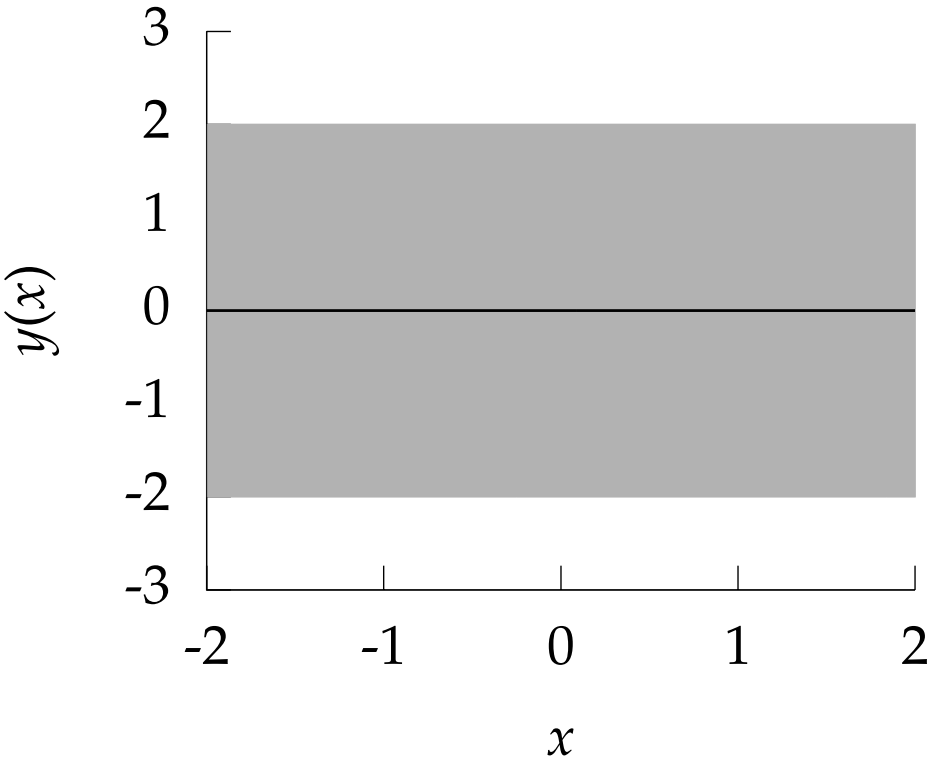


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

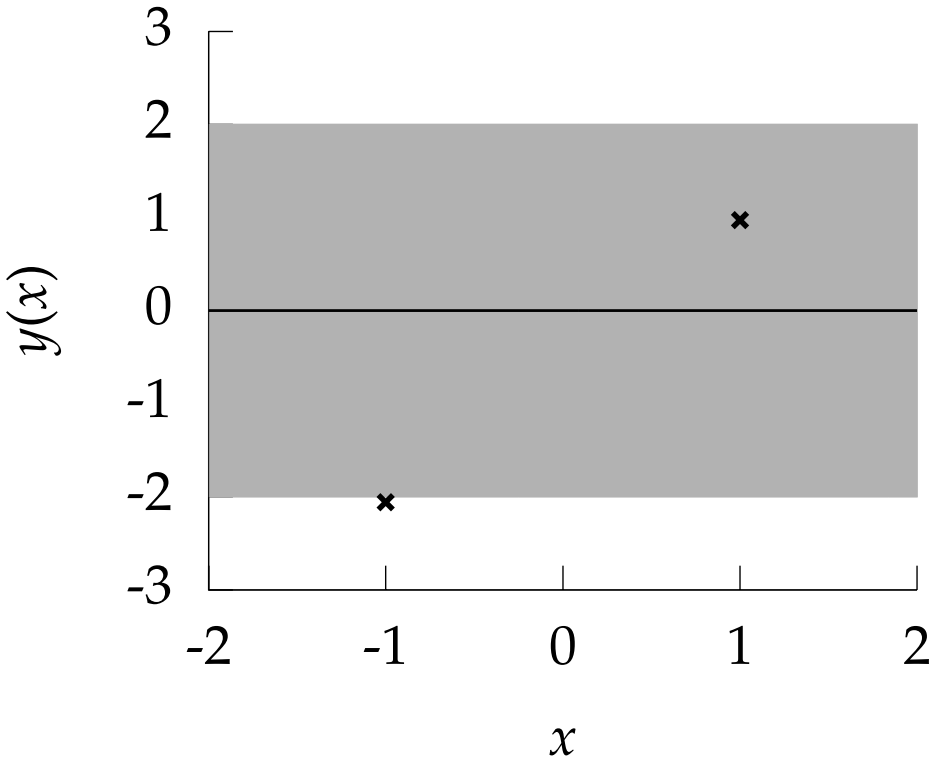


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

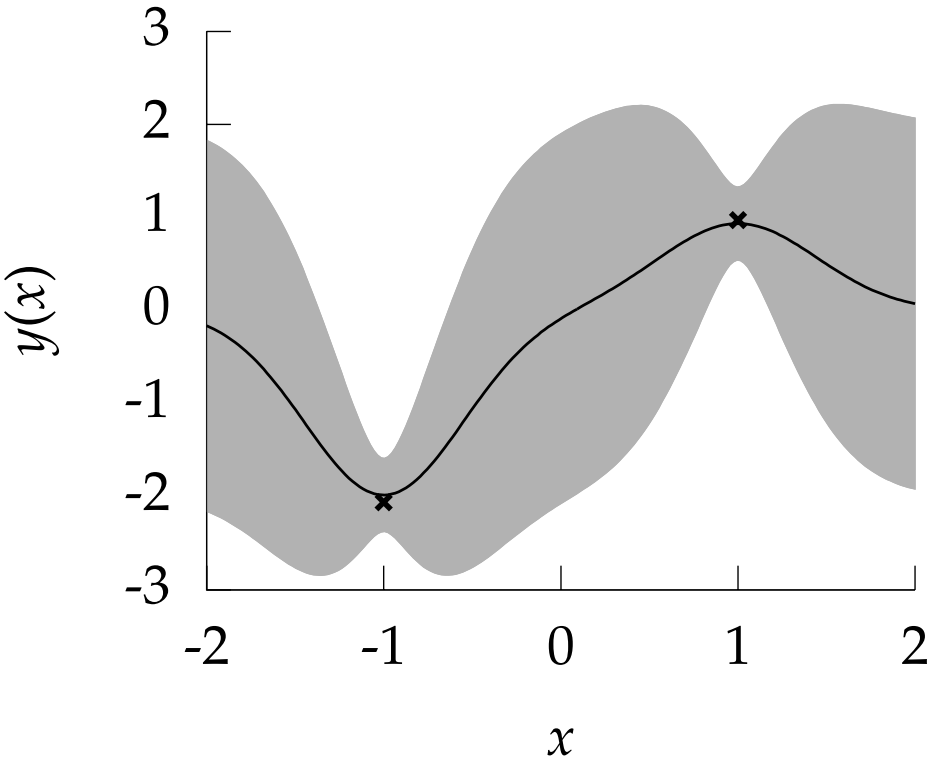


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

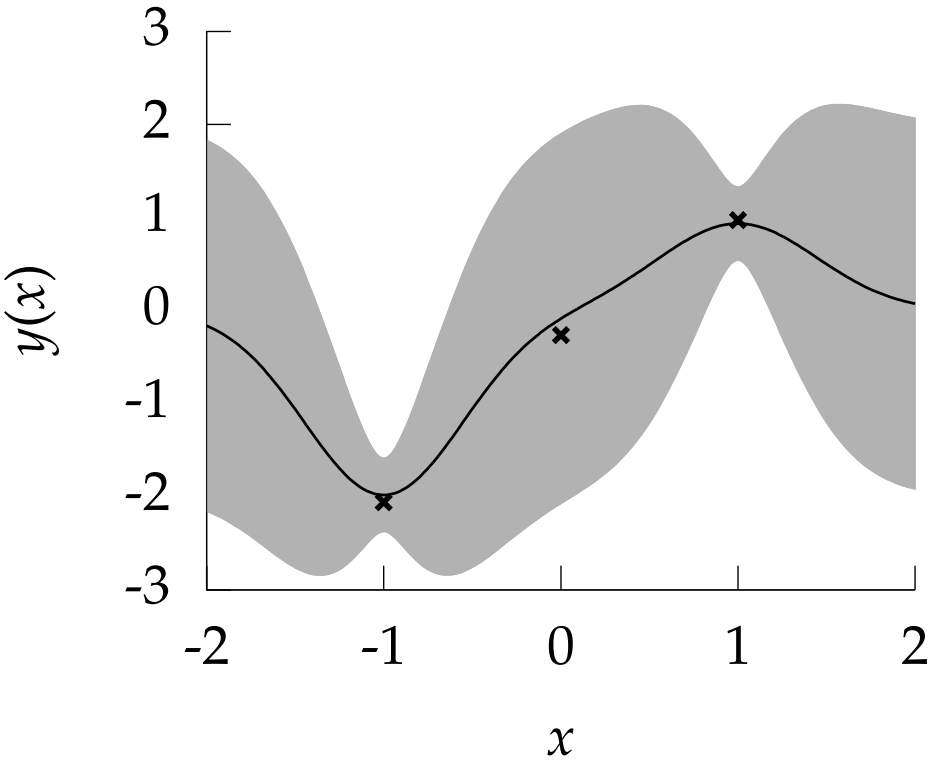


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

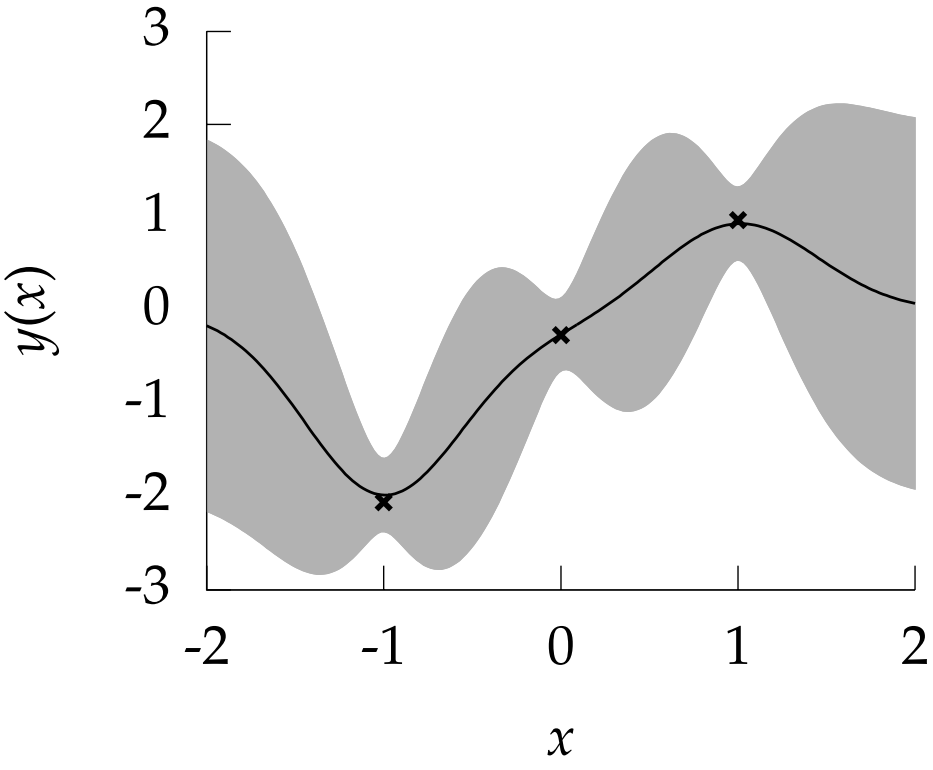


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

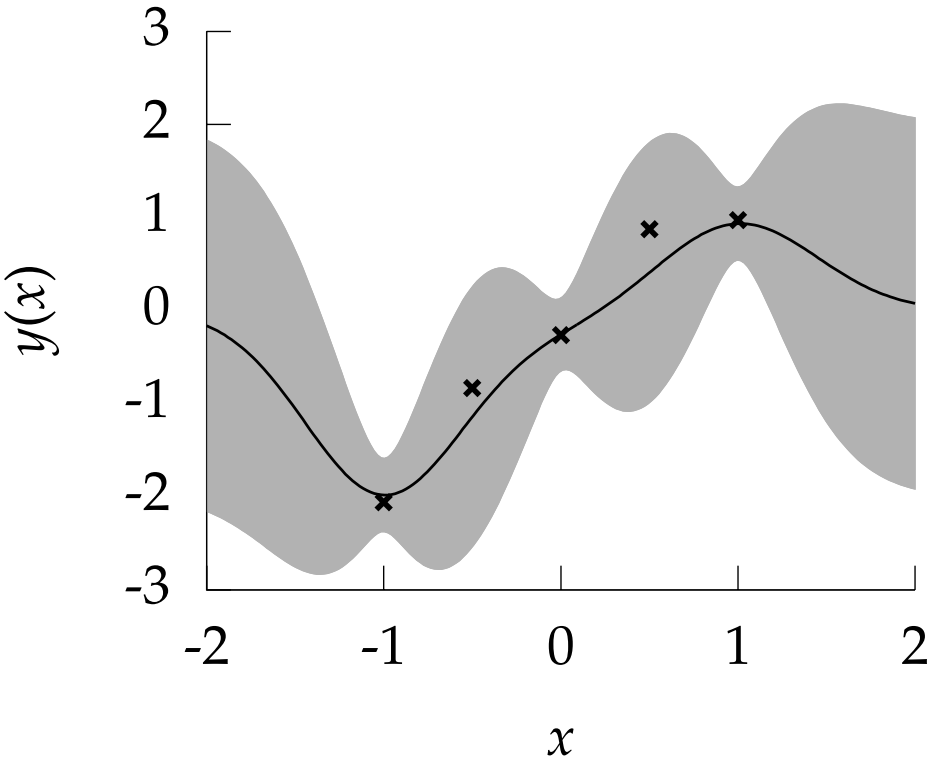


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

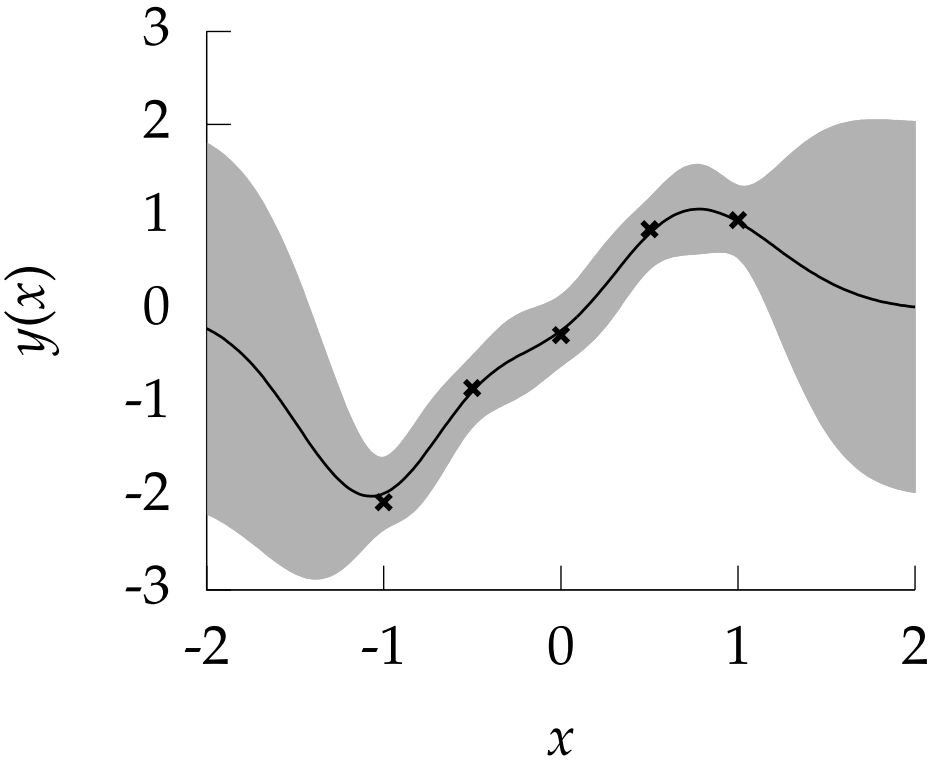


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

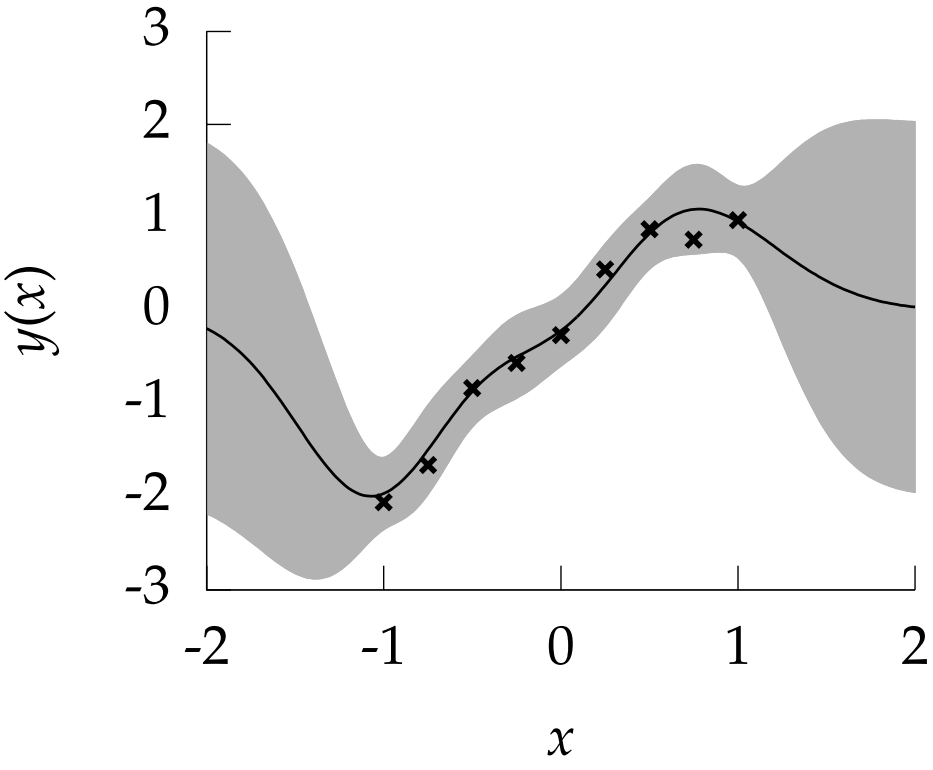


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

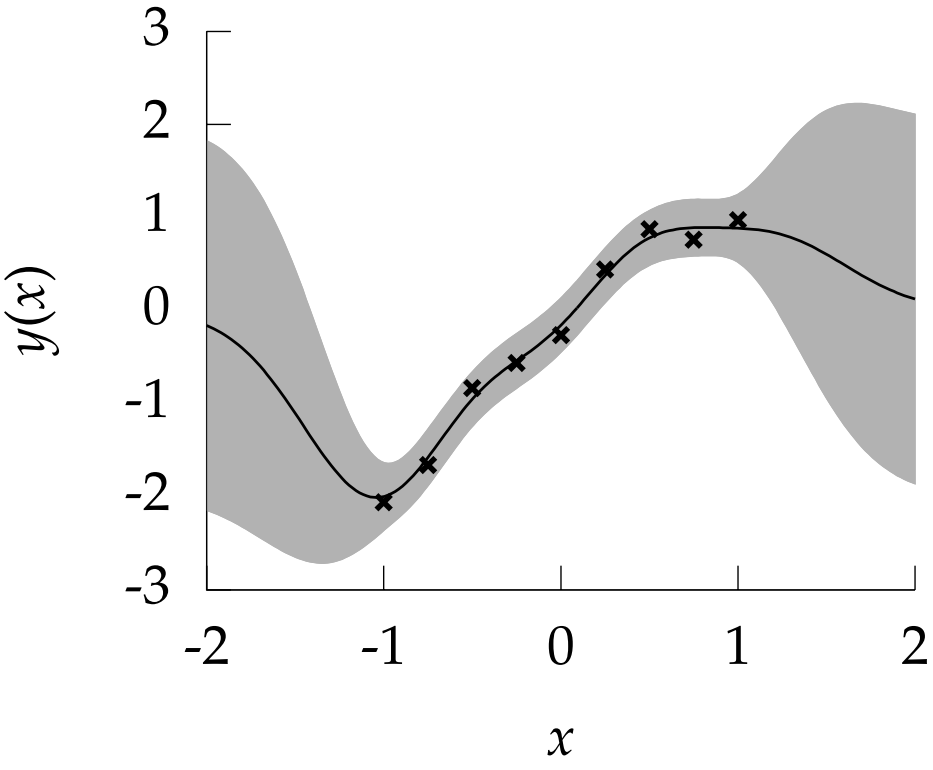
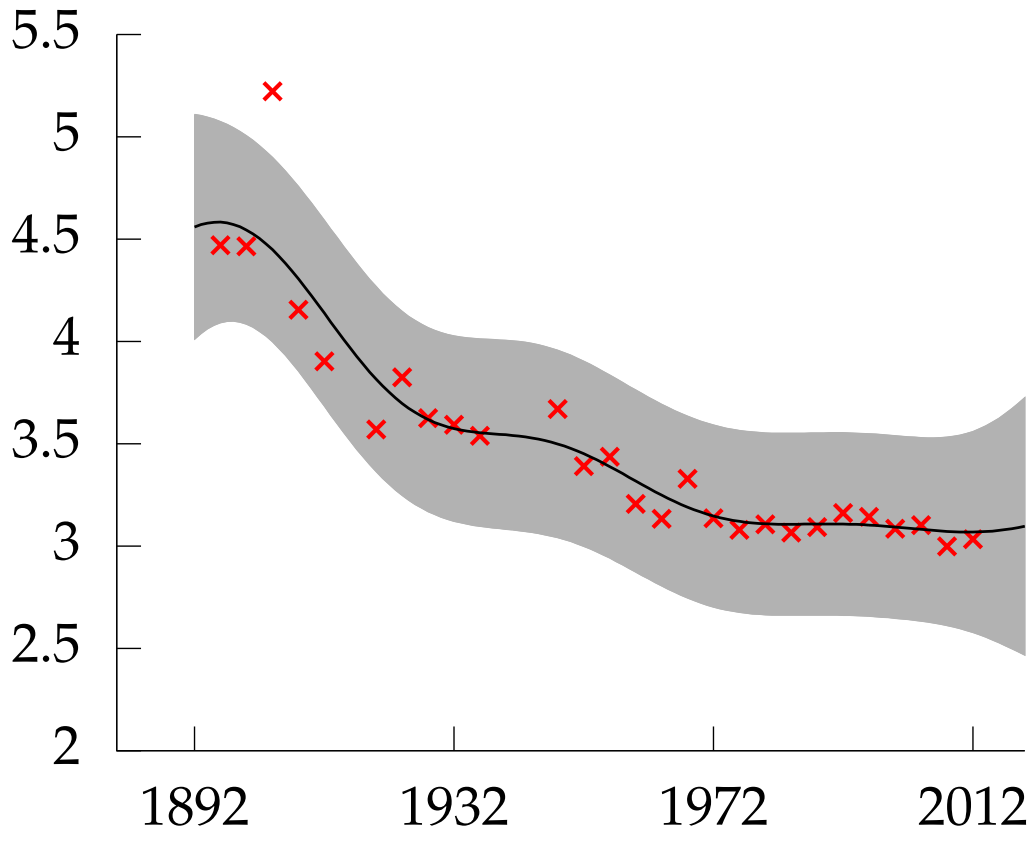


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Fit to Olympic Marathon Data



Outline

General Noise Models

Graph of a GP

- Relates input variables, $\mathbf{X}$, to vector, $\mathbf{y}$, through $\mathbf{f}$ given kernel parameters $\boldsymbol{\theta}$.
- Plate notation indicates independence of $y_i|f_i$.
- In general $p\left(y_i|f_i\right)$ is non-Gaussian.
- We approximate with Gaussian $p\left(y_i|f_i\right) \approx \mathcal{N}\left(m_i|f_i, \beta_i^{-1}\right)$.

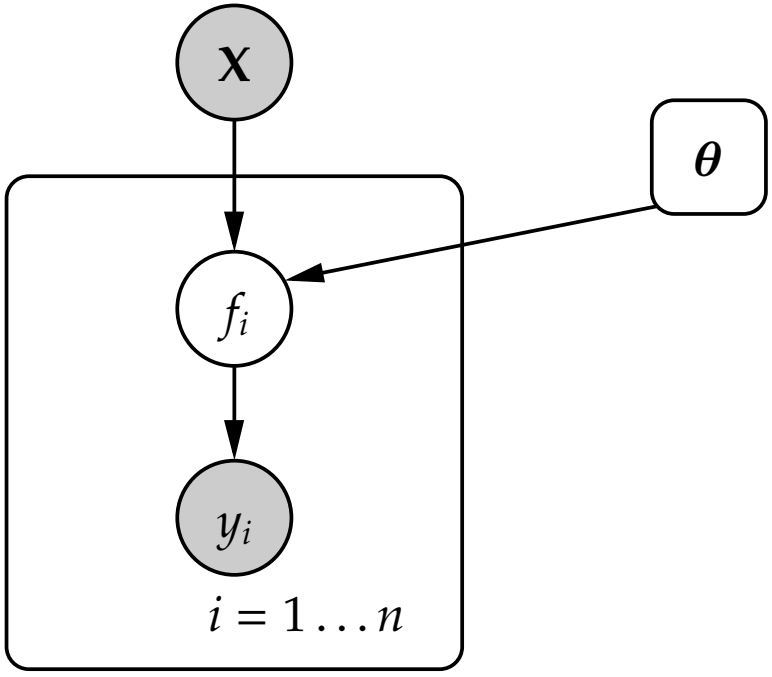


Figure: The Gaussian process depicted graphically.

Gaussian Noise

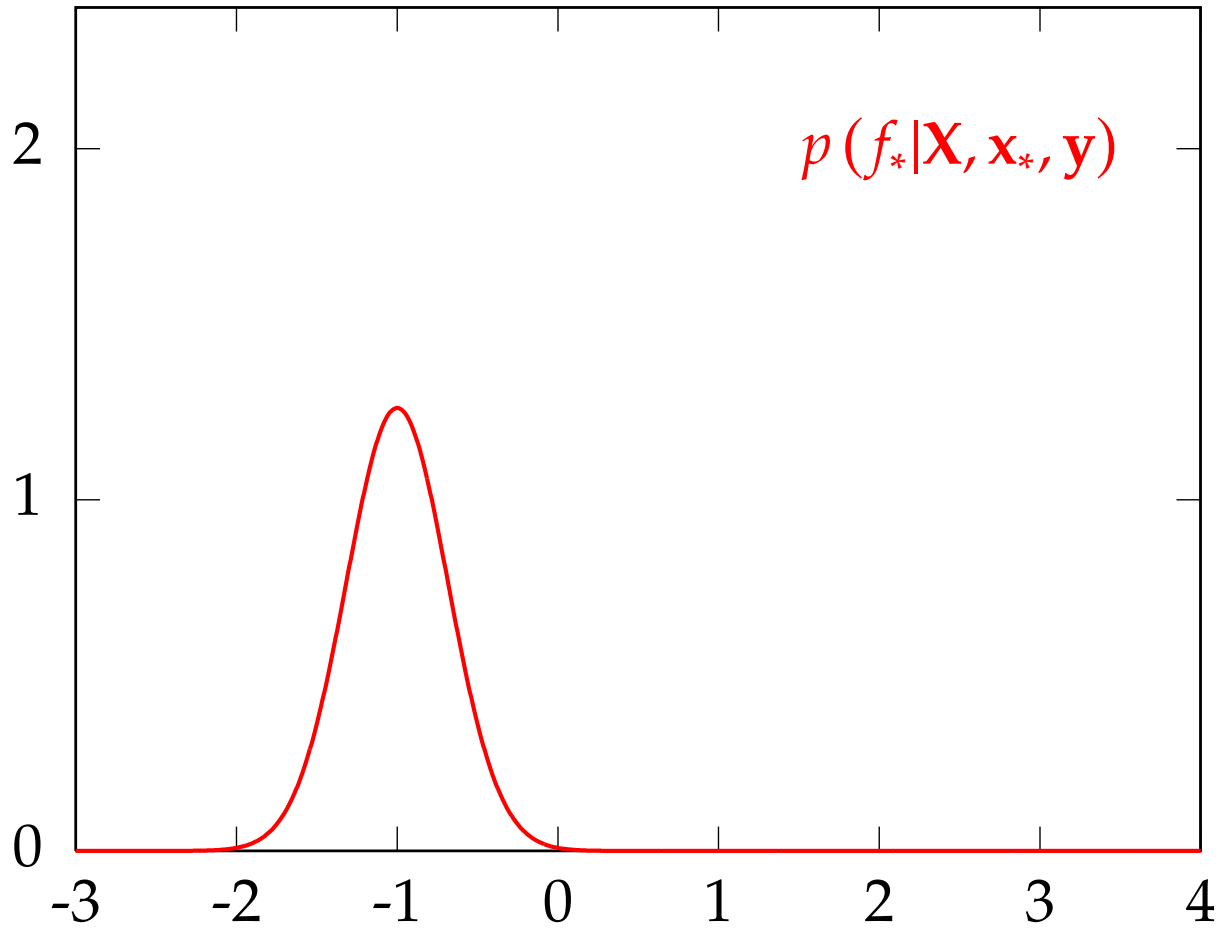


Figure: Inclusion of a data point with Gaussian noise.

Gaussian Noise

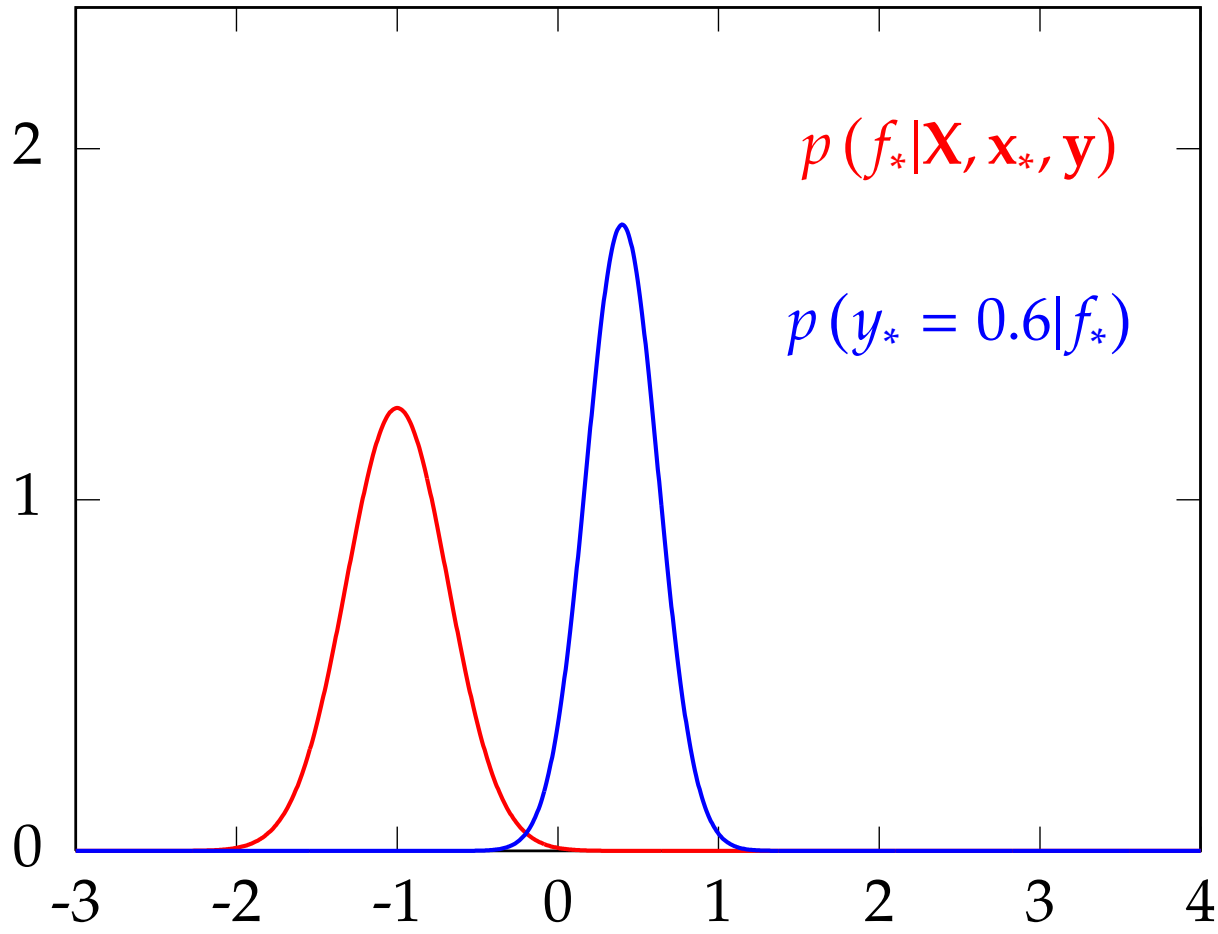


Figure: Inclusion of a data point with Gaussian noise.

Gaussian Noise

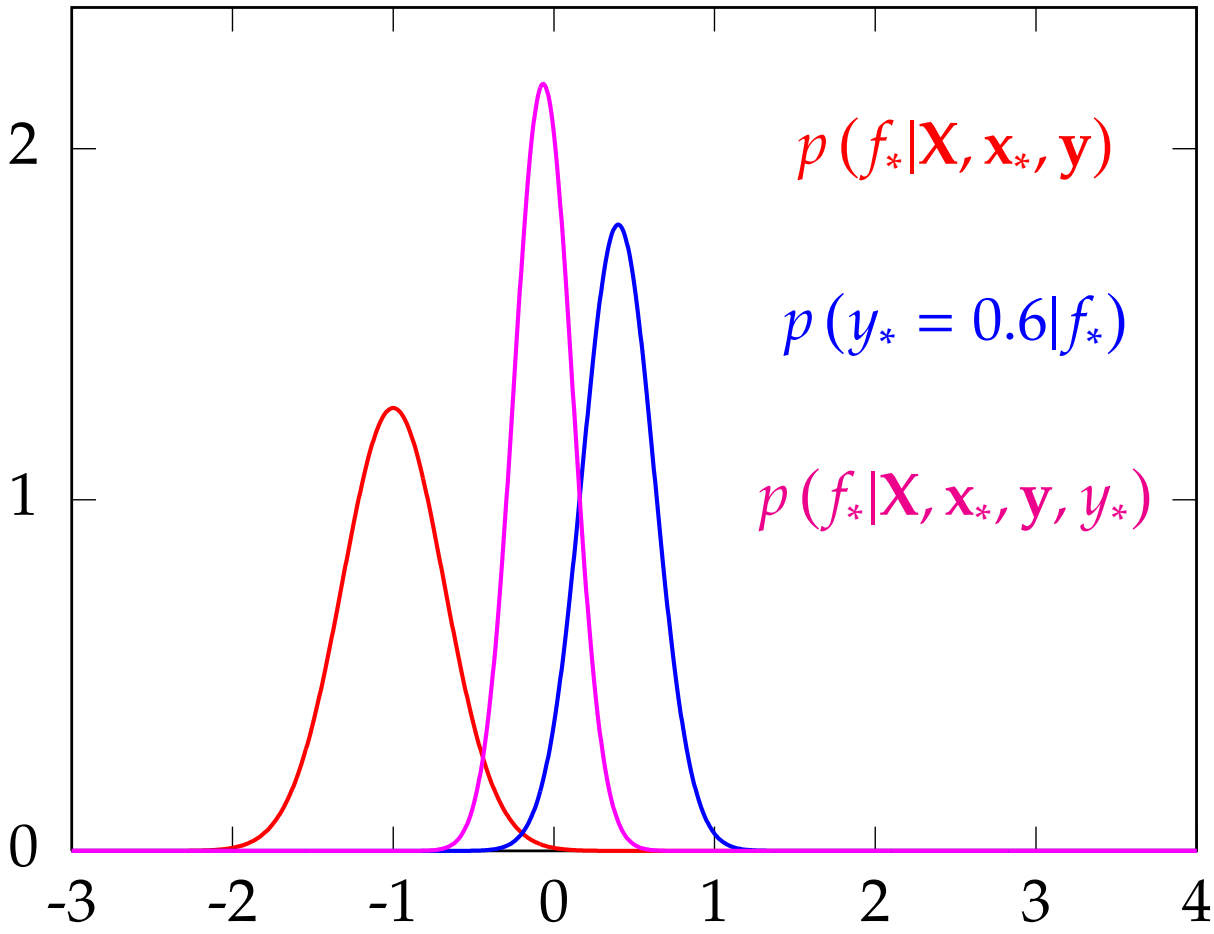


Figure: Inclusion of a data point with Gaussian noise.

Expectation Propagation

Local Moment Matching

- Easiest to consider a single previously unseen data point, $y_*, \mathbf{x}_*$.
- Before seeing data point, prediction of f_* is a GP, $q(f_*|\mathbf{y}, \mathbf{X})$.
- Update prediction using Bayes' Rule,

$$p(f_*|\mathbf{y}, y_*, \mathbf{X}, \mathbf{x}_*) = \frac{p(y_*|f_*)\, p(f_*|\mathbf{y}, \mathbf{X}, \mathbf{x}_*)}{p(\mathbf{y}, y_*|\mathbf{X}, \mathbf{x}_*)}.$$

This posterior is not a Gaussian process if $p(y_*|f_*)$ is non-Gaussian.

Classification Noise Model

Probit Noise Model

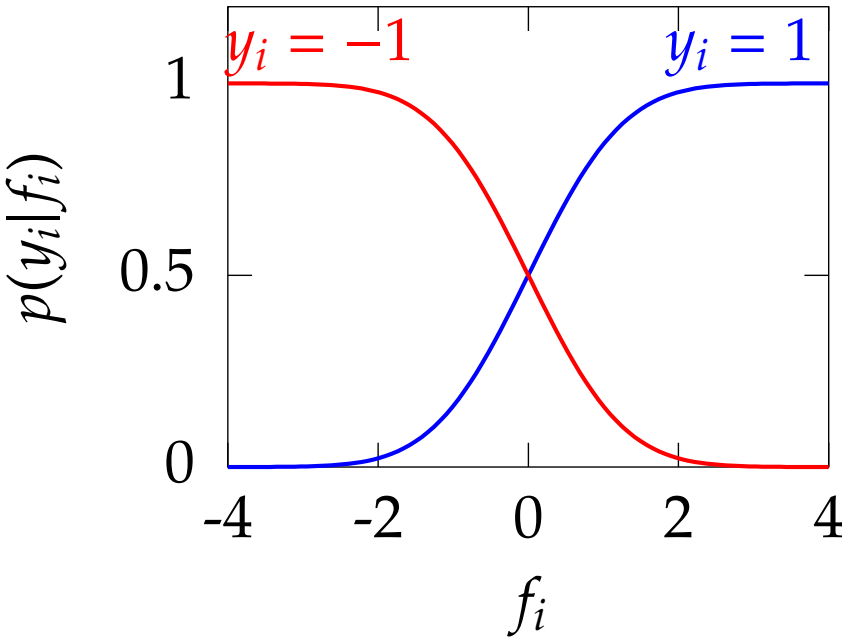


Figure: The probit model (classification). The plot shows $p(y_i|f_i)$ for different values of y_i. For $y_i = 1$ we have

$$p(y_i|f_i) = \phi(f_i) = \int_{-\infty}^{f_i} \mathcal{N}(z|0,1)\,\mathrm{d}z.$$

Expectation Propagation II

Match Moments

- Idea behind EP — approximate with a Gaussian process at this stage by matching moments.
- This is equivalent to minimizing the following KL divergence where $q\left(f_*|\mathbf{y}, y_*, \mathbf{X}, \mathbf{x}_*\right)$ is constrained to be a GP.

$$q\left(f_*|\mathbf{y}, y_*\mathbf{X}, \mathbf{x}_*\right) = \text{argmin}_{q\left(f_*|\mathbf{y}, y_*\mathbf{X}, \mathbf{x}_*\right)} \text{KL}\left(p\left(f_*|\mathbf{y}, y_*\mathbf{X}, \mathbf{x}_*\right) || q\left(f_*|\mathbf{y}, y_*, \mathbf{X}, \mathbf{x}_*\right)\right)$$

- This is equivalent to setting

$$\left\langle f_* \right\rangle_{q\left(f_*|\mathbf{y}, y_*, \mathbf{X}, \mathbf{x}_*\right)} = \left\langle f_* \right\rangle_{p\left(f_*|\mathbf{y}, y_*, \mathbf{X}, \mathbf{x}_*\right)}$$

$$\left\langle f_*^2 \right\rangle_{q\left(f_*|\mathbf{y}, y_*, \mathbf{X}, \mathbf{x}_*\right)} = \left\langle f_*^2 \right\rangle_{p\left(f_*|\mathbf{y}, y_*, \mathbf{X}, \mathbf{x}_*\right)}$$

Expectation Propagation III

Equivalent Gaussian

- This is achieved by replacing $p\left(y_*|f_*\right)$ with a Gaussian distribution

$$p\left(f_*|\mathbf{y}, y_*, \mathbf{X}, \mathbf{x}_*\right) = \frac{p\left(y_*|f_*\right) p\left(f_*|\mathbf{y}, \mathbf{X}, \mathbf{x}_*\right)}{p\left(\mathbf{y}, y_*|\mathbf{X}, \mathbf{x}_*\right)}$$

becomes

$$q\left(f_*|\mathbf{y}, y_*, \mathbf{X}, \mathbf{x}_*\right) = \frac{\mathcal{N}\left(m_*|f_*, \beta_m^{-1}\right) p\left(f_*|\mathbf{y}, \mathbf{X}, \mathbf{x}_*\right)}{p\left(\mathbf{y}, y_*|\mathbf{X}, \mathbf{x}_*\right)}.$$

Classification

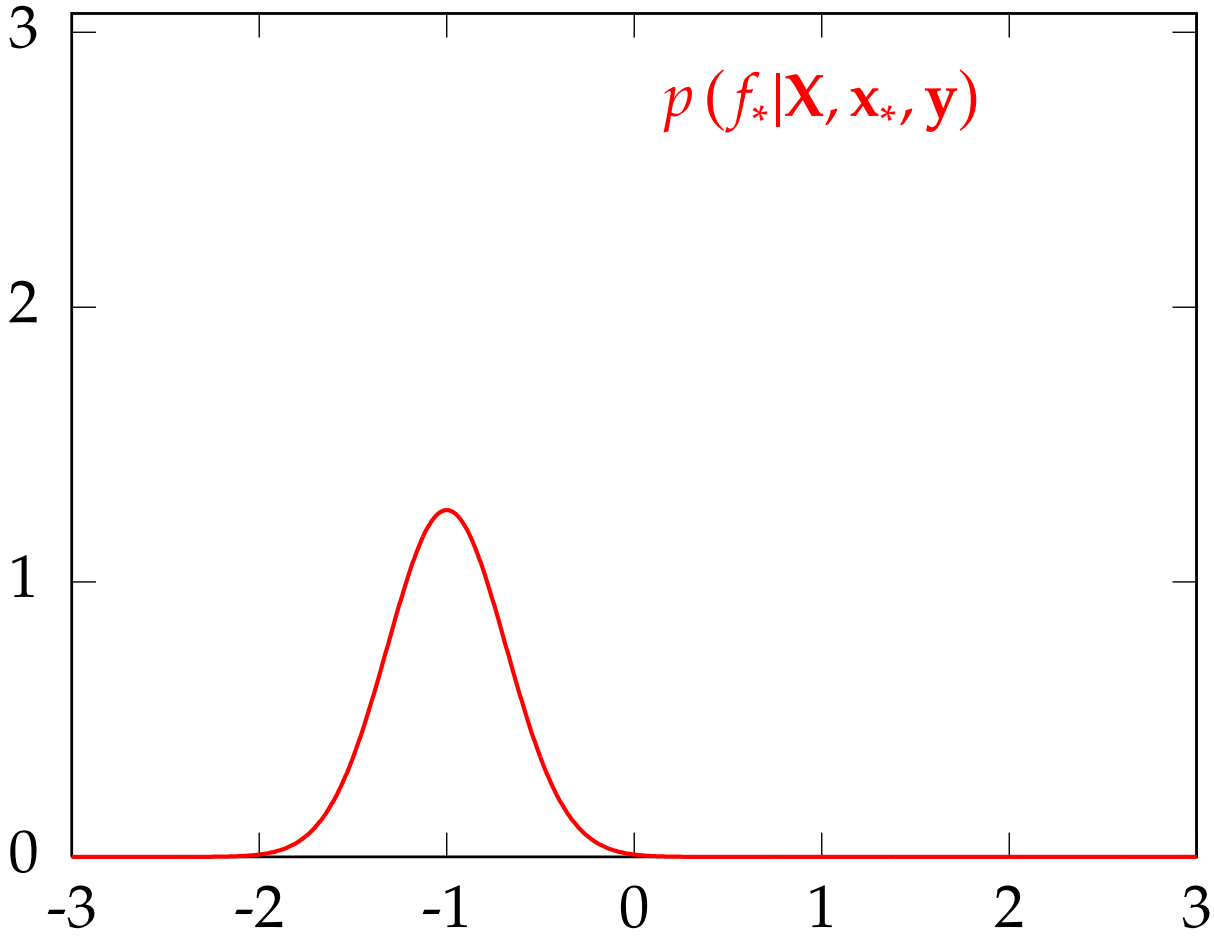


Figure: An EP style update with a classification noise model.

Classification

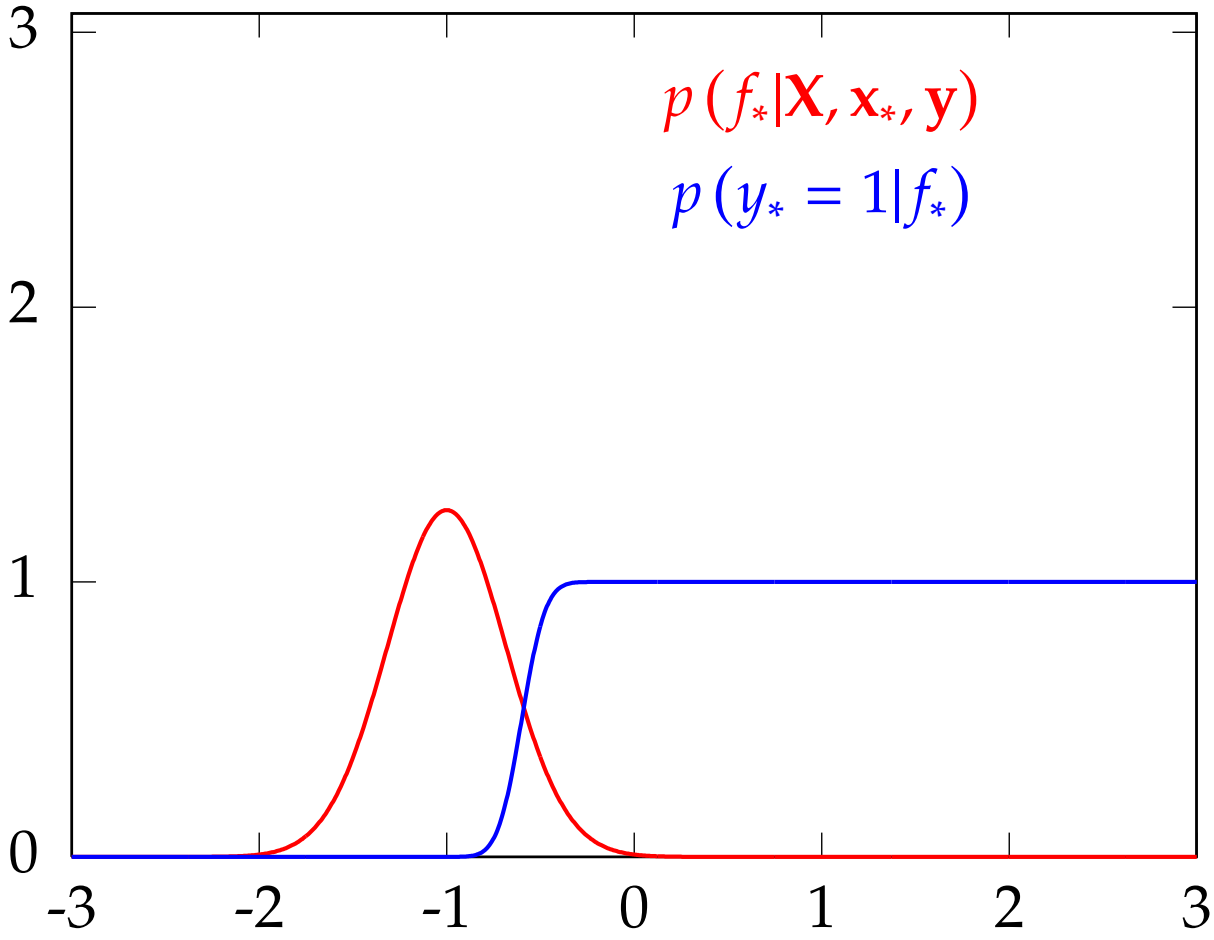


Figure: An EP style update with a classification noise model.

Classification

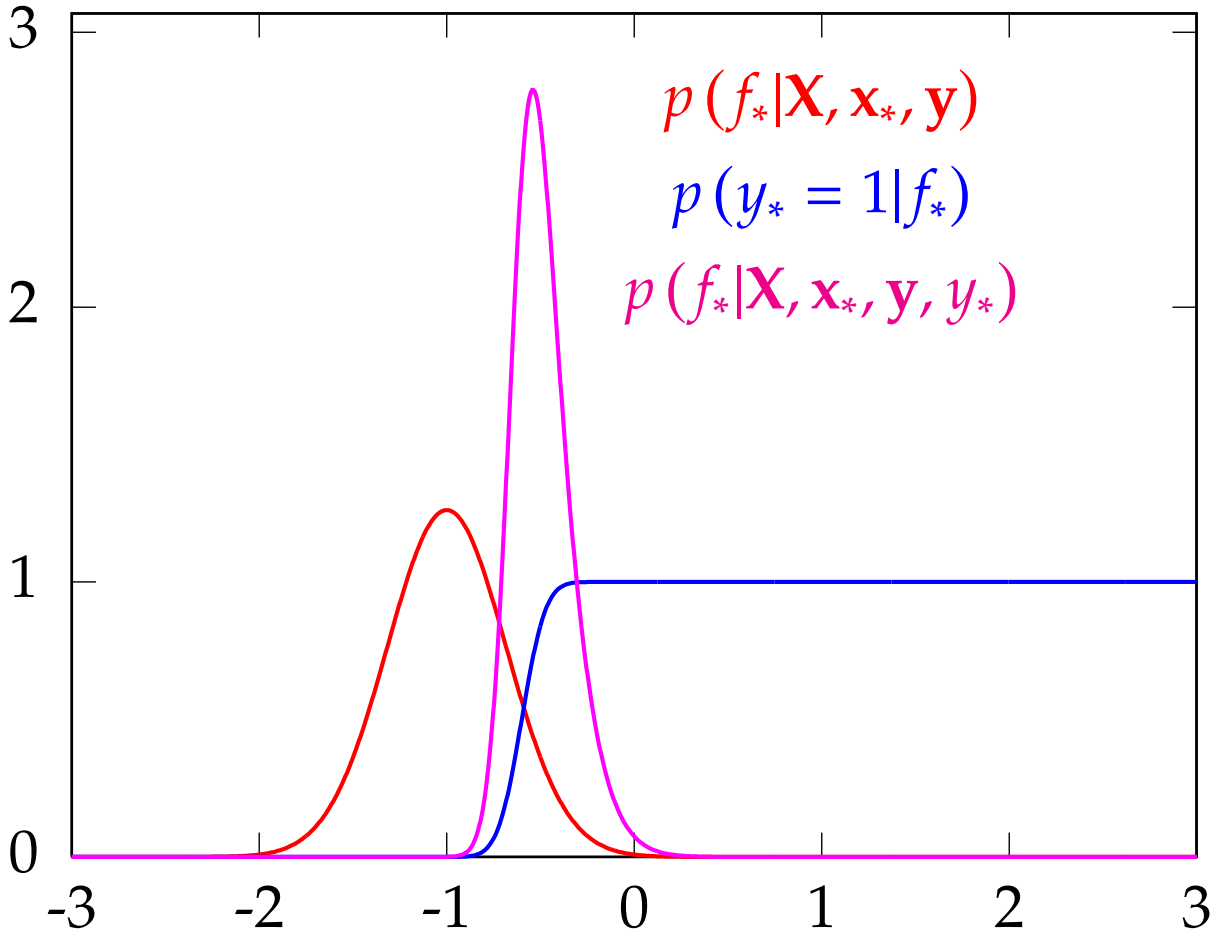


Figure: An EP style update with a classification noise model.

Classification

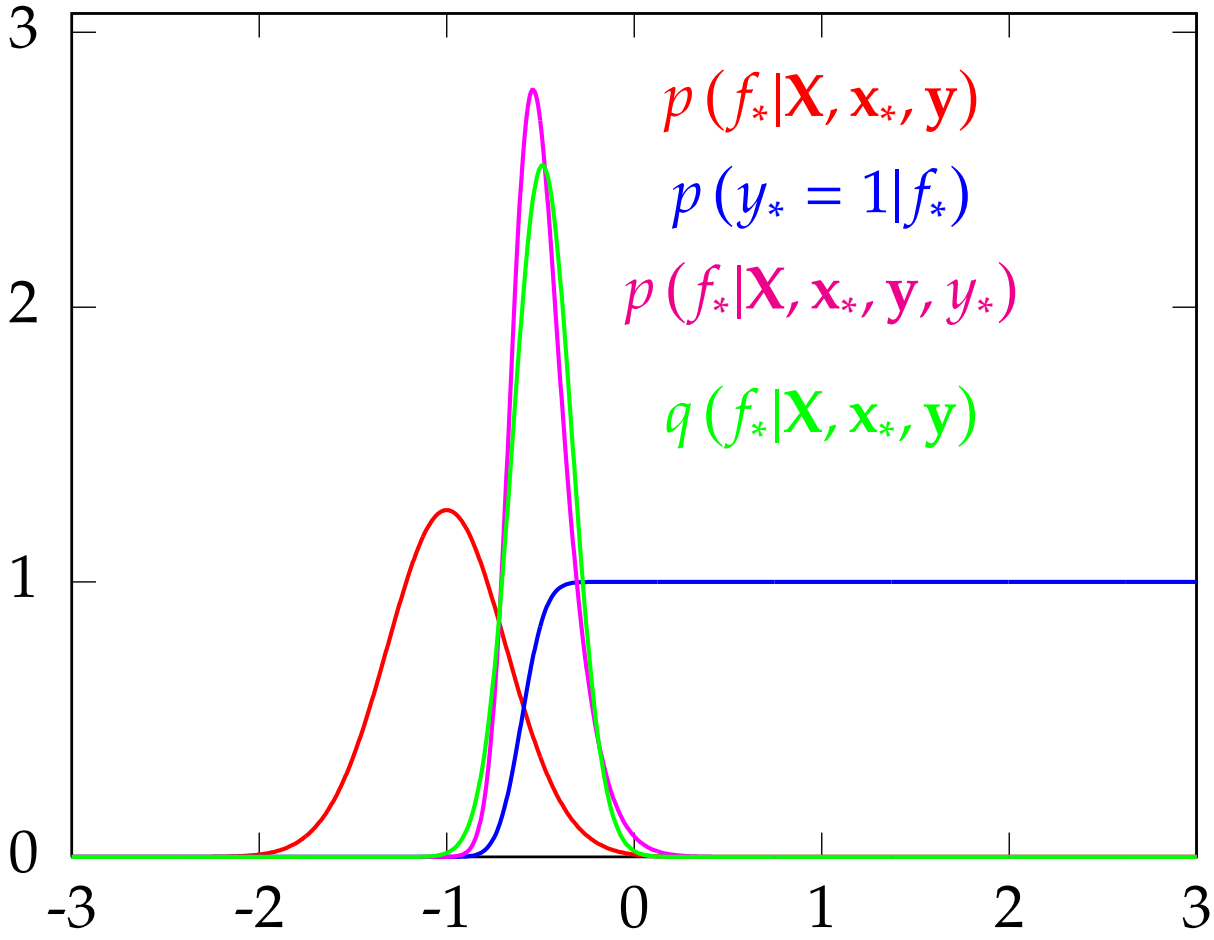


Figure: An EP style update with a classification noise model.

Ordinal Noise Model

Ordered Categories

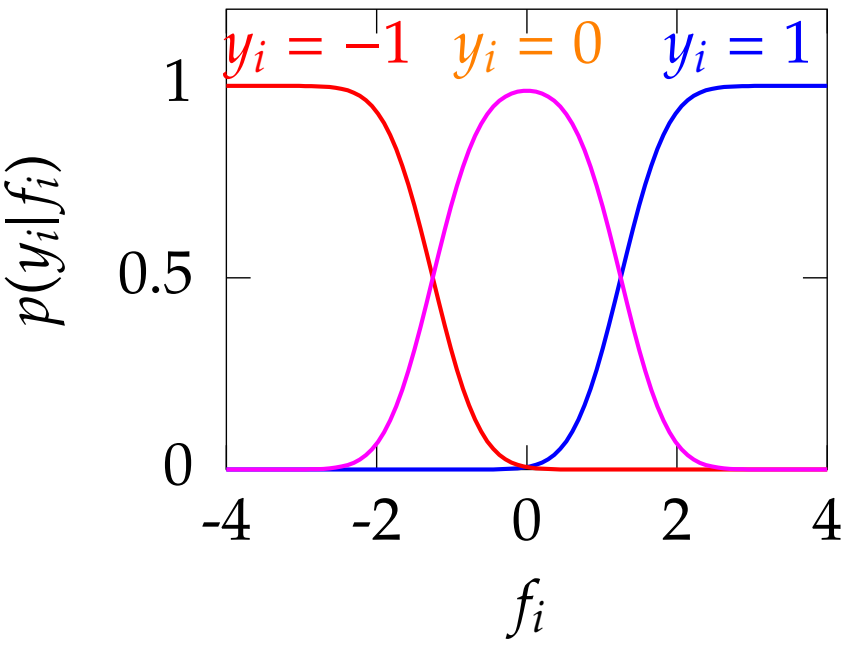


Figure: The ordered categorical noise model (ordinal regression). The plot shows $p(y_i|f_i)$ for different values of y_i. Here we have assumed three categories.

Laplace Approximation

- Equivalent Gaussian is found by making a local 2nd order Taylor approximation at the mode.
- Laplace was the first to suggest this[1], so it's known as the *Laplace approximation*.

Learning Covariance Parameters

Can we determine covariance parameters from the data?

$$\mathcal{N}\left(\mathbf{y}|\mathbf{0},\mathbf{K}\right) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{K}|^{\frac{1}{2}}}\exp\left(-\frac{\mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y}}{2}\right)$$

The parameters are *inside* the covariance function (matrix).

$$k_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$$

Learning Covariance Parameters

Can we determine covariance parameters from the data?

$$\mathcal{N}(\mathbf{y}|\mathbf{0},\mathbf{K}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{K}|^{\frac{1}{2}}}\exp\left(-\frac{\mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y}}{2}\right)$$

The parameters are *inside* the covariance function (matrix).

$$k_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$$

Learning Covariance Parameters

Can we determine covariance parameters from the data?

$$\log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}) = -\frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2} - \frac{n}{2}\log 2\pi$$

The parameters are *inside* the covariance function (matrix).

$$k_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$$

Learning Covariance Parameters

Can we determine covariance parameters from the data?

$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

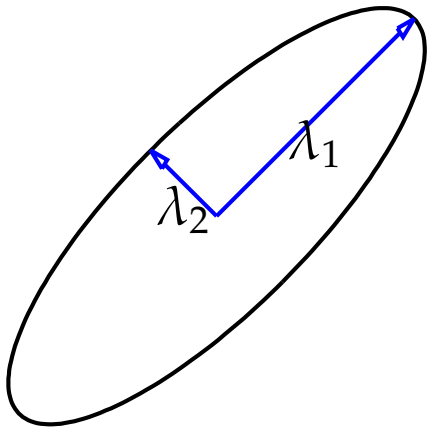
The parameters are *inside* the covariance function (matrix).

$$k_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$$

Eigendecomposition of Covariance

A useful decomposition for understanding the objective function.

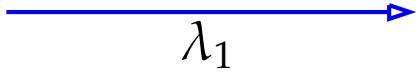
$$\mathbf{K} = \mathbf{R}\mathbf{\Lambda}^2\mathbf{R}^\top$$



Diagonal of $\mathbf{\Lambda}$ represents distance along axes.
$\mathbf{R}$ gives a rotation of these axes.

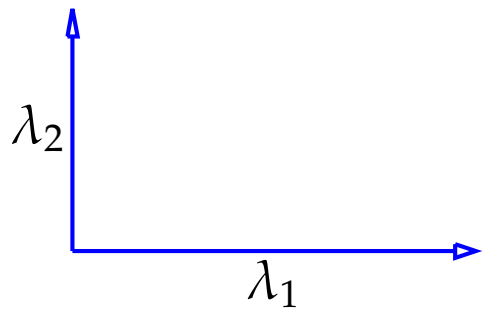
Capacity control: $\log|\mathbf{K}|$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



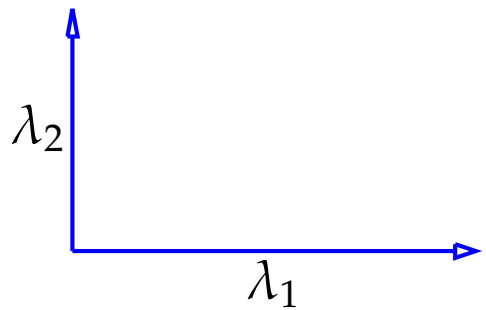
Capacity control: $\log|\mathbf{K}|$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



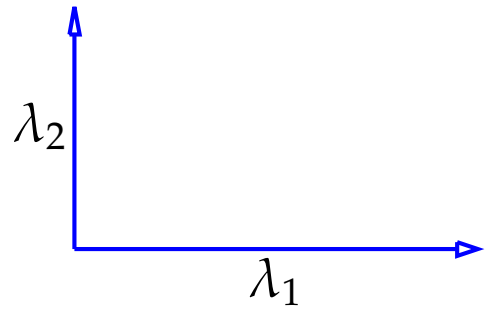
Capacity control: $\log|\mathbf{K}|$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



Capacity control: $\log|\mathbf{K}|$

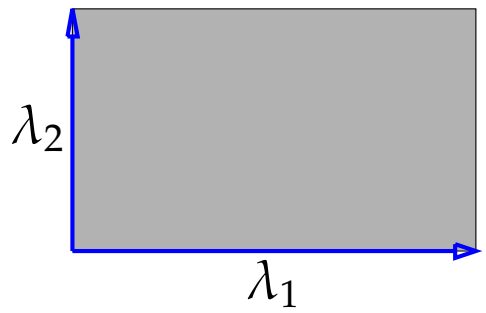
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

Capacity control: $\log|\mathbf{K}|$

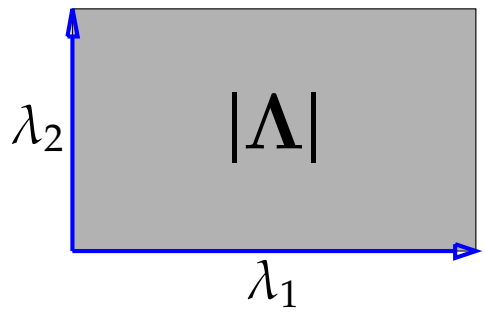
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

Capacity control: $\log|\mathbf{K}|$

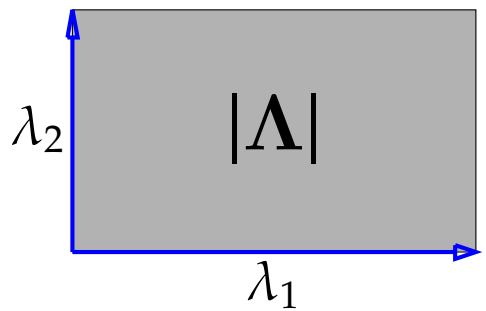
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

Capacity control: $\log|\mathbf{K}|$

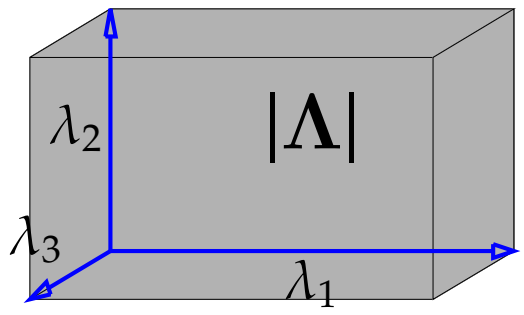
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

Capacity control: $\log |\mathbf{K}|$

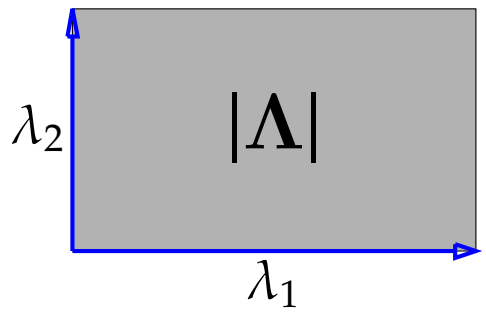
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2 \lambda_3$$

Capacity control: $\log|\mathbf{K}|$

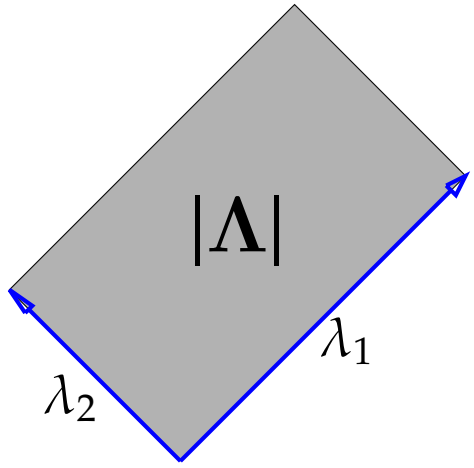
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

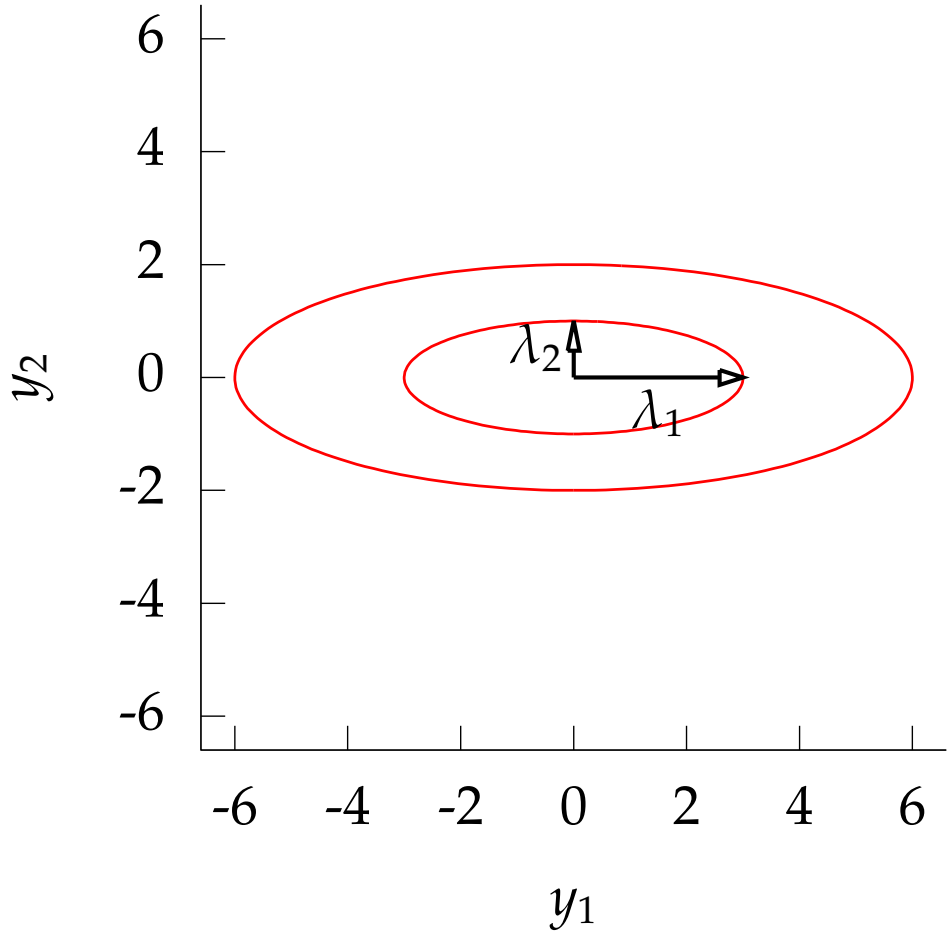
Capacity control: $\log|\mathbf{K}|$

$$\mathbf{R\Lambda} = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix}$$

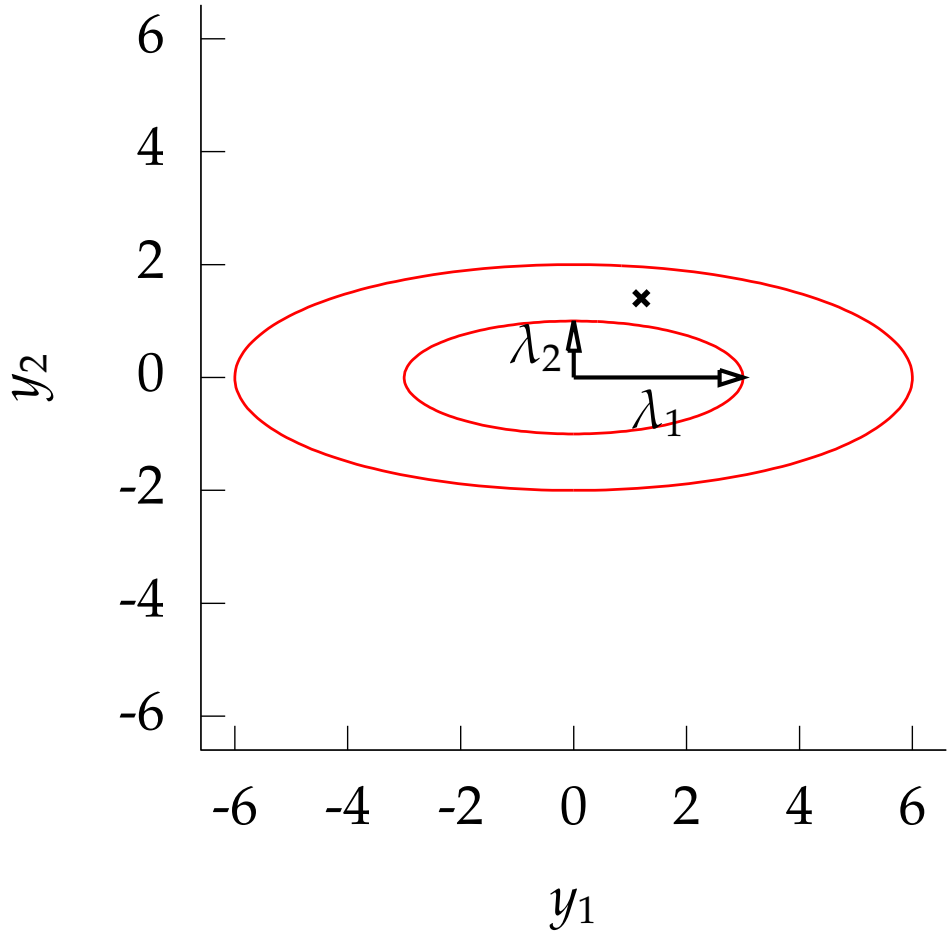


$$|\mathbf{R\Lambda}| = \lambda_1\lambda_2$$

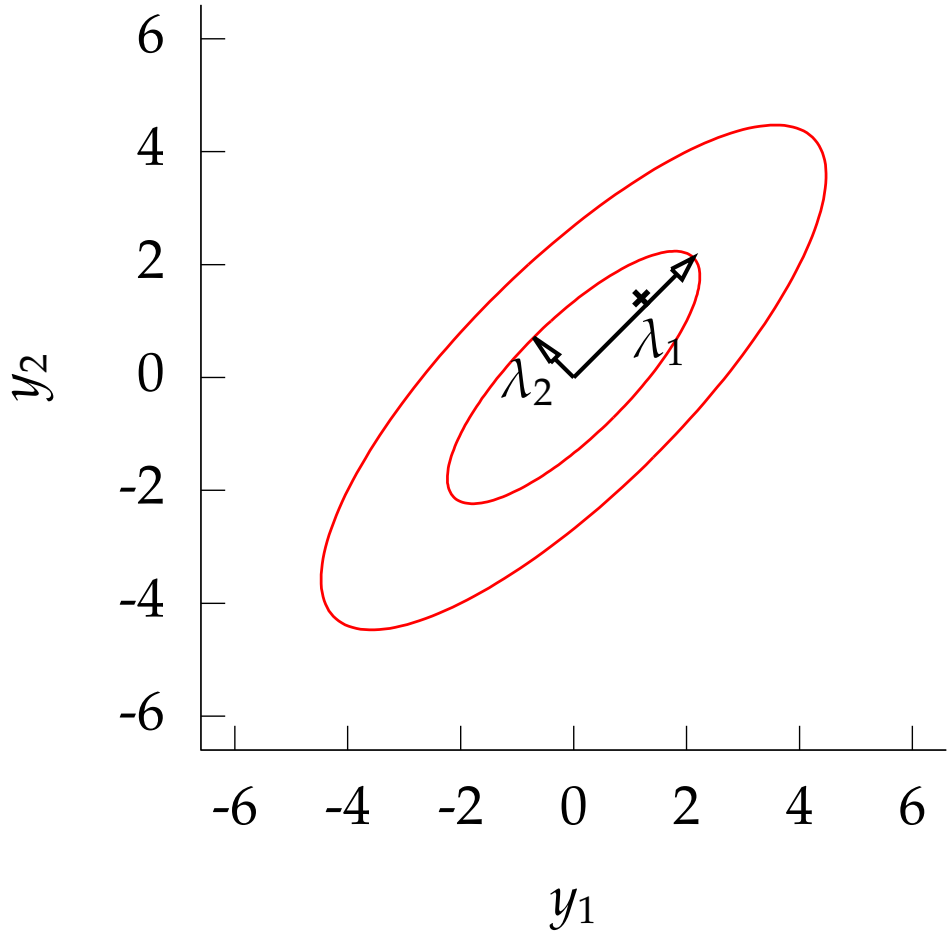
Data Fit: $\frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$



Data Fit: $\frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$

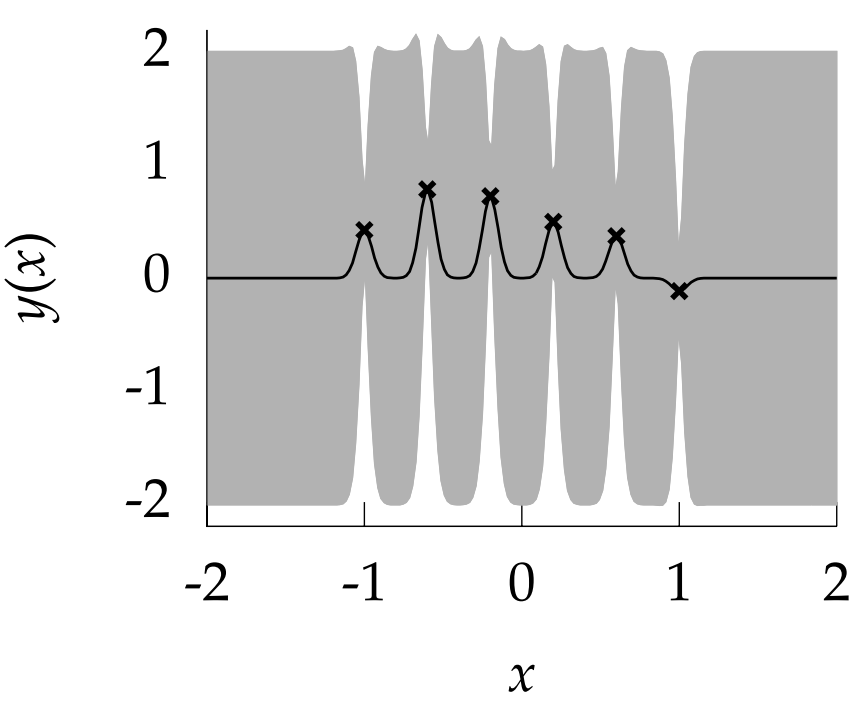


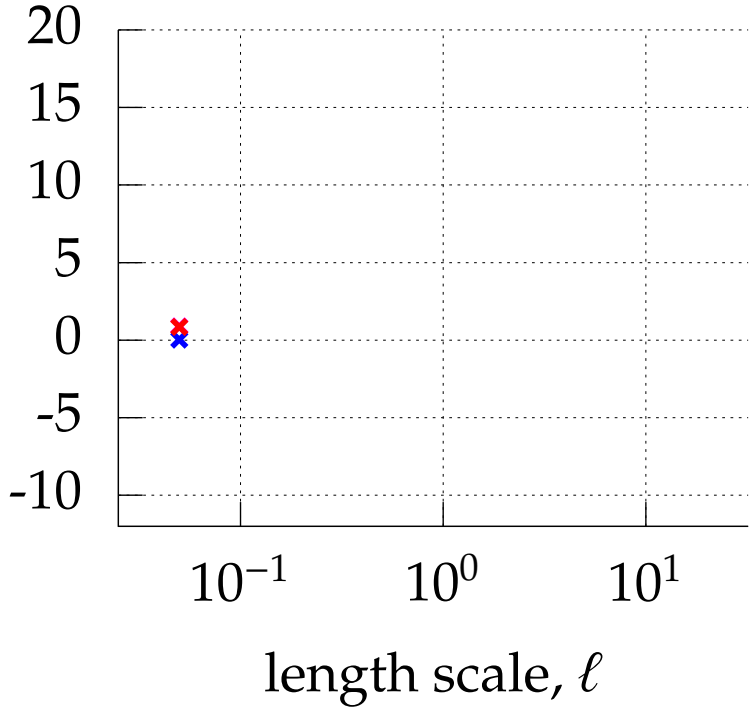
Data Fit: $\frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$



Learning Covariance Parameters

Can we determine length scales and noise levels from the data?

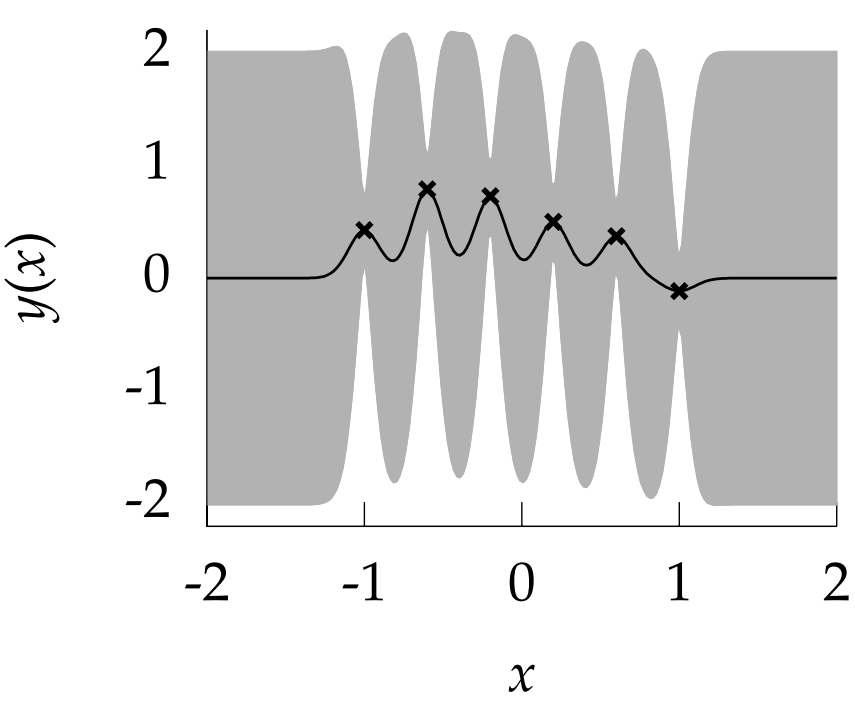


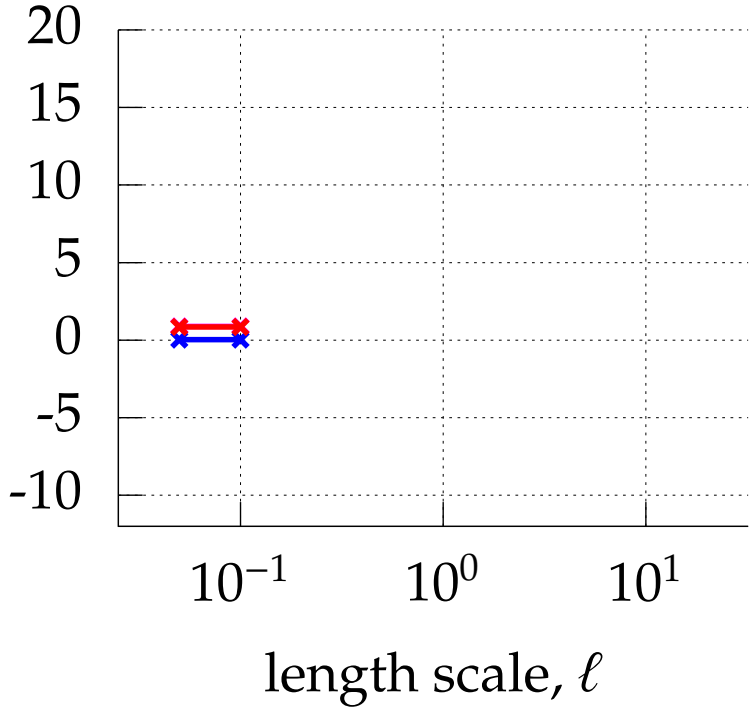


$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

Learning Covariance Parameters

Can we determine length scales and noise levels from the data?

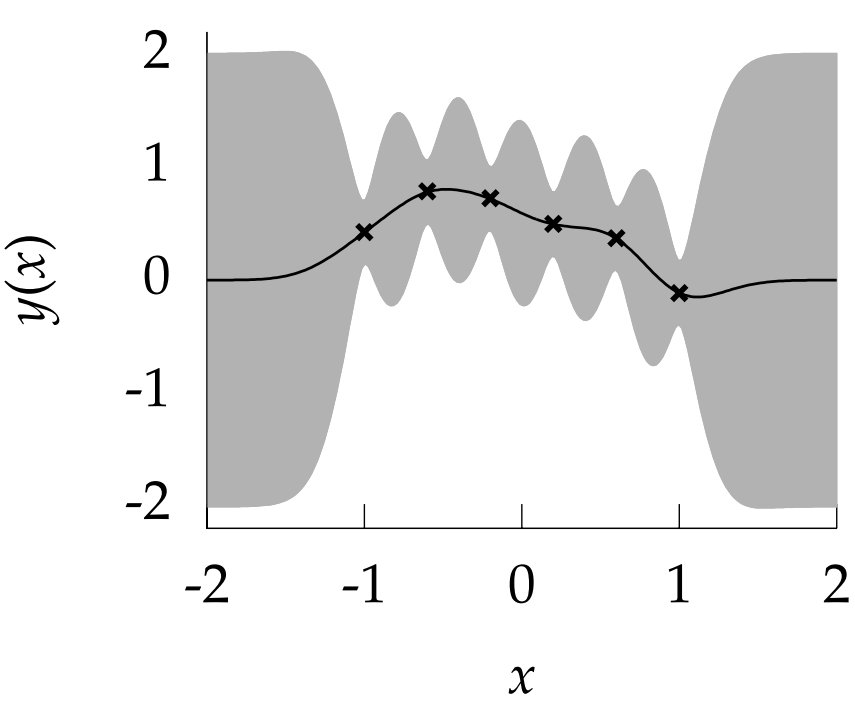


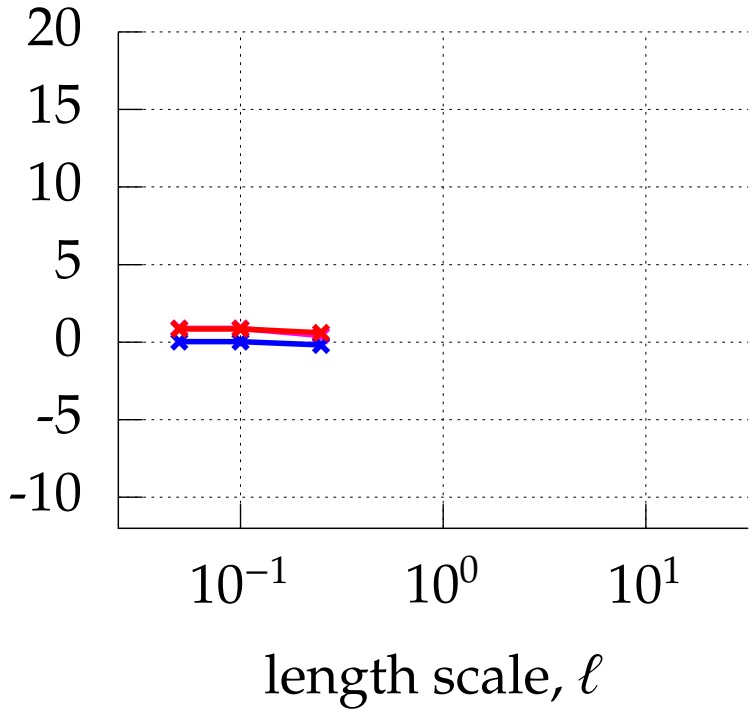


$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

Learning Covariance Parameters

Can we determine length scales and noise levels from the data?

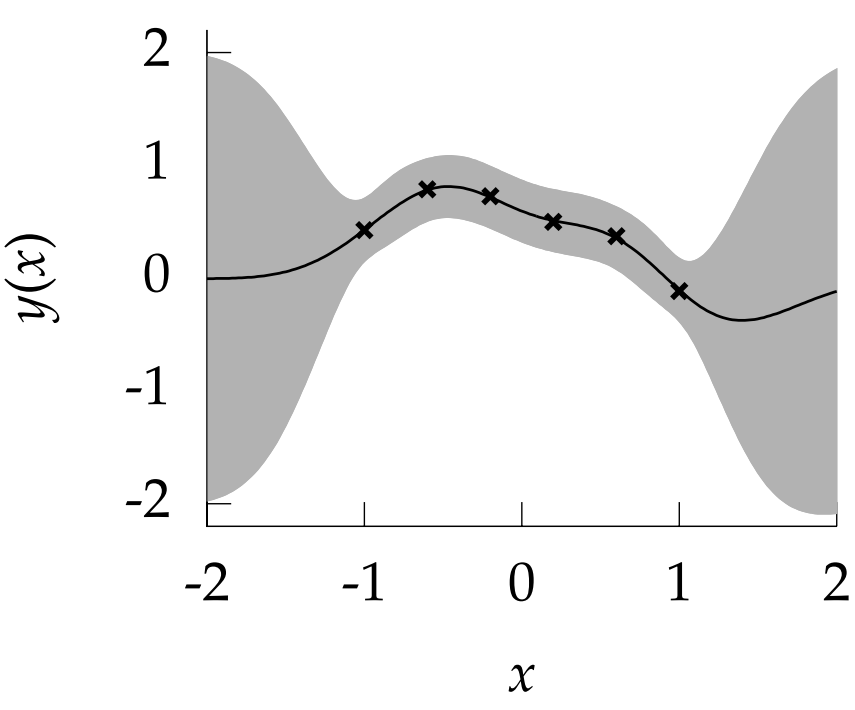


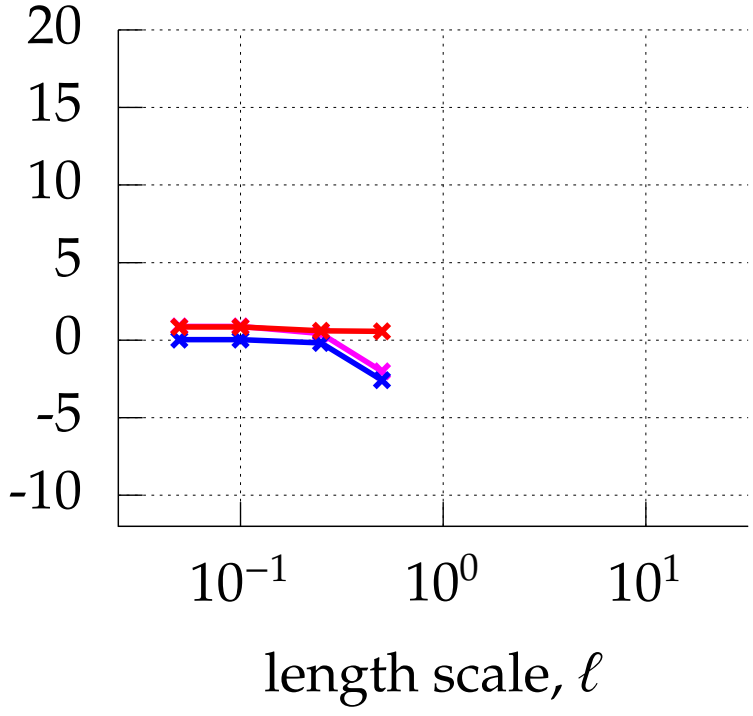


$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

Learning Covariance Parameters

Can we determine length scales and noise levels from the data?

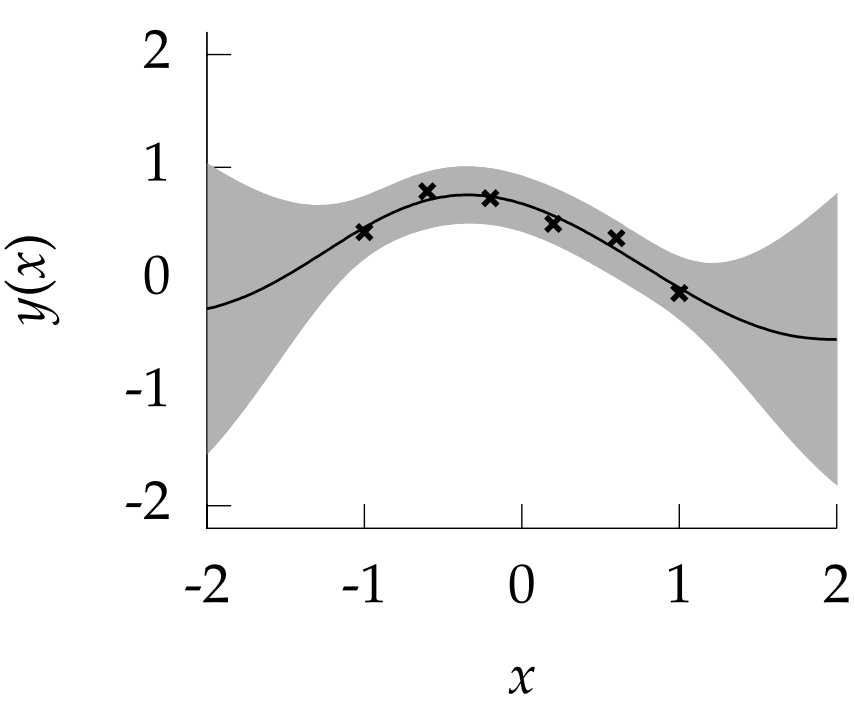


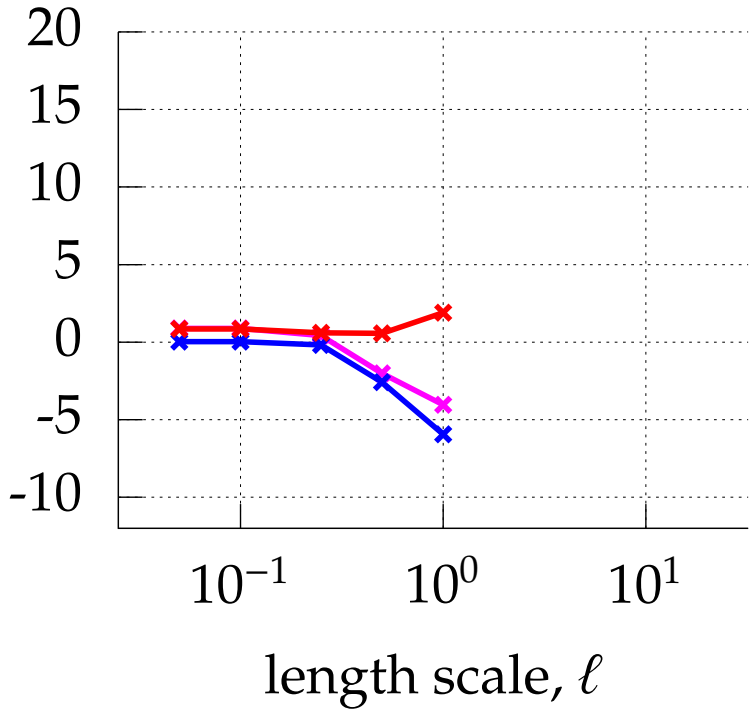


$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

Learning Covariance Parameters

Can we determine length scales and noise levels from the data?

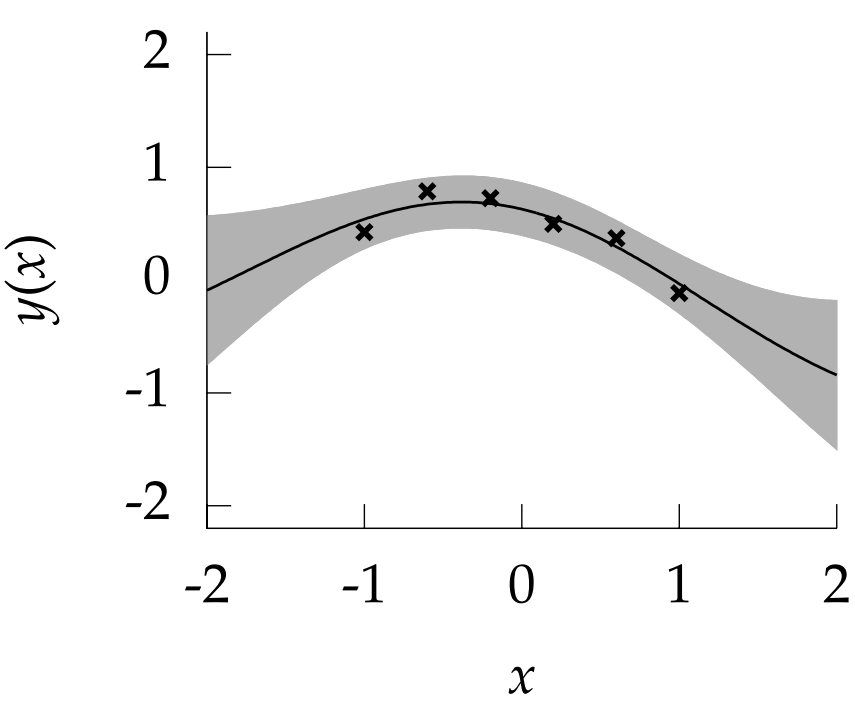


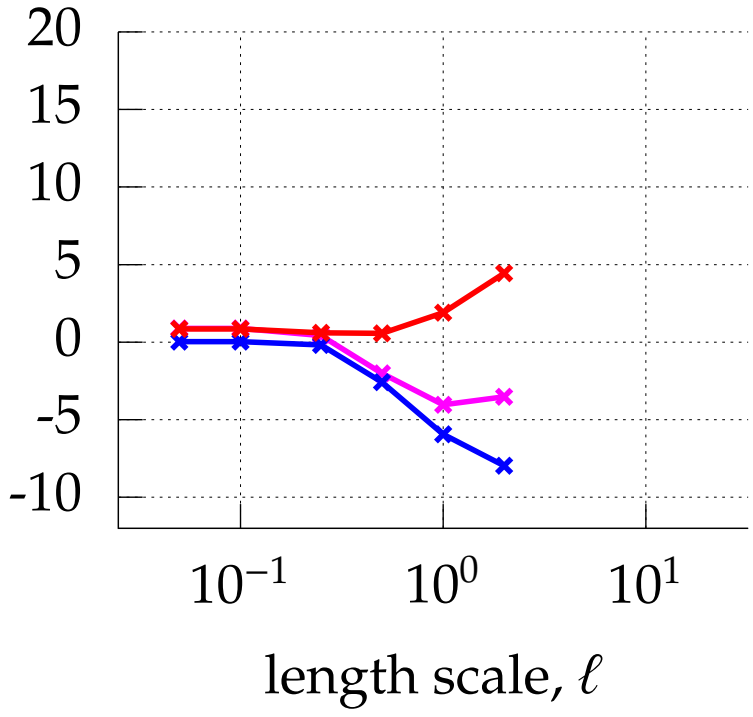


$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

Learning Covariance Parameters

Can we determine length scales and noise levels from the data?

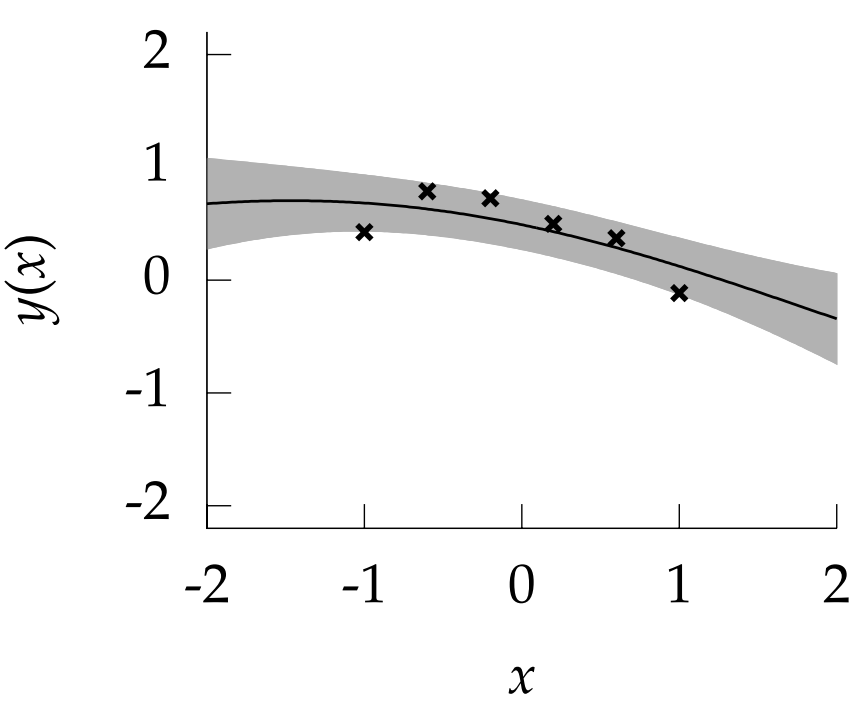


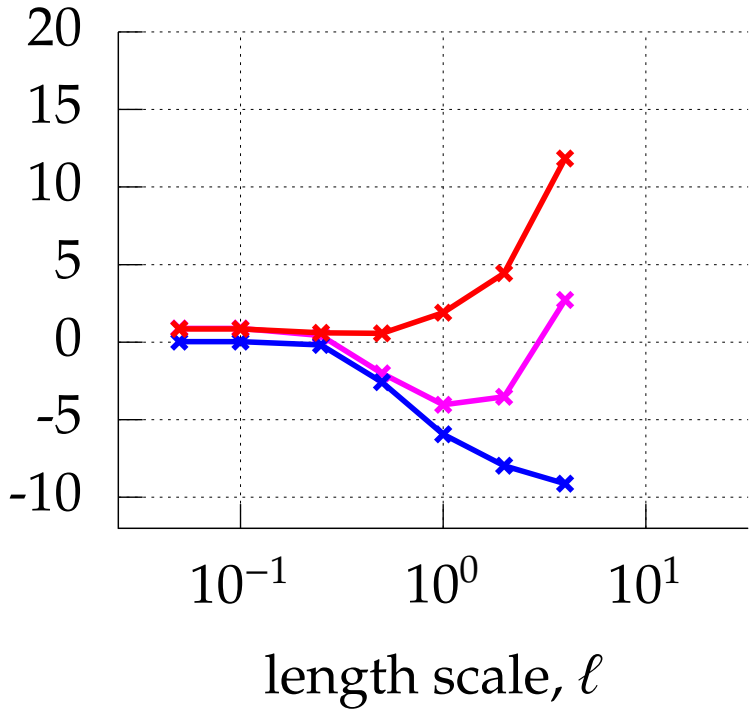


$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

Learning Covariance Parameters

Can we determine length scales and noise levels from the data?

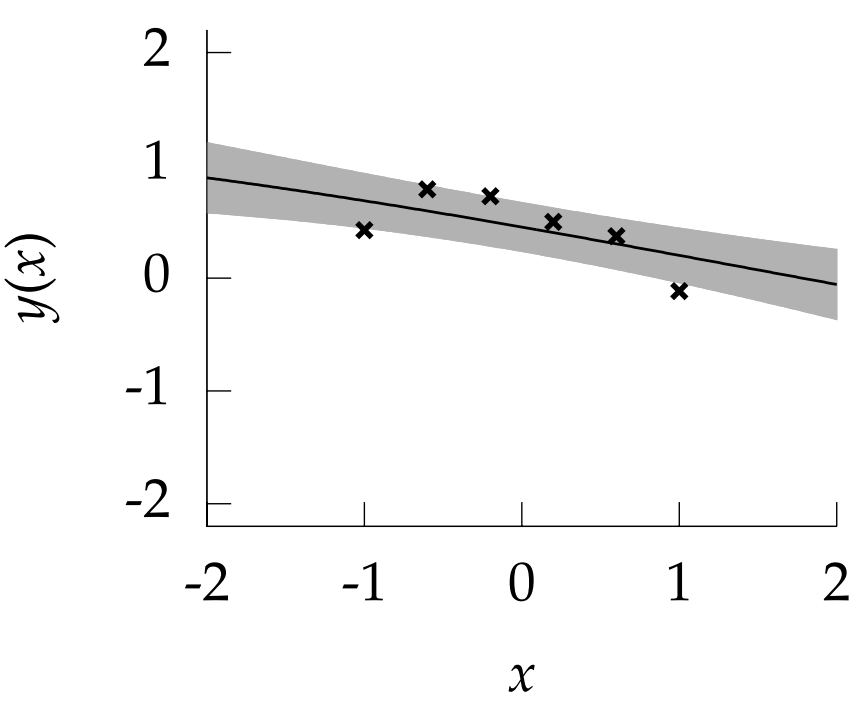


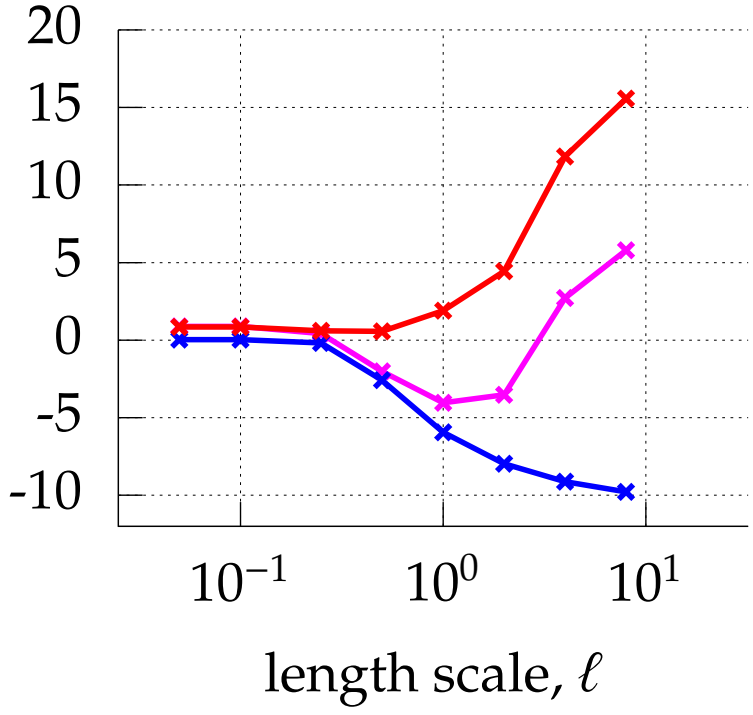


$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

Learning Covariance Parameters

Can we determine length scales and noise levels from the data?

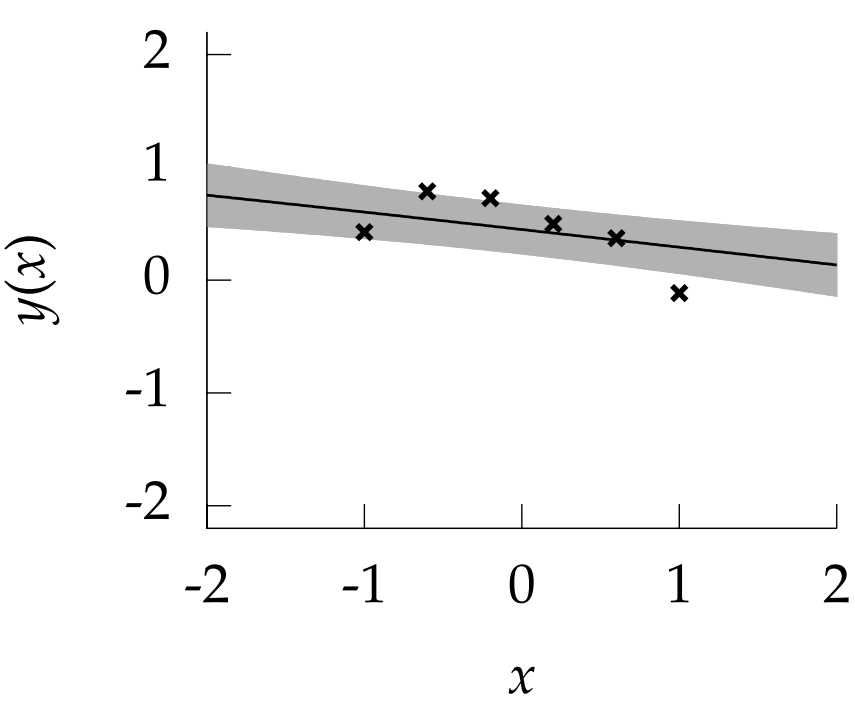


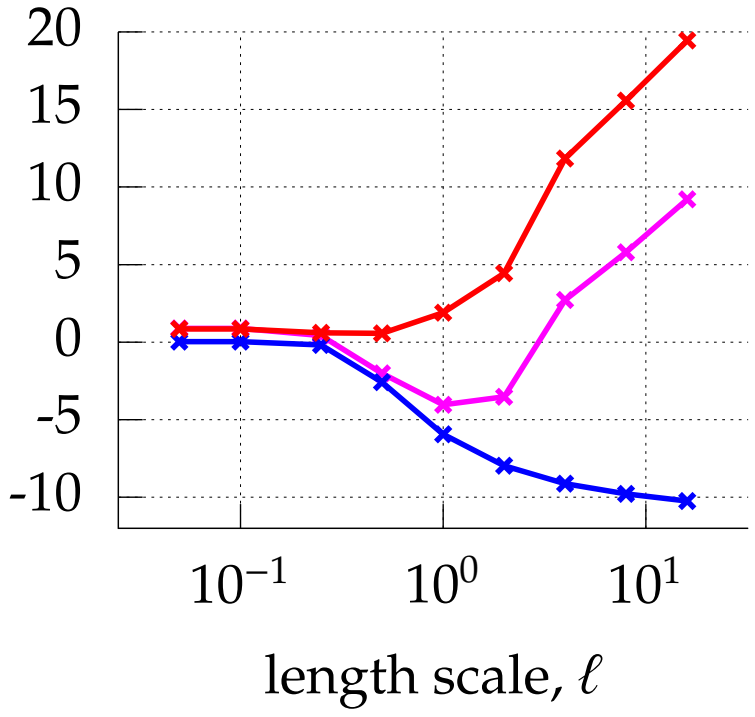


$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

Learning Covariance Parameters

Can we determine length scales and noise levels from the data?





$$E(\boldsymbol{\theta}) = \frac{1}{2}\log|\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}{2}$$

Gene Expression Example

- Given given expression levels in the form of a time series from Della Gatta et al. (2008).
- Want to detect if a gene is expressed or not, fit a GP to each gene (Kalaitzis and Lawrence, 2011).

Kalaitzis and Lawrence *BMC Bioinformatics* 2011, **12**:180
http://www.biomedcentral.com/1471-2105/12/180

BMC Bioinformatics

RESEARCH ARTICLE Open Access

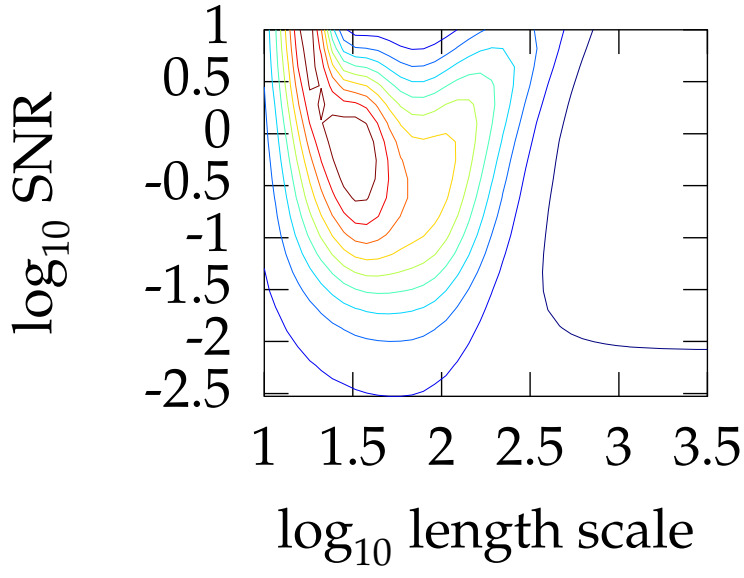
A Simple Approach to Ranking Differentially Expressed Gene Expression Time Courses through Gaussian Process Regression

Alfredo A Kalaitzis* and Neil D Lawrence*

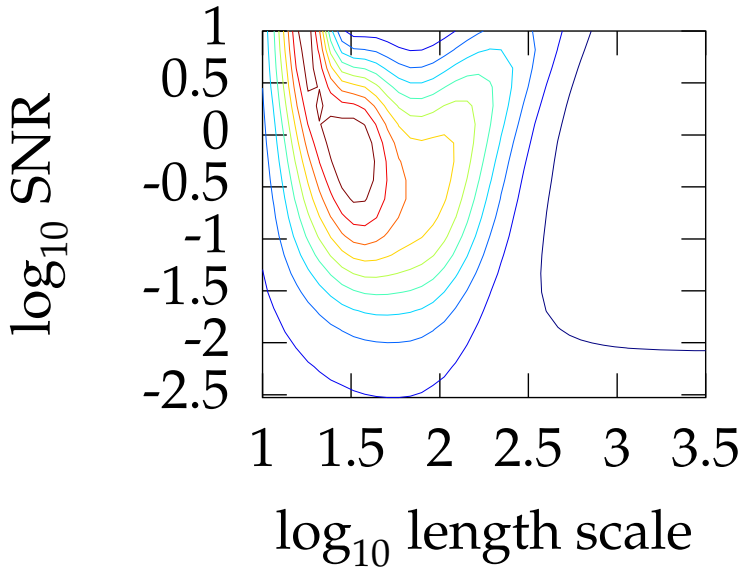
Abstract

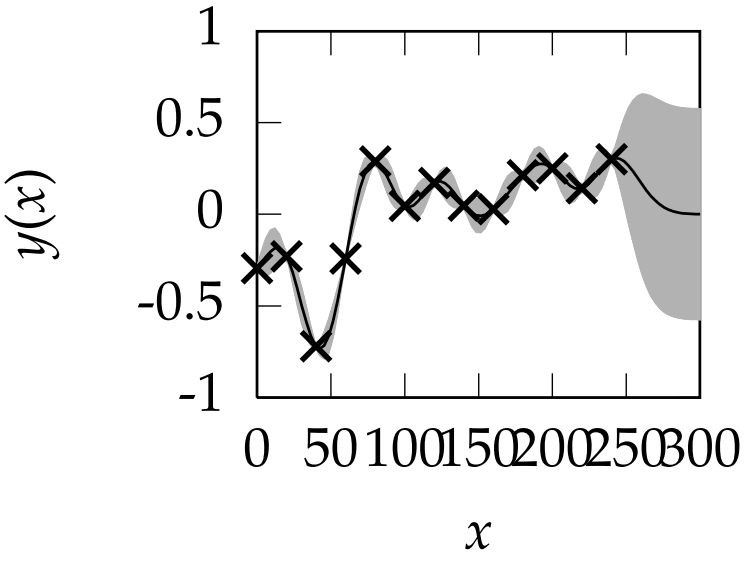
Background: The analysis of gene expression from time series underpins many biological studies. Two basic forms of analysis recur for data of this type: removing inactive (quiet) genes from the study and determining which genes are differentially expressed. Often these analysis stages are applied disregarding the fact that the data is drawn from a time series. In this paper we propose a simple model for accounting for the underlying temporal nature of the data based on a Gaussian process.

Results: We review Gaussian process (GP) regression for estimating the continuous trajectories underlying in gene expression time-series. We present a simple approach which can be used to filter quiet genes, or for the case of time series in the form of expression ratios, quantify differential expression. We assess via ROC curves the rankings produced by our regression framework and compare them to a recently proposed hierarchical Bayesian model for the analysis of gene expression time-series (BATS). We compare on both simulated and experimental data showing that the proposed approach considerably outperforms the current state of the art.

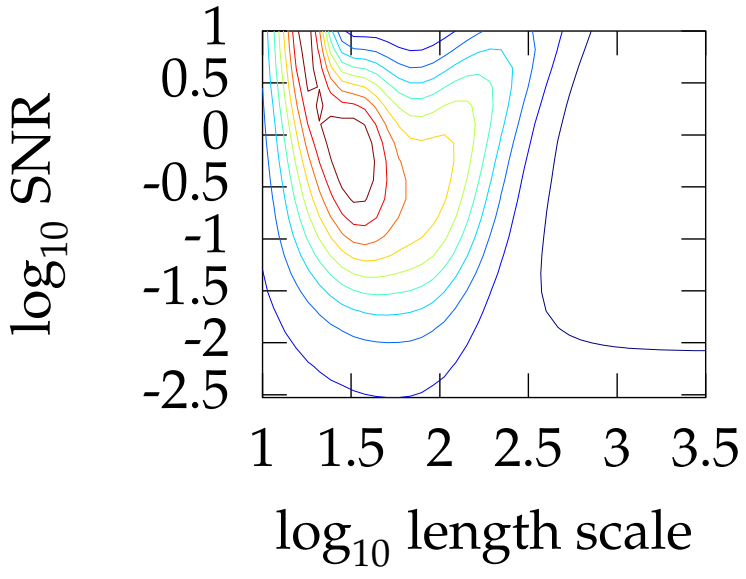


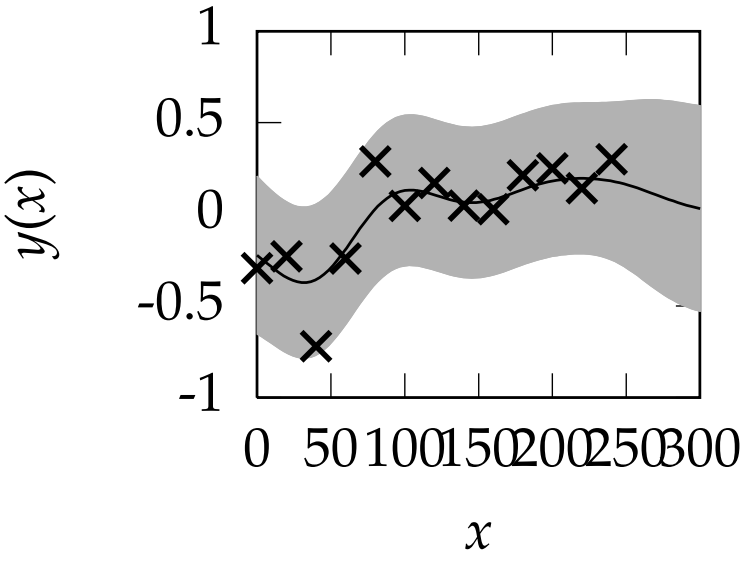
Contour plot of Gaussian process likelihood.



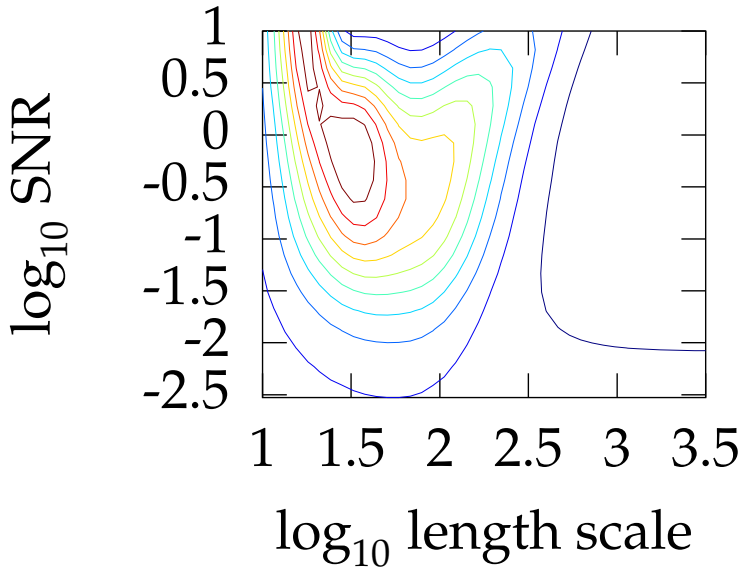


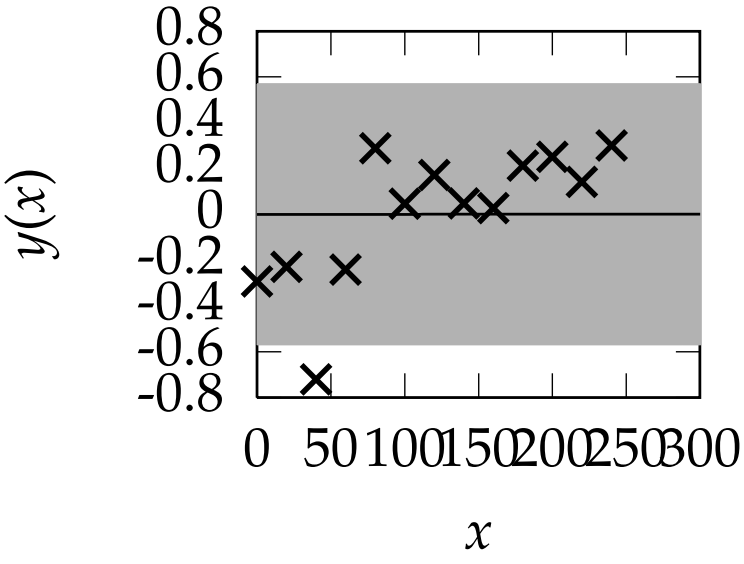
Optima: length scale of 1.2221 and $\log_{10}$ SNR of 1.9654 log likelihood is -0.22317.





Optima: length scale of 1.5162 and $\log_{10}$ SNR of 0.21306 log likelihood is -0.23604.





Optima: length scale of 2.9886 and $\log_{10}$ SNR of -4.506 log likelihood is -2.1056.

Basis Function Form

Radial basis functions commonly have the form

$$\phi_k(\mathbf{x}_i) = \exp\left(-\frac{|\mathbf{x}_i - \boldsymbol{\mu}_k|^2}{2\ell^2}\right).$$

- Basis function maps data into a "feature space" in which a linear sum is a non linear function.

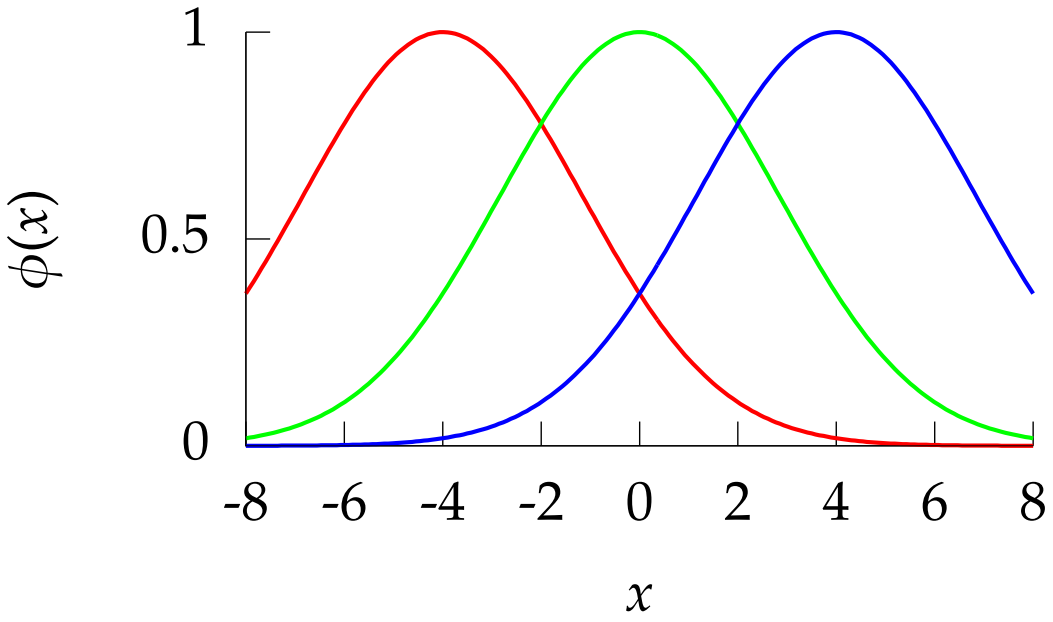


Figure: A set of radial basis functions with width $\ell = 2$ and location parameters $\boldsymbol{\mu} = [-4 \;\; 0 \;\; 4]^\top$.

Basis Function Representations

- Represent a function by a linear sum over a basis,

$$f(\mathbf{x}_{i,:}; \mathbf{w}) = \sum_{k=1}^{m} w_k \phi_k(\mathbf{x}_{i,:}), \tag{1}$$

- Here: m basis functions and $\phi_k(\cdot)$ is kth basis function and

$$\mathbf{w} = [w_1, \ldots, w_m]^\top .$$

- For standard linear model: $\phi_k(\mathbf{x}_{i,:}) = x_{i,k}$.

Random Functions

Functions derived using:

$$f(x) = \sum_{k=1}^{m} w_k \phi_k(x),$$

where elements of $\mathbf{w}$ are independently sampled from a Gaussian density,

$$w_k \sim \mathcal{N}(0, \alpha).$$

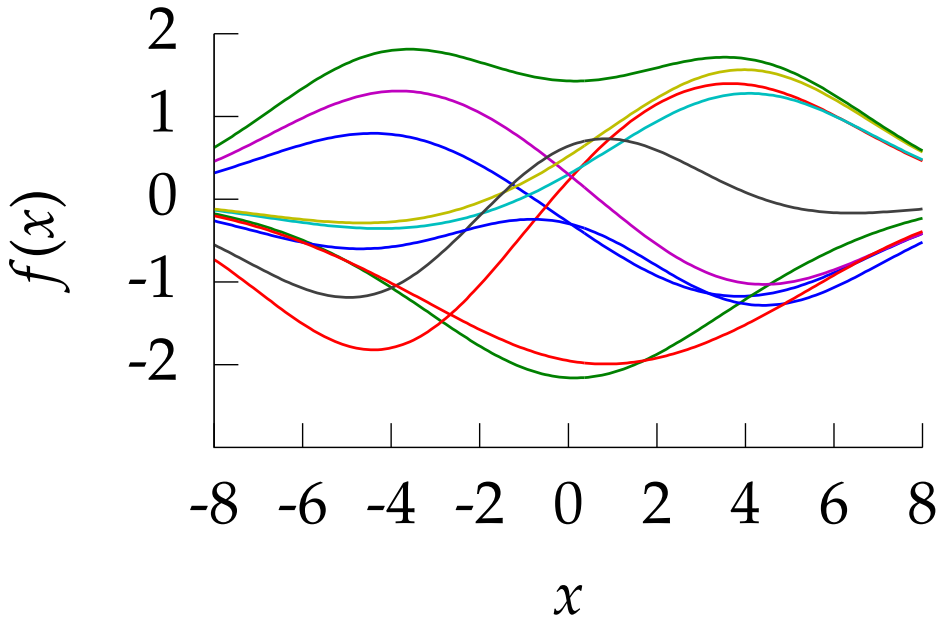


Figure: Functions sampled using the basis set from figure 9. Each line is a separate sample, generated by a weighted sum of the basis set. The weights, $\mathbf{w}$ are sampled from a Gaussian density with variance $\alpha = 1$.

Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

$$wy \sim \mathcal{N}\left(w\mu, w^2\sigma^2\right)$$

Multivariate Consequence

- If

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Multivariate Consequence

- If

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- And

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

Multivariate Consequence

- If
$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
- And
$$\mathbf{y} = \mathbf{W}\mathbf{x}$$
- Then
$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{W}\boldsymbol{\mu}, \mathbf{W}\boldsymbol{\Sigma}\mathbf{W}^{\top}\right)$$

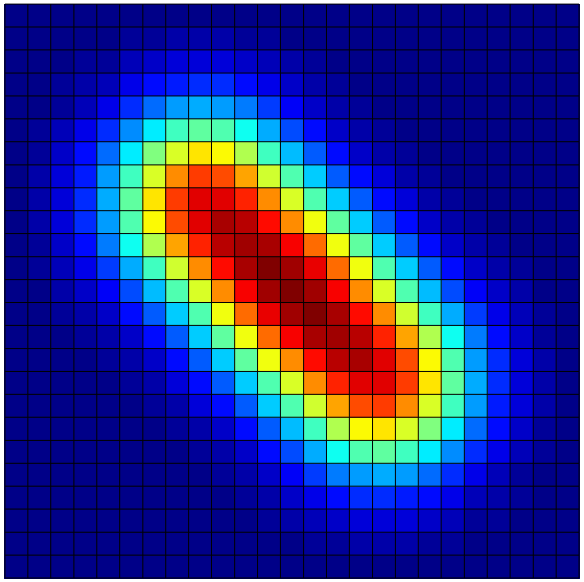
Covariance Functions

RBF Basis Functions

$$k(\mathbf{x}, \mathbf{x}') = \alpha \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$$

$$\phi_k(x) = \exp\left(-\frac{\|x - \mu_k\|_2^2}{\ell^2}\right)$$

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



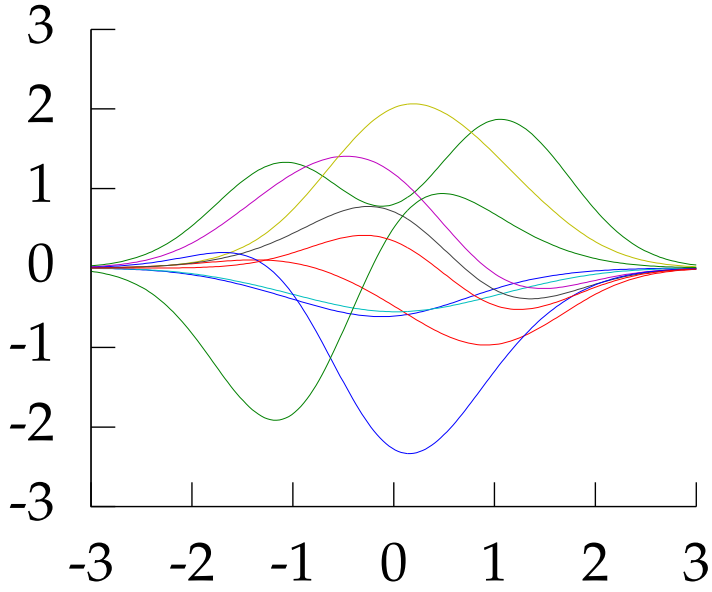
Covariance Functions

RBF Basis Functions

$$k(\mathbf{x}, \mathbf{x}') = \alpha \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$$

$$\phi_k(x) = \exp\left(-\frac{\|x - \mu_k\|_2^2}{\ell^2}\right)$$

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

$$\sum_{i=1}^{n} y_i \sim \mathcal{N}\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

$$wy \sim \mathcal{N}\left(w\mu, w^2\sigma^2\right)$$

Multivariate Consequence

- If

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Multivariate Consequence

- If

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- And

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

Multivariate Consequence

- If
$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
- And
$$\mathbf{y} = \mathbf{W}\mathbf{x}$$
- Then
$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{W}\boldsymbol{\mu}, \mathbf{W}\boldsymbol{\Sigma}\mathbf{W}^{\top}\right)$$

Selecting Number and Location of Basis

- Need to choose
  1. location of centers
  2. number of basis functions

  Restrict analysis to 1-D input, x.
- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \boldsymbol{\phi}_k(x_i)^\top \boldsymbol{\phi}_k(x_j)$$

Selecting Number and Location of Basis

- Need to choose
  1. location of centers
  2. number of basis functions

  Restrict analysis to 1-D input, x.
- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \sum_{k=1}^{m} \phi_k(x_i)\phi_k(x_j)$$

Selecting Number and Location of Basis

- Need to choose
  1. location of centers
  2. number of basis functions

  Restrict analysis to 1-D input, x.
- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \sum_{k=1}^{m} \exp\left(-\frac{(x_i - \mu_k)^2}{2\ell^2}\right) \exp\left(-\frac{(x_j - \mu_k)^2}{2\ell^2}\right)$$

Selecting Number and Location of Basis

- Need to choose
  1. location of centers
  2. number of basis functions

  Restrict analysis to 1-D input, x.
- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \sum_{k=1}^{m} \exp\left(-\frac{(x_i - \mu_k)^2}{2\ell^2} - \frac{(x_j - \mu_k)^2}{2\ell^2}\right)$$

Selecting Number and Location of Basis

- Need to choose
  1. location of centers
  2. number of basis functions

  Restrict analysis to 1-D input, x.
- Consider uniform spacing over a region:

$$k\left(x_i, x_j\right) = \alpha \sum_{k=1}^{m} \exp\left(-\frac{x_i^2 + x_j^2 - 2\mu_k\left(x_i + x_j\right) + 2\mu_k^2}{2\ell^2}\right),$$

Uniform Basis Functions

- Set each center location to

$$\mu_k = a + \Delta\mu \cdot (k - 1).$$

Uniform Basis Functions

- Set each center location to

$$\mu_k = a + \Delta\mu \cdot (k-1).$$

- Specify the basis functions in terms of their indices,

$$k\left(x_i, x_j\right) = \alpha' \Delta\mu \sum_{k=1}^{m} \exp\left(-\frac{x_i^2 + x_j^2}{2\ell^2} - \frac{2\left(a + \Delta\mu \cdot (k-1)\right)\left(x_i + x_j\right) + 2\left(a + \Delta\mu \cdot (k-1)\right)^2}{2\ell^2} \right).$$

Uniform Basis Functions

- Set each center location to

$$\mu_k = a + \Delta\mu \cdot (k-1).$$

- Specify the basis functions in terms of their indices,

$$k\left(x_i, x_j\right) = \alpha' \Delta\mu \sum_{k=1}^{m} \exp\left(-\frac{x_i^2 + x_j^2}{2\ell^2} - \frac{2\left(a + \Delta\mu \cdot (k-1)\right)\left(x_i + x_j\right) + 2\left(a + \Delta\mu \cdot (k-1)\right)^2}{2\ell^2} \right).$$

- Here we've scaled variance of process by $\Delta\mu$.

Infinite Basis Functions

- Take

$$\mu_1 = a \text{ and } \mu_m = b \text{ so } b = a + \Delta\mu \cdot (m - 1)$$

Infinite Basis Functions

- Take
$$\mu_1 = a \text{ and } \mu_m = b \text{ so } b = a + \Delta\mu \cdot (m-1)$$

- This implies
$$b - a = \Delta\mu(m-1)$$

Infinite Basis Functions

- Take
$$\mu_1 = a \text{ and } \mu_m = b \text{ so } b = a + \Delta\mu \cdot (m-1)$$

- This implies
$$b - a = \Delta\mu(m-1)$$
and therefore
$$m = \frac{b-a}{\Delta\mu} + 1$$

Infinite Basis Functions

- Take
$$\mu_1 = a \text{ and } \mu_m = b \text{ so } b = a + \Delta\mu \cdot (m-1)$$

- This implies
$$b - a = \Delta\mu(m-1)$$
and therefore
$$m = \frac{b-a}{\Delta\mu} + 1$$

- Take limit as $\Delta\mu \to 0$ so $m \to \infty$

Infinite Basis Functions

- Take
$$\mu_1 = a \text{ and } \mu_m = b \text{ so } b = a + \Delta\mu \cdot (m-1)$$

- This implies
$$b - a = \Delta\mu(m-1)$$
and therefore
$$m = \frac{b-a}{\Delta\mu} + 1$$

- Take limit as $\Delta\mu \to 0$ so $m \to \infty$

$$k(x_i, x_j) = \alpha' \int_a^b \exp\left(-\frac{x_i^2 + x_j^2}{2\ell^2} + \frac{2\left(\mu - \frac{1}{2}\left(x_i + x_j\right)\right)^2 - \frac{1}{2}\left(x_i + x_j\right)^2}{2\ell^2}\right) \mathrm{d}\mu,$$

where we have used $a + k \cdot \Delta\mu \to \mu$.

Result

- Performing the integration leads to

$$k(x_i,x_j) = \alpha' \sqrt{\pi\ell^2} \exp\left(-\frac{\left(x_i - x_j\right)^2}{4\ell^2}\right)$$
$$\times \frac{1}{2}\left[\operatorname{erf}\left(\frac{\left(b - \frac{1}{2}\left(x_i + x_j\right)\right)}{\ell}\right) - \operatorname{erf}\left(\frac{\left(a - \frac{1}{2}\left(x_i + x_j\right)\right)}{\ell}\right)\right],$$

Result

- Performing the integration leads to

$$k(x_i,x_j) = \alpha' \sqrt{\pi\ell^2} \exp\left(-\frac{\left(x_i - x_j\right)^2}{4\ell^2}\right) \times \frac{1}{2}\left[\operatorname{erf}\left(\frac{\left(b - \frac{1}{2}\left(x_i + x_j\right)\right)}{\ell}\right) - \operatorname{erf}\left(\frac{\left(a - \frac{1}{2}\left(x_i + x_j\right)\right)}{\ell}\right)\right],$$

- Now take limit as $a \to -\infty$ and $b \to \infty$

Result

- Performing the integration leads to

$$k(x_i,x_j) = \alpha' \sqrt{\pi\ell^2} \exp\left(-\frac{\left(x_i - x_j\right)^2}{4\ell^2}\right) \times \frac{1}{2}\left[\operatorname{erf}\left(\frac{\left(b - \frac{1}{2}\left(x_i + x_j\right)\right)}{\ell}\right) - \operatorname{erf}\left(\frac{\left(a - \frac{1}{2}\left(x_i + x_j\right)\right)}{\ell}\right)\right],$$

- Now take limit as $a \to -\infty$ and $b \to \infty$

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left(x_i - x_j\right)^2}{4\ell^2}\right).$$

where $\alpha = \alpha' \sqrt{\pi\ell^2}$.

Infinite Feature Space

- An RBF model with infinite basis functions is a Gaussian process.

Infinite Feature Space

- An RBF model with infinite basis functions is a Gaussian process.
- The covariance function is given by the exponentiated quadratic covariance function.

$$k\left(x_i, x_j\right) = \alpha \exp\left(-\frac{\left(x_i - x_j\right)^2}{4\ell^2}\right).$$

Infinite Feature Space

- An RBF model with infinite basis functions is a Gaussian process.
- The covariance function is the exponentiated quadratic.
- **Note:** The functional form for the covariance function and basis functions are similar.
  - this is a special case,
  - in general they are very different

Similar results can obtained for multi-dimensional input models Williams (1998); Neal (1996).

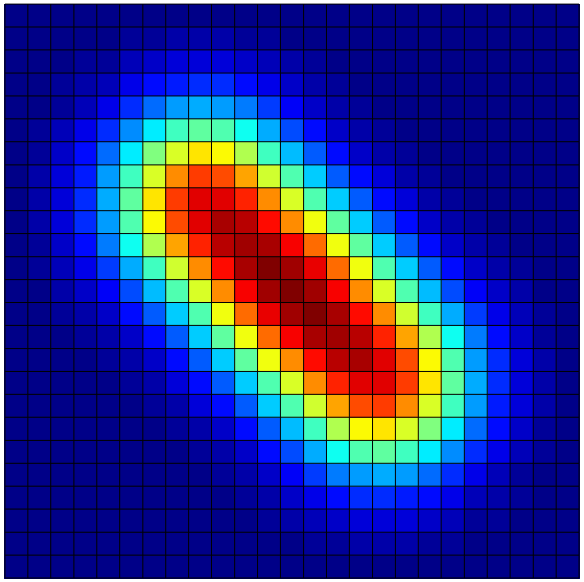
Covariance Functions

RBF Basis Functions

$$k(\mathbf{x}, \mathbf{x}') = \alpha \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$$

$$\phi_k(x) = \exp\left(-\frac{\|x - \mu_k\|_2^2}{\ell^2}\right)$$

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



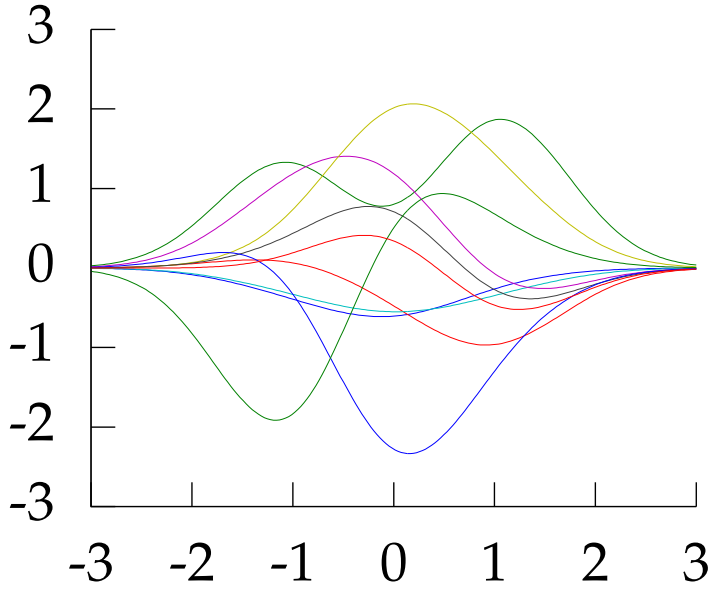
Covariance Functions

RBF Basis Functions

$$k(\mathbf{x}, \mathbf{x}') = \alpha \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$$

$$\phi_k(x) = \exp\left(-\frac{\|x - \mu_k\|_2^2}{\ell^2}\right)$$

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$



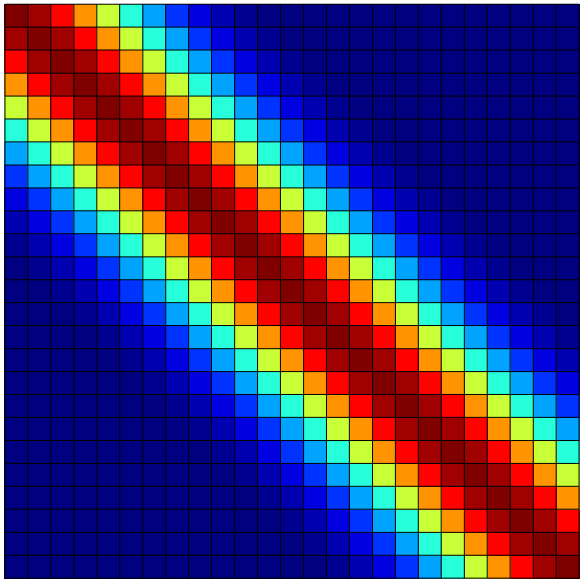
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



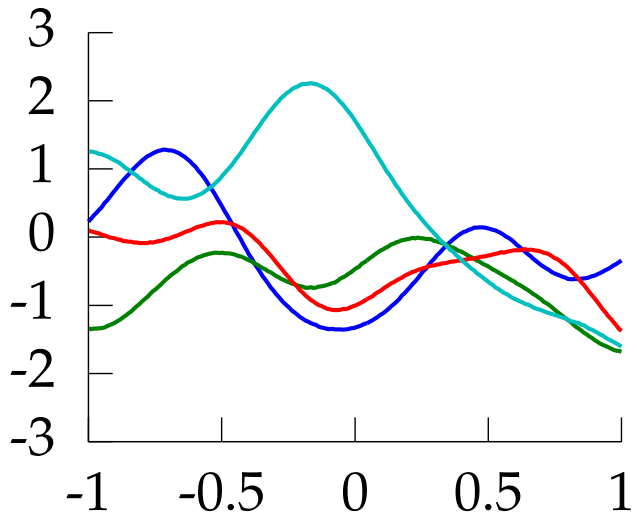
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



Covariance Functions

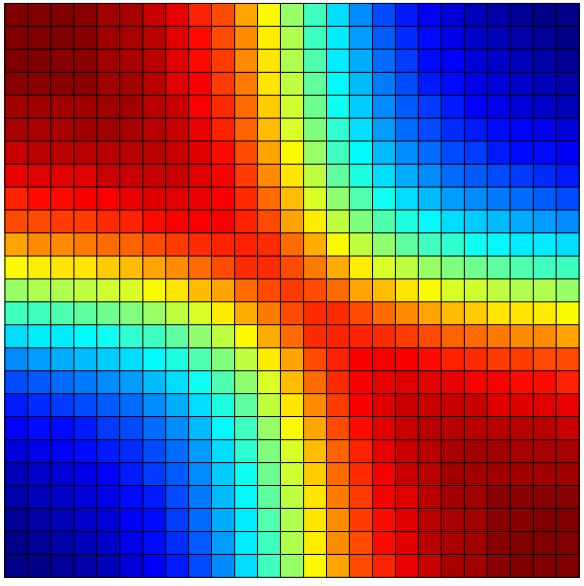
MLP Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha \text{asin}\left(\frac{w\mathbf{x}^\top\mathbf{x}' + b}{\sqrt{w\mathbf{x}^\top\mathbf{x} + b + 1}\sqrt{w\mathbf{x}'^\top\mathbf{x}' + b + 1}}\right)$$

- Based on infinite neural network model.

$$w = 40$$

$$b = 4$$



Covariance Functions

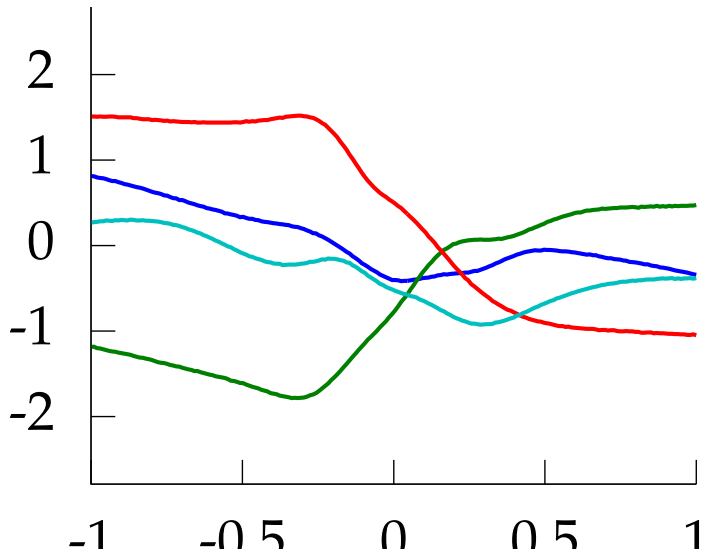
MLP Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha \text{asin}\left(\frac{w\mathbf{x}^\top \mathbf{x}' + b}{\sqrt{w\mathbf{x}^\top \mathbf{x} + b + 1}\sqrt{w\mathbf{x}'^\top \mathbf{x}' + b + 1}}\right)$$

- Based on infinite neural network model.

$$w = 40$$

$$b = 4$$



Constructing Covariance Functions

- Sum of two covariances is also a covariance function.

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$$

Constructing Covariance Functions

- Product of two covariances is also a covariance function.

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$$

Multiply by Deterministic Function

- If $f(\mathbf{x})$ is a Gaussian process.
- $g(\mathbf{x})$ is a deterministic function.
- $h(\mathbf{x}) = f(\mathbf{x})g(\mathbf{x})$
- Then

$$k_h(\mathbf{x}, \mathbf{x}') = g(\mathbf{x})k_f(\mathbf{x}, \mathbf{x}')g(\mathbf{x}')$$

where k_h is covariance for $h(\cdot)$ and k_f is covariance for $f(\cdot)$.

Covariance Functions

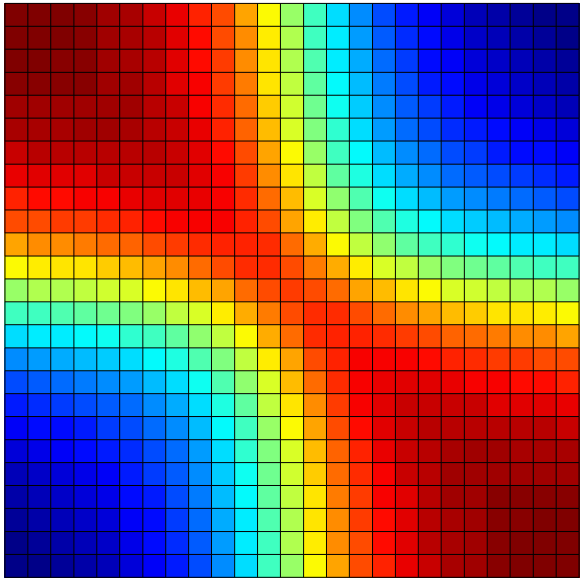
MLP Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha \text{asin}\left(\frac{w\mathbf{x}^\top\mathbf{x}' + b}{\sqrt{w\mathbf{x}^\top\mathbf{x} + b + 1}\sqrt{w\mathbf{x}'^\top\mathbf{x}' + b + 1}}\right)$$

- Based on infinite neural network model.

$$w = 40$$

$$b = 4$$



Covariance Functions

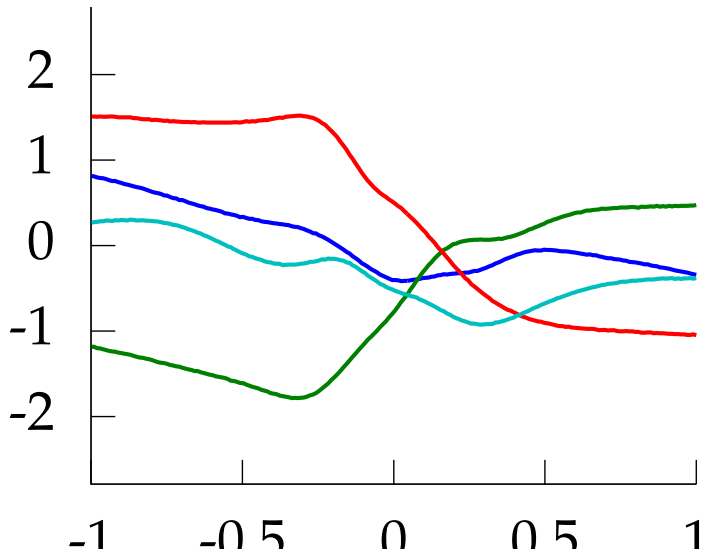
MLP Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha \text{asin}\left(\frac{w\mathbf{x}^\top \mathbf{x}' + b}{\sqrt{w\mathbf{x}^\top \mathbf{x} + b + 1}\sqrt{w\mathbf{x}'^\top \mathbf{x}' + b + 1}}\right)$$

- Based on infinite neural network model.

$$w = 40$$

$$b = 4$$



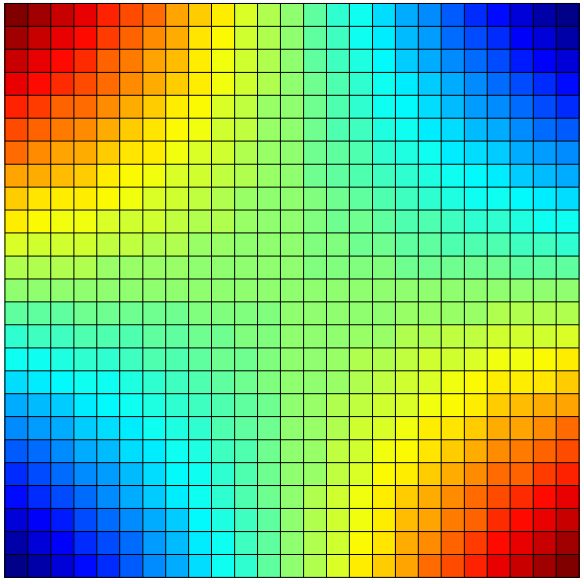
Covariance Functions

Linear Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha \mathbf{x}^\top \mathbf{x}'$$

- Bayesian linear regression.

$$\alpha = 1$$



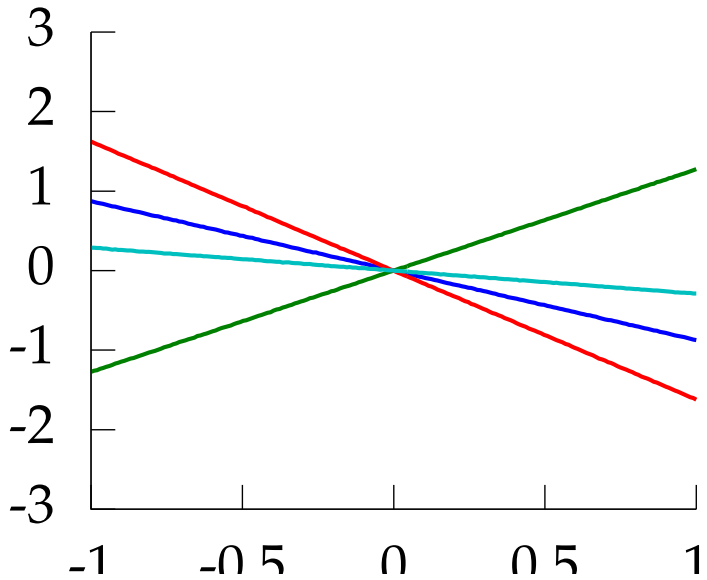
Covariance Functions

Linear Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha \mathbf{x}^\top \mathbf{x}'$$

- Bayesian linear regression.

$$\alpha = 1$$



Outline

Limitations of Gaussian Processes

- Inference is $O(n^3)$ due to matrix inverse (in practice use Cholesky).
- Gaussian processes don't deal well with discontinuities (financial crises, phosphorylation, collisions, edges in images).
- Widely used exponentiated quadratic covariance (RBF) can be too smooth in practice (but there are many alternatives!!).

Outline

Simple Markov Chain

- Assume 1-d latent state, a vector over time, $\mathbf{x} = [x_1 \dots x_T]$.
- Markov property,

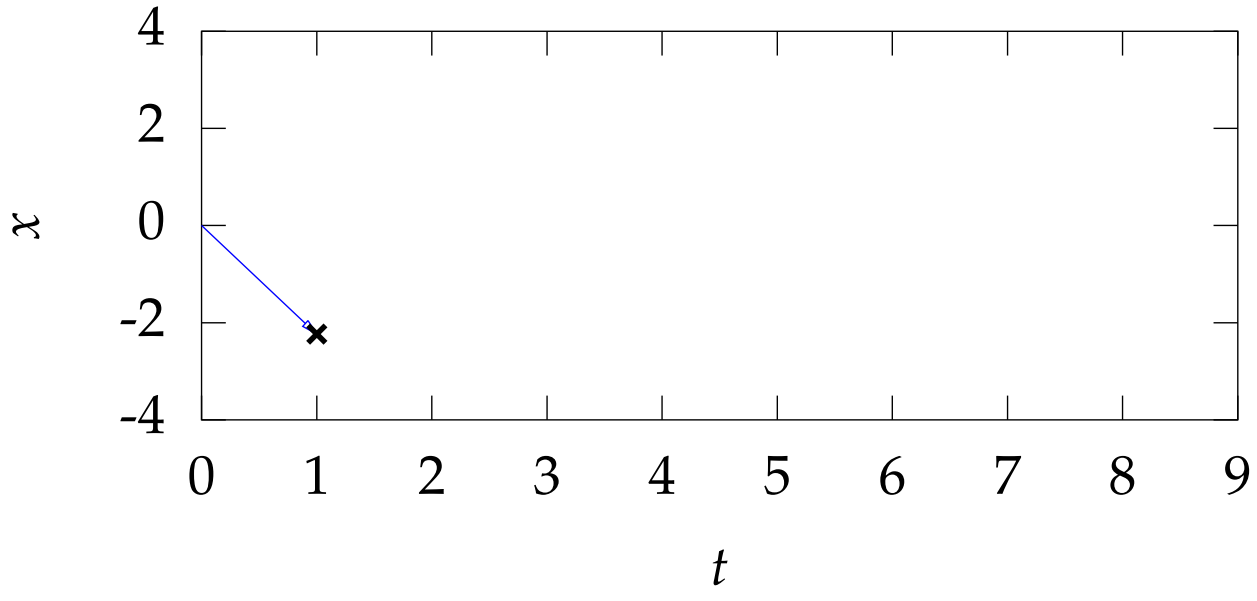
$$\begin{aligned} x_i &= x_{i-1} + \epsilon_i, \\ \epsilon_i &\sim \mathcal{N}(0, \alpha) \\ \implies x_i &\sim \mathcal{N}(x_{i-1}, \alpha) \end{aligned}$$

- Initial state,

$$x_0 \sim \mathcal{N}(0, \alpha_0)$$

- If $x_0 \sim \mathcal{N}(0, \alpha)$ we have a Markov chain for the latent states.
- Markov chain it is specified by an initial distribution (Gaussian) and a transition distribution (Gaussian).

Gauss Markov Chain

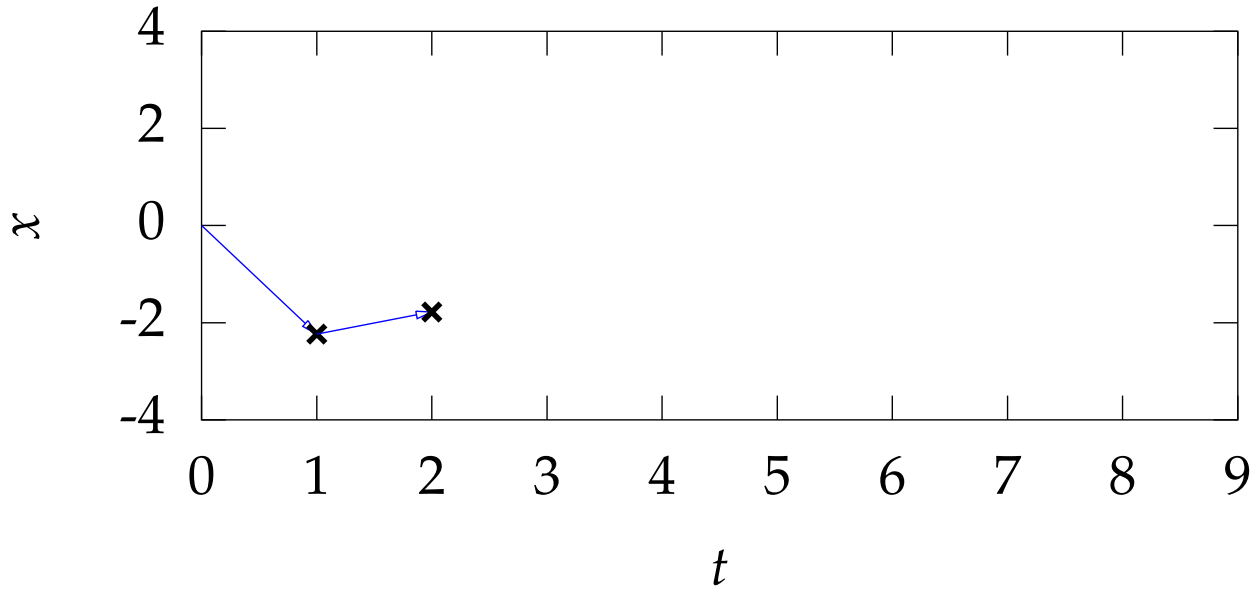


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0,1)$$

$$x_0 = 0.000, \quad \epsilon_1 = -2.24$$

$$x_1 = 0.000 - 2.24 = -2.24$$

Gauss Markov Chain

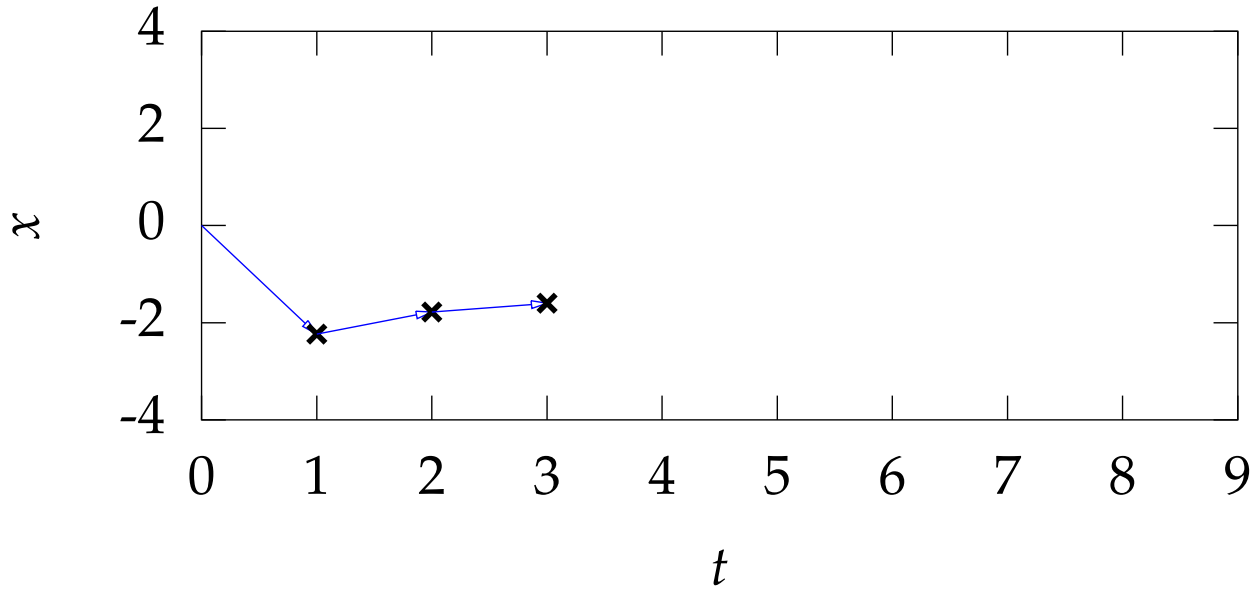


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0,1)$$

$$x_1 = -2.24, \quad \epsilon_2 = 0.457$$

$$x_2 = -2.24 + 0.457 = -1.78$$

Gauss Markov Chain

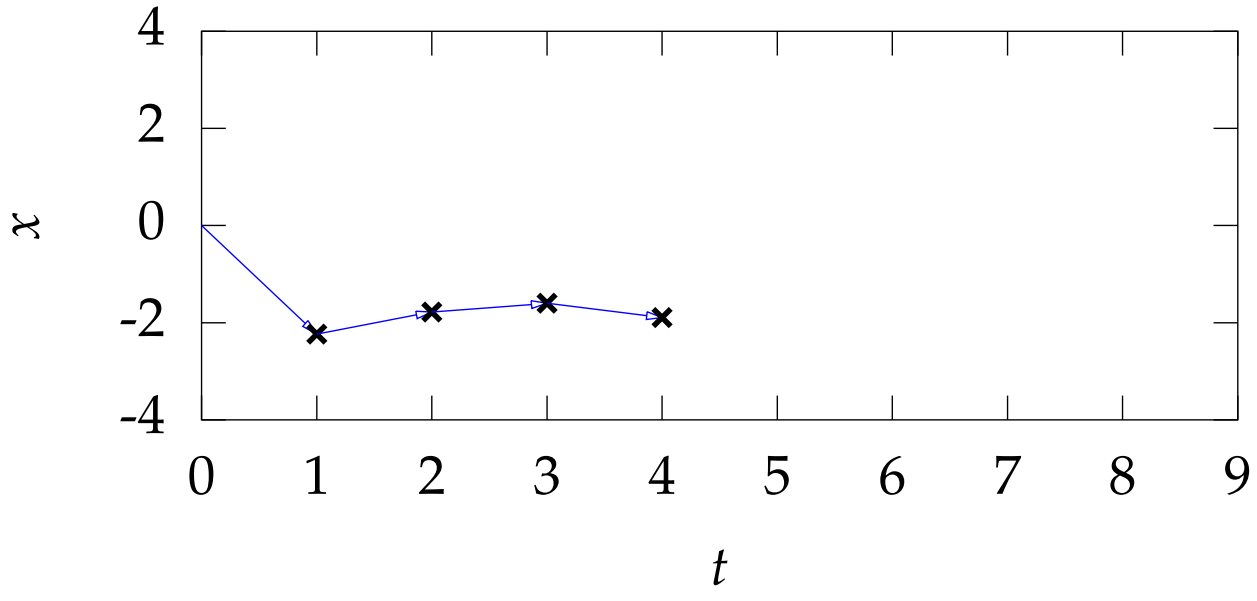


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0,1)$$

$$x_2 = -1.78, \quad \epsilon_3 = 0.178$$

$$x_3 = -1.78 + 0.178 = -1.6$$

Gauss Markov Chain

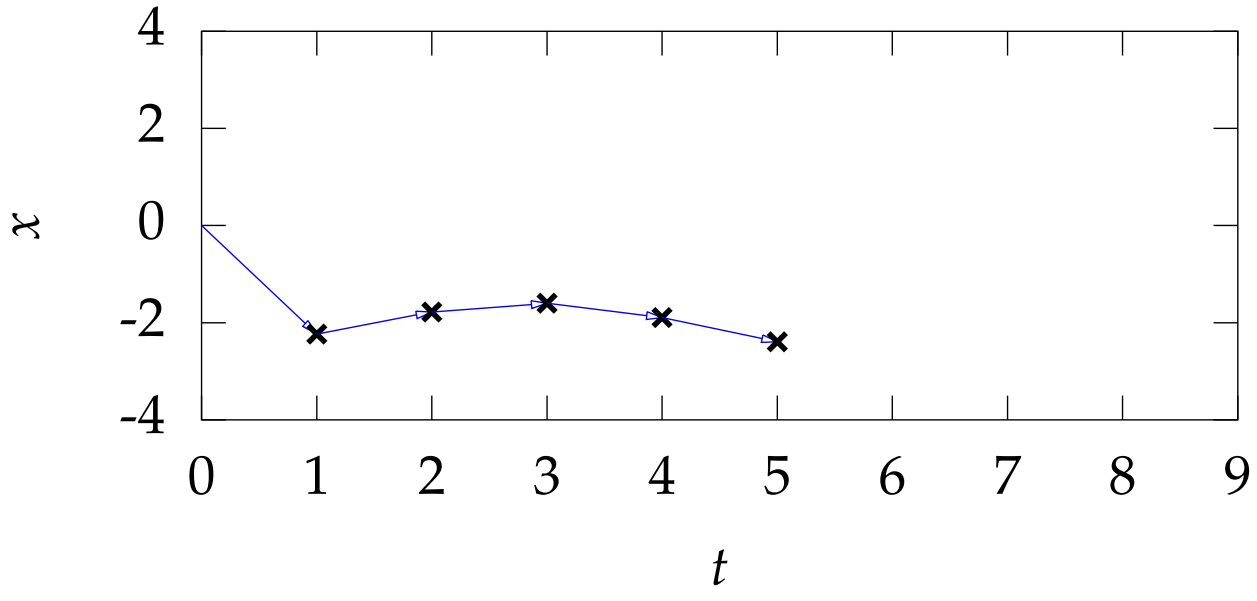


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0,1)$$

$$x_3 = -1.6, \quad \epsilon_4 = -0.292$$

$$x_4 = -1.6 - 0.292 = -1.89$$

Gauss Markov Chain

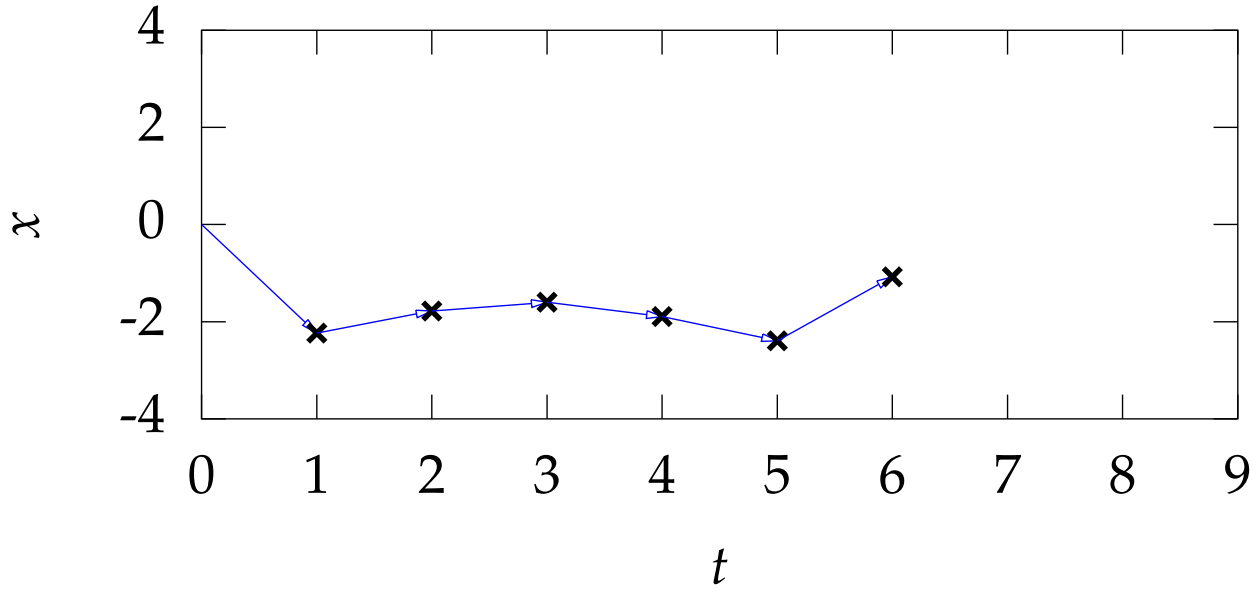


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0,1)$$

$$x_4 = -1.89, \quad \epsilon_5 = -0.501$$

$$x_5 = -1.89 - 0.501 = -2.39$$

Gauss Markov Chain

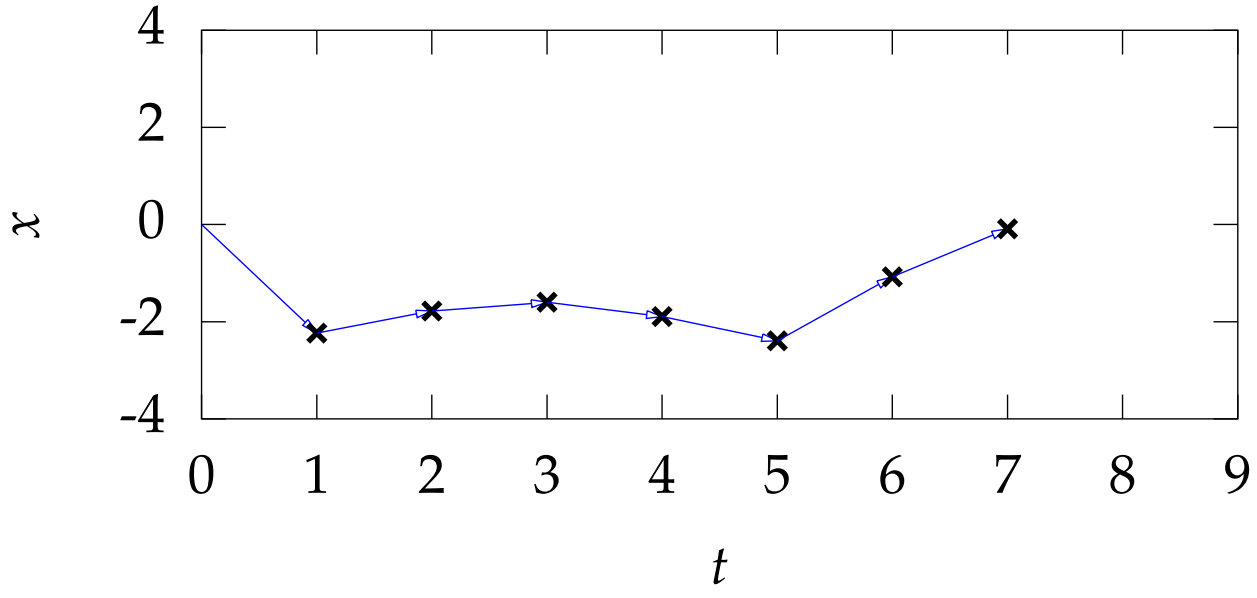


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0,1)$$

$$x_5 = -2.39, \quad \epsilon_6 = 1.32$$

$$x_6 = -2.39 + 1.32 = -1.08$$

Gauss Markov Chain

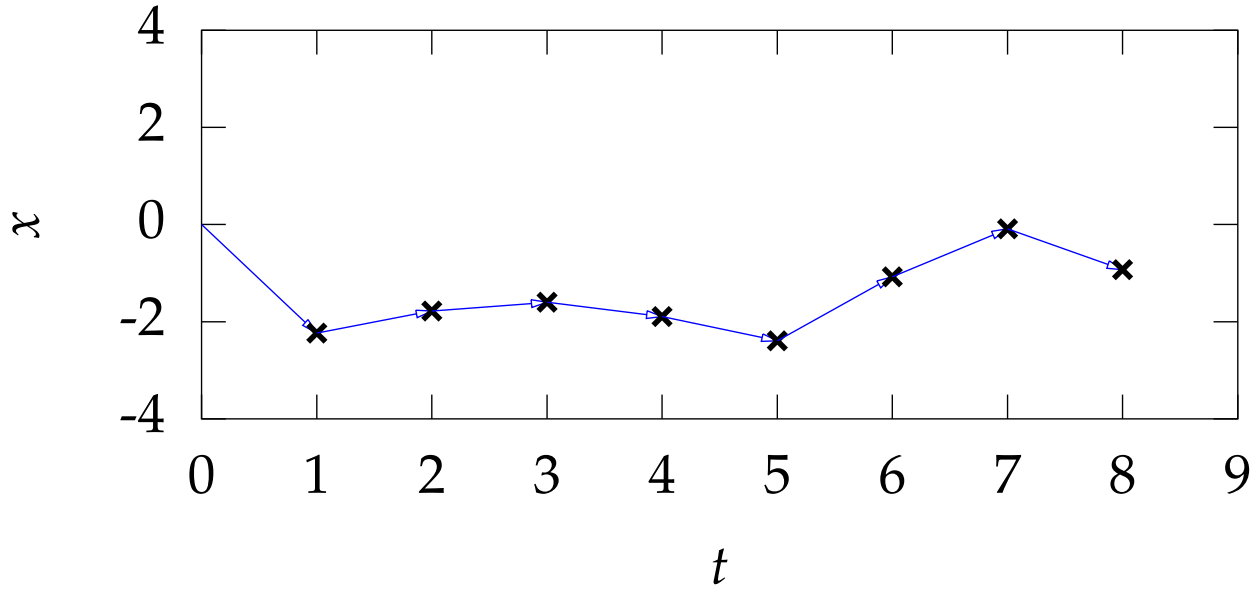


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0,1)$$

$$x_6 = -1.08, \quad \epsilon_7 = 0.989$$

$$x_7 = -1.08 + 0.989 = -0.0881$$

Gauss Markov Chain

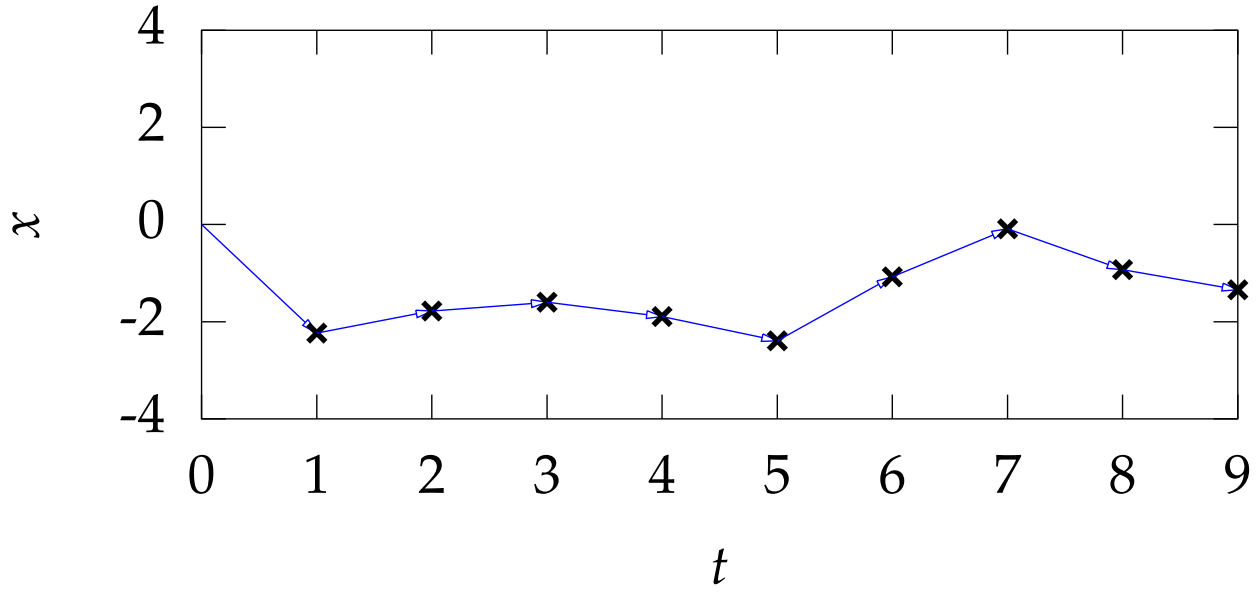


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0,1)$$

$$x_7 = -0.0881, \quad \epsilon_8 = -0.842$$

$$x_8 = -0.0881 - 0.842 = -0.93$$

Gauss Markov Chain



$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0,1)$$

$$x_8 = -0.93, \quad \epsilon_9 = -0.41$$

$$x_9 = -0.93 - 0.410 = -1.34$$

Multivariate Gaussian Properties: Reminder

If

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

and

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b}$$

then

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{W}\mu + \mathbf{b}, \mathbf{W}\mathbf{C}\mathbf{W}^{\top}\right)$$

Multivariate Gaussian Properties: Reminder

Simplified: If

$$\mathbf{z} \sim \mathcal{N}\left(0, \sigma^2 \mathbf{I}\right)$$

and

$$\mathbf{x} = \mathbf{W}\mathbf{z}$$

then

$$\mathbf{x} \sim \mathcal{N}\left(0, \sigma^2 \mathbf{W}\mathbf{W}^\top\right)$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_1 = \epsilon_1$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_2 = \epsilon_1 + \epsilon_2$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_3 = \epsilon_1 + \epsilon_2 + \epsilon_3$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_4 = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_5 = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 + \epsilon_5$$

Matrix Representation of Latent Variables

$$\mathbf{x} = \mathbf{L}_1 \times \epsilon$$

Multivariate Process

- Since $\mathbf{x}$ is linearly related to $\boldsymbol{\epsilon}$ we know $\mathbf{x}$ is a Gaussian process.
- Trick: we only need to compute the mean and covariance of $\mathbf{x}$ to determine that Gaussian.

Latent Process Mean

$$\mathbf{x} = \mathbf{L}_1 \epsilon$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \langle \mathbf{L}_1 \boldsymbol{\epsilon} \rangle$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{L}_1 \langle \boldsymbol{\epsilon} \rangle$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{L}_1 \langle \boldsymbol{\epsilon} \rangle$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \alpha \mathbf{I})$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{L}_1 \mathbf{0}$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{0}$$

Latent Process Covariance

$$\mathbf{x}\mathbf{x}^\top = \mathbf{L}_1 \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \mathbf{L}_1^\top$$

$$\mathbf{x}^\top = \boldsymbol{\epsilon}^\top \mathbf{L}^\top$$

Latent Process Covariance

$$\left\langle \mathbf{x}\mathbf{x}^\top \right\rangle = \left\langle \mathbf{L}_1 \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \mathbf{L}_1^\top \right\rangle$$

Latent Process Covariance

$$\left\langle \mathbf{x}\mathbf{x}^\top \right\rangle = \mathbf{L}_1 \left\langle \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \right\rangle \mathbf{L}_1^\top$$

Latent Process Covariance

$$\left\langle \mathbf{x}\mathbf{x}^\top \right\rangle = \mathbf{L}_1 \left\langle \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \right\rangle \mathbf{L}_1^\top$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{I}\right)$$

Latent Process Covariance

$$\left\langle \mathbf{x}\mathbf{x}^{\top} \right\rangle = \alpha \mathbf{L}_1 \mathbf{L}_1^{\top}$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \epsilon$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \alpha \mathbf{I})$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \alpha \mathbf{I})$$

$$\Longrightarrow$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \alpha \mathbf{I})$$

$$\Longrightarrow$$

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{L}_1 \mathbf{L}_1^\top\right)$$

Covariance for Latent Process II

- Make the variance dependent on time interval.
- Assume variance grows *linearly* with time.
- Justification: sum of two Gaussian distributed random variables is distributed as Gaussian with sum of variances.
- If variable's movement is additive over time (as described) variance scales linearly with time.

Covariance for Latent Process II

- Given
$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \alpha\mathbf{I}) \Longrightarrow \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha\mathbf{L_1}\mathbf{L_1}^\top\right).$$
Then
$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \Delta t\alpha\mathbf{I}) \Longrightarrow \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \Delta t\alpha\mathbf{L_1}\mathbf{L_1}^\top\right).$$
where Δt is the time interval between observations.

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \alpha \Delta t \mathbf{I}\right), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha \Delta t \mathbf{L_1} \mathbf{L_1}^\top\right)$$

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \alpha \Delta t \mathbf{I}\right), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha \Delta t \mathbf{L_1} \mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha \Delta t \mathbf{L_1} \mathbf{L_1}^\top$$

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \alpha\Delta t \mathbf{I}\right), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha\Delta t \mathbf{L_1}\mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha\Delta t \mathbf{L_1}\mathbf{L_1}^\top$$

$$k_{i,j} = \alpha\Delta t \mathbf{l}_{:,i}^\top \mathbf{l}_{:,j}$$

where $\mathbf{l}_{:,k}$ is a vector from the kth row of $\mathbf{L_1}$: the first k elements are one, the next $T - k$ are zero.

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \alpha\Delta t \mathbf{I}\right), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha\Delta t \mathbf{L_1}\mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha\Delta t \mathbf{L_1}\mathbf{L_1}^\top$$

$$k_{i,j} = \alpha\Delta t \mathbf{l}_{:,i}^\top \mathbf{l}_{:,j}$$

where $\mathbf{l}_{:,k}$ is a vector from the kth row of $\mathbf{L_1}$: the first k elements are one, the next $T - k$ are zero.

$$k_{i,j} = \alpha\Delta t \min(i, j)$$

define $\Delta t i = t_i$ so

$$k_{i,j} = \alpha \min(t_i, t_j) = k(t_i, t_j)$$

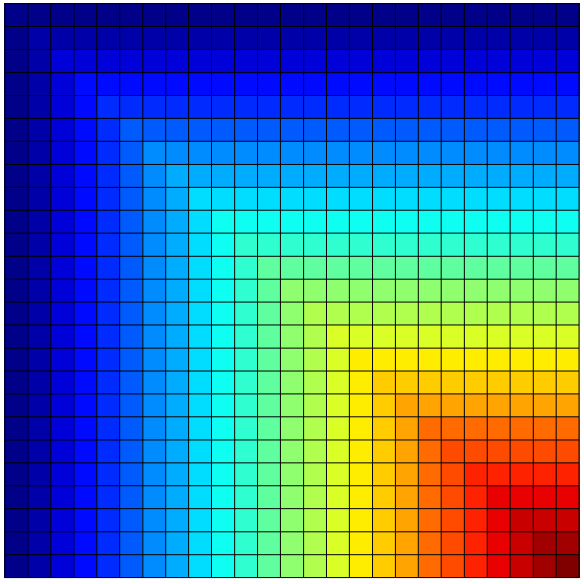
Covariance Functions

Where did this covariance matrix come from?

Markov Process

$$k(t, t') = \alpha \min(t, t')$$

- Covariance matrix is built using the *inputs* to the function t.



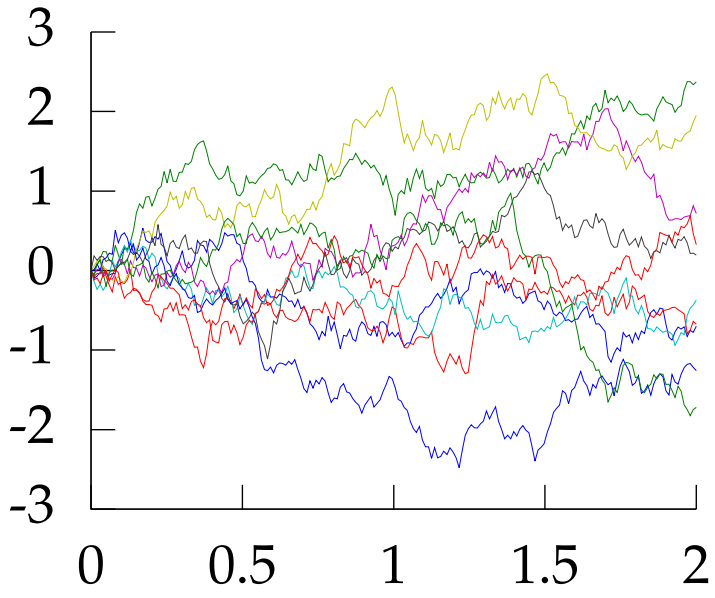
Covariance Functions

Where did this covariance matrix come from?

Markov Process

$$k(t, t') = \alpha \min(t, t')$$

- Covariance matrix is built using the *inputs* to the function t.



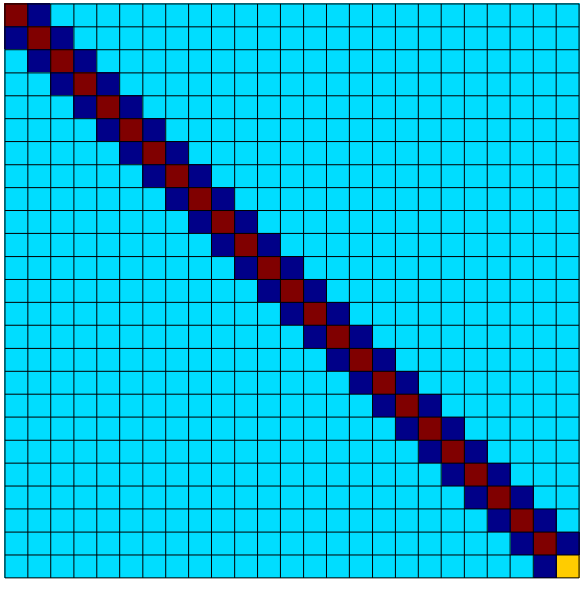
Covariance Functions

Where did this covariance matrix come from?

Markov Process

Visualization of inverse covariance (precision).

- Precision matrix is sparse: only neighbours in matrix are non-zero.
- This reflects *conditional* independencies in data.
- In this case *Markov* structure.



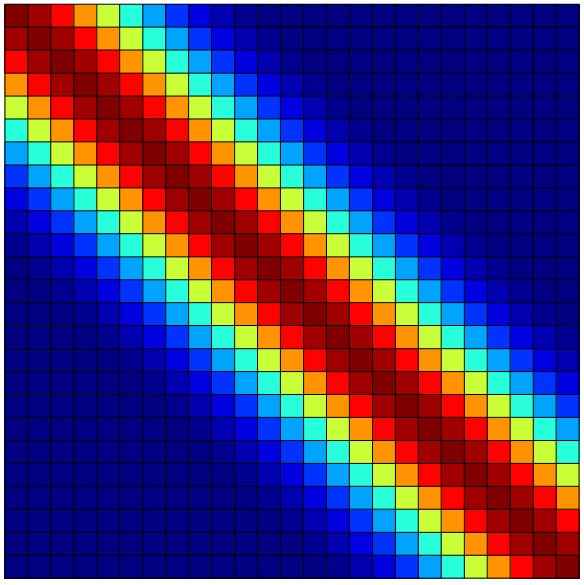
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



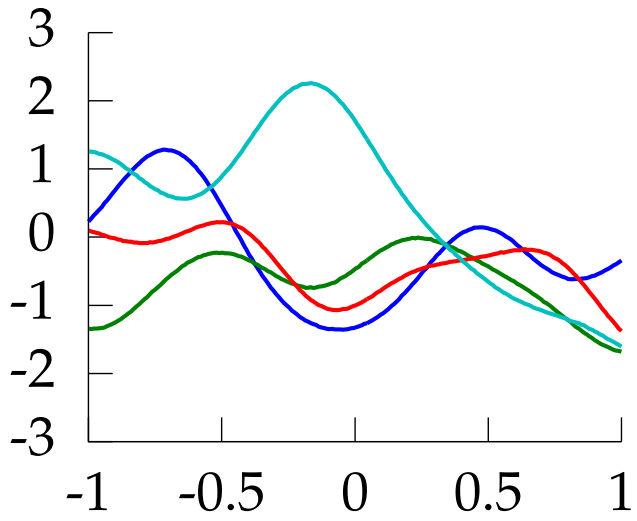
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



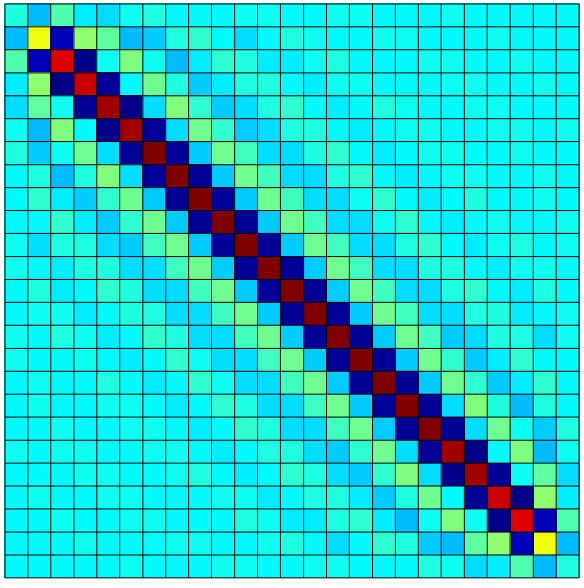
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic

Visualization of inverse covariance (precision).

- Precision matrix is not sparse.
- Each point is dependent on all the others.
- In this case non-Markovian.



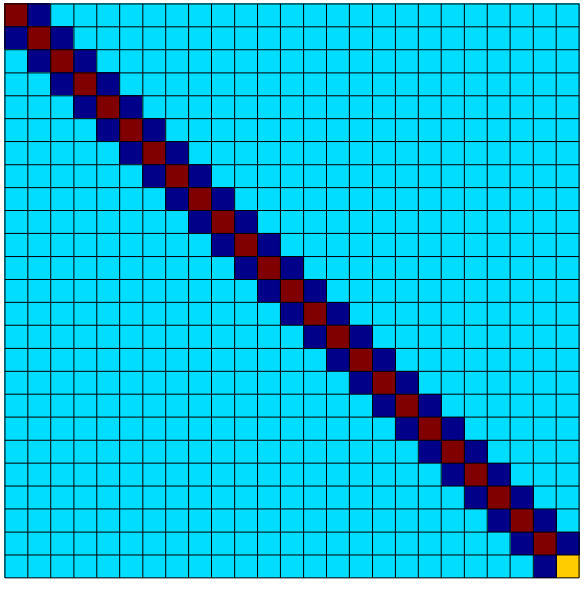
Covariance Functions

Where did this covariance matrix come from?

Markov Process

Visualization of inverse covariance (precision).

- Precision matrix is sparse: only neighbours in matrix are non-zero.
- This reflects *conditional* independencies in data.
- In this case *Markov* structure.



Simple Kalman Filter I

- We have state vector $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \dots \mathbf{x}_q \end{bmatrix} \in \mathbb{R}^{T \times q}$ and if each state evolves independently we have

$$p(\mathbf{X}) = \prod_{i=1}^{q} p(\mathbf{x}_{:,i})$$
$$p(\mathbf{x}_{:,i}) = \mathcal{N}\left(\mathbf{x}_{:,i} | \mathbf{0}, \mathbf{K}\right).$$

- We want to obtain outputs through:

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:}$$

Stacking and Kronecker Products I

- Represent with a 'stacked' system:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{I} \otimes \mathbf{K})$$

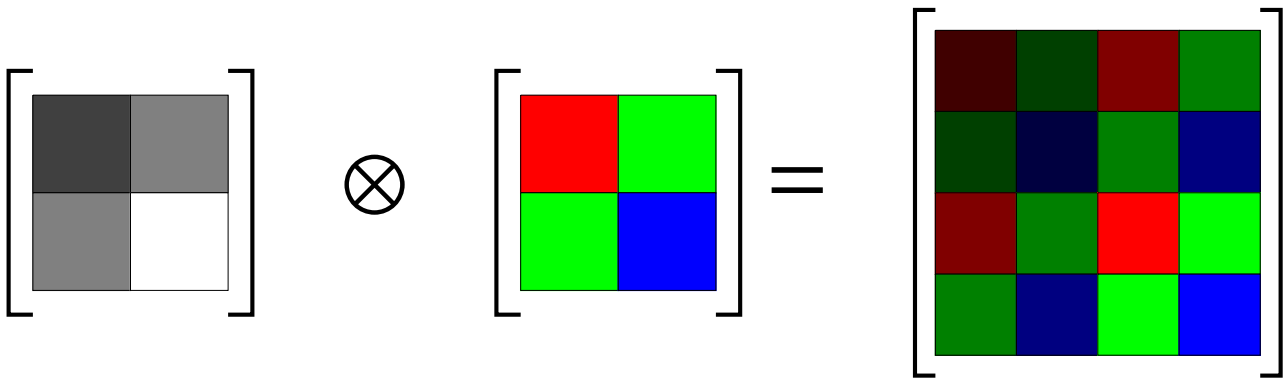
where the stacking is placing each column of $\mathbf{X}$ one on top of another as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{:,1} \\ \mathbf{x}_{:,2} \\ \vdots \\ \mathbf{x}_{:,q} \end{bmatrix}$$

Kronecker Product

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \mathbf{K} = \begin{bmatrix} a\mathbf{K} & b\mathbf{K} \\ c\mathbf{K} & d\mathbf{K} \end{bmatrix}$$

Kronecker Product



Stacking and Kronecker Products I

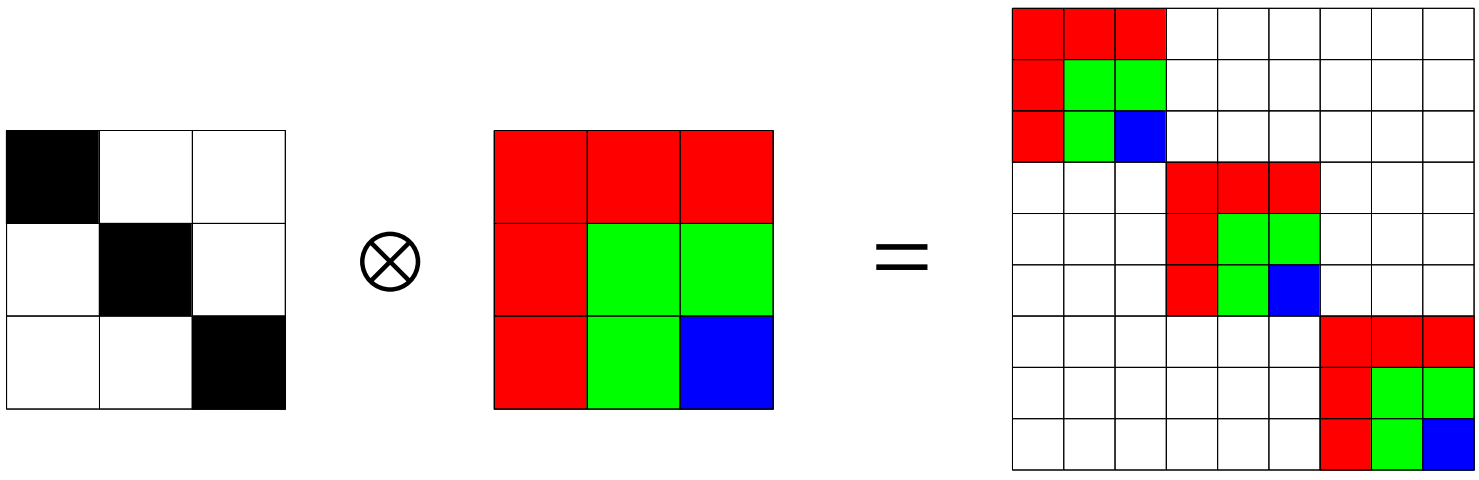
- Represent with a 'stacked' system:

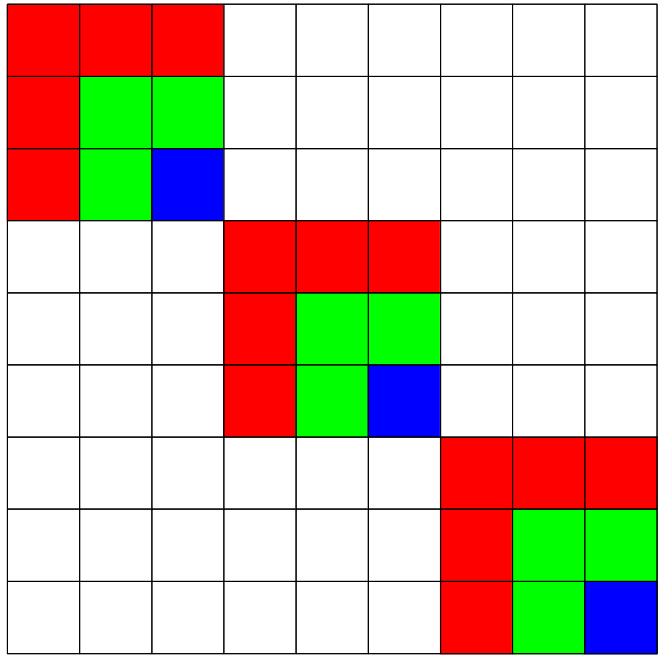
$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{I} \otimes \mathbf{K})$$

where the stacking is placing each column of $\mathbf{X}$ one on top of another as

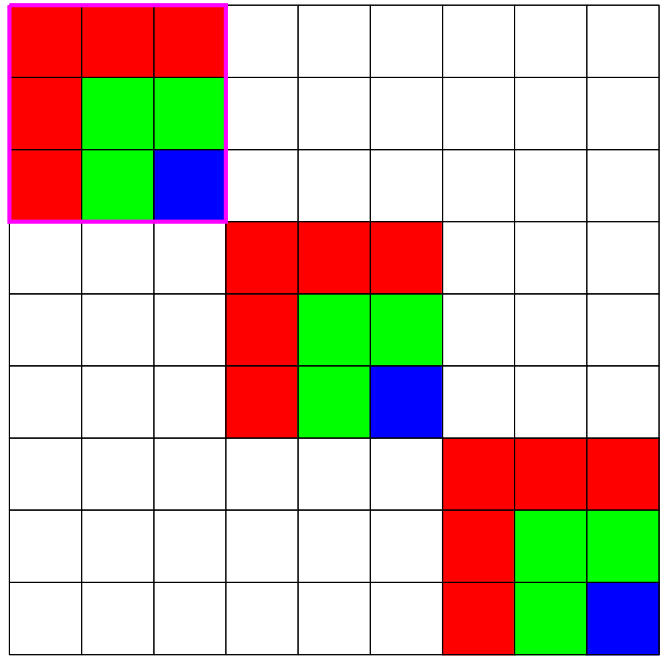
$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{:,1} \\ \mathbf{x}_{:,2} \\ \vdots \\ \mathbf{x}_{:,q} \end{bmatrix}$$

Column Stacking

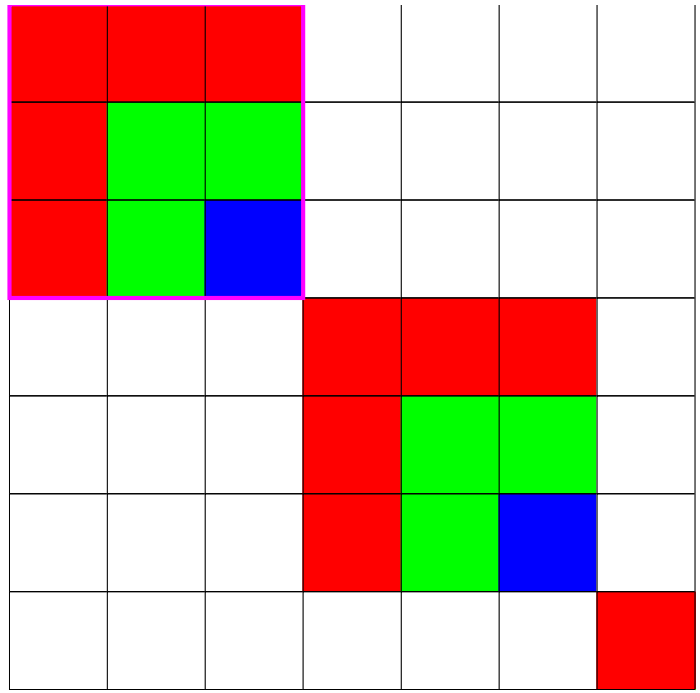




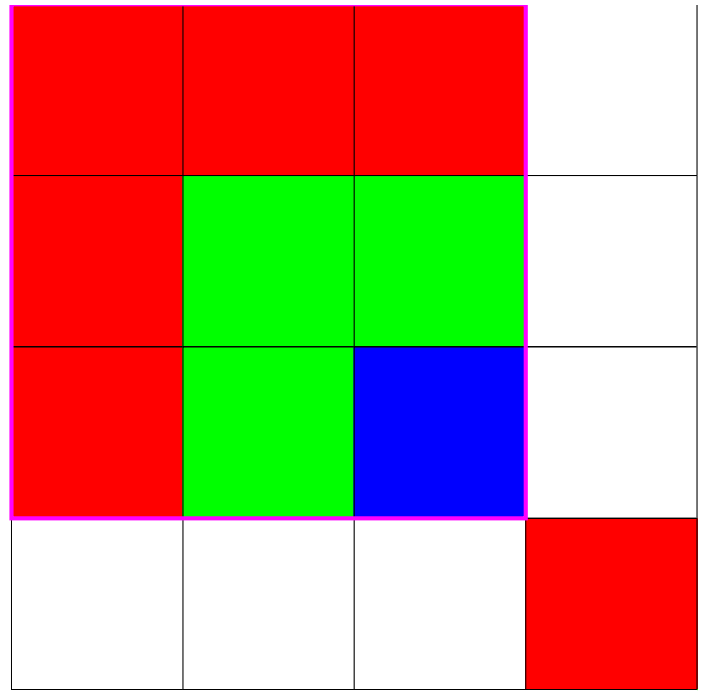
For this stacking the marginal distribution over *time* is given by the block diagonals.



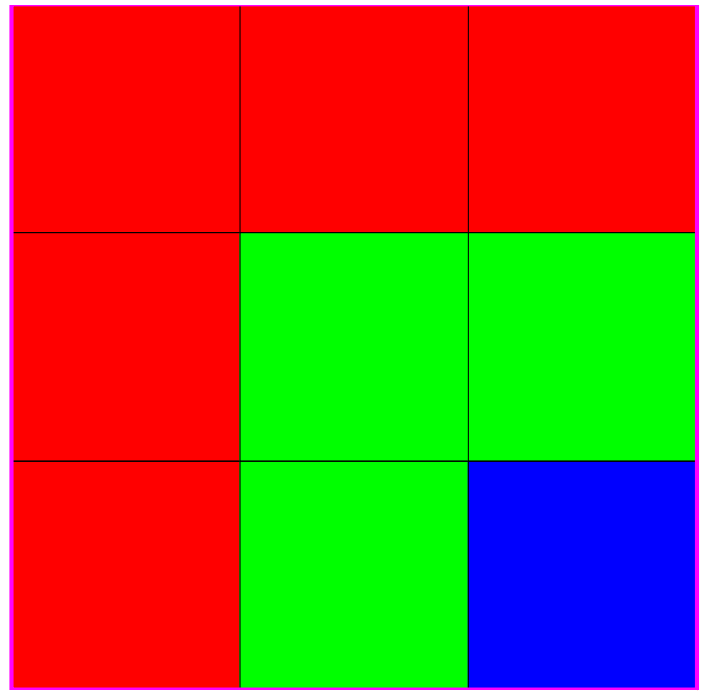
For this stacking the marginal distribution over *time* is given by the block diagonals.



For this stacking the marginal distribution over *time* is given by the block diagonals.



For this stacking the marginal distribution over *time* is given by the block diagonals.



For this stacking the marginal distribution over *time* is given by the block diagonals.

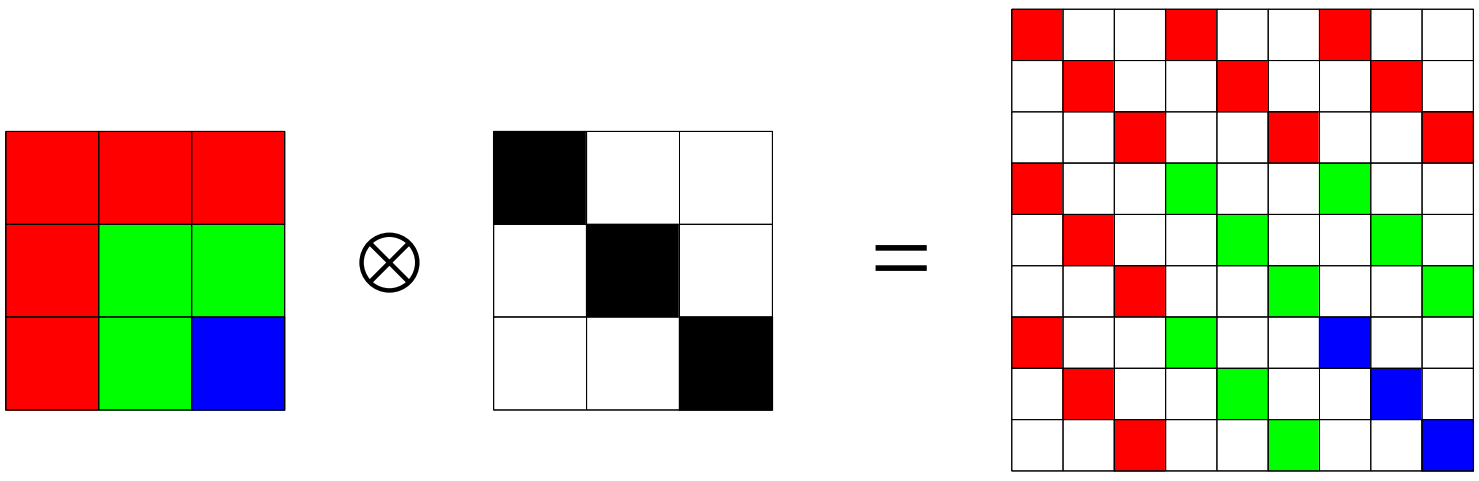
Two Ways of Stacking

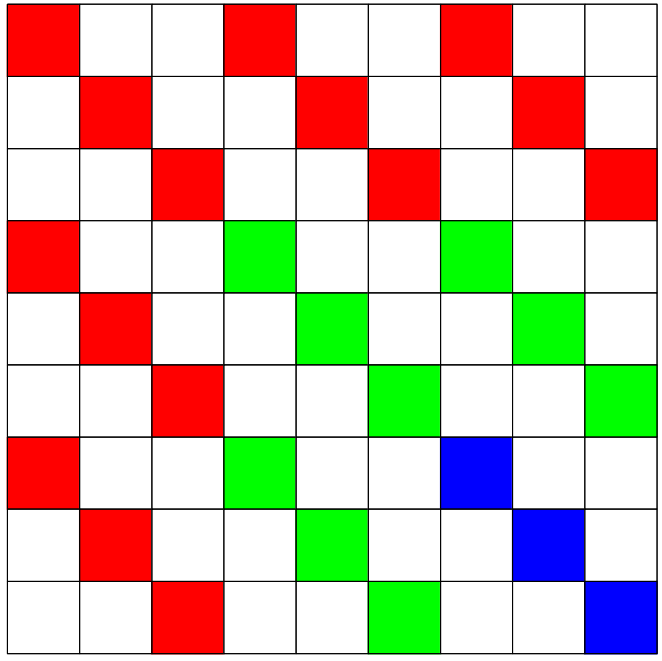
Can also stack each row of $\mathbf{X}$ to form column vector:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{1,:} \\ \mathbf{x}_{2,:} \\ \vdots \\ \mathbf{x}_{T,:} \end{bmatrix}$$

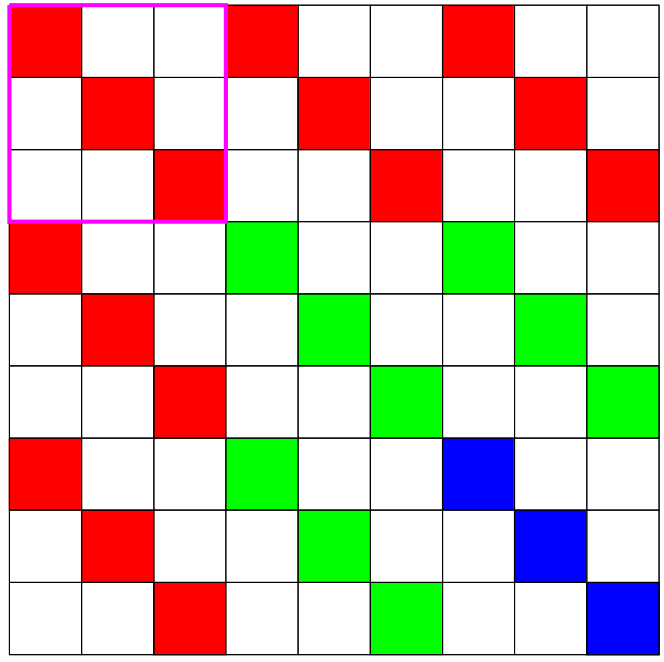
$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\mathbf{0}, \mathbf{K} \otimes \mathbf{I}\right)$$

Row Stacking

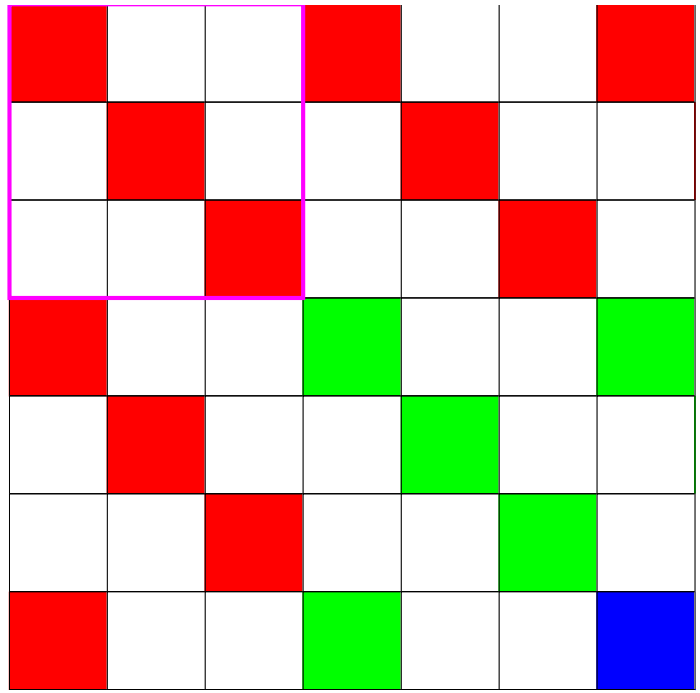




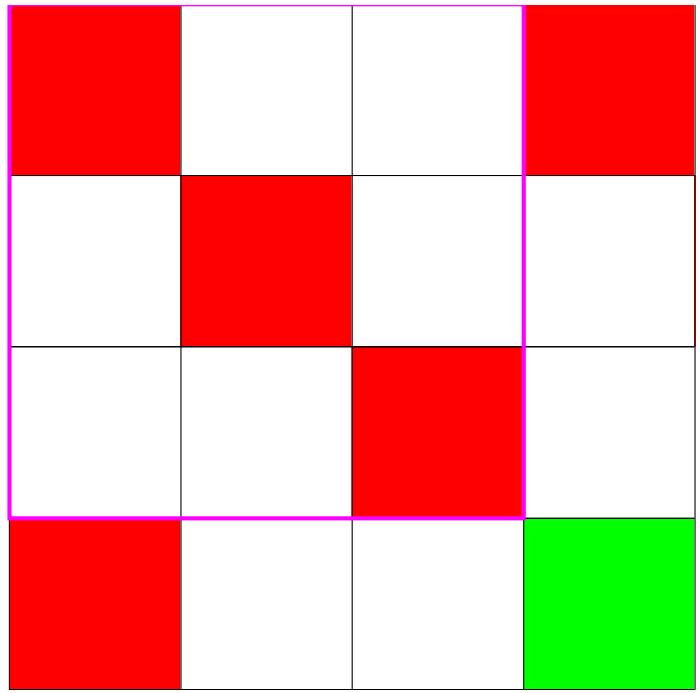
For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.



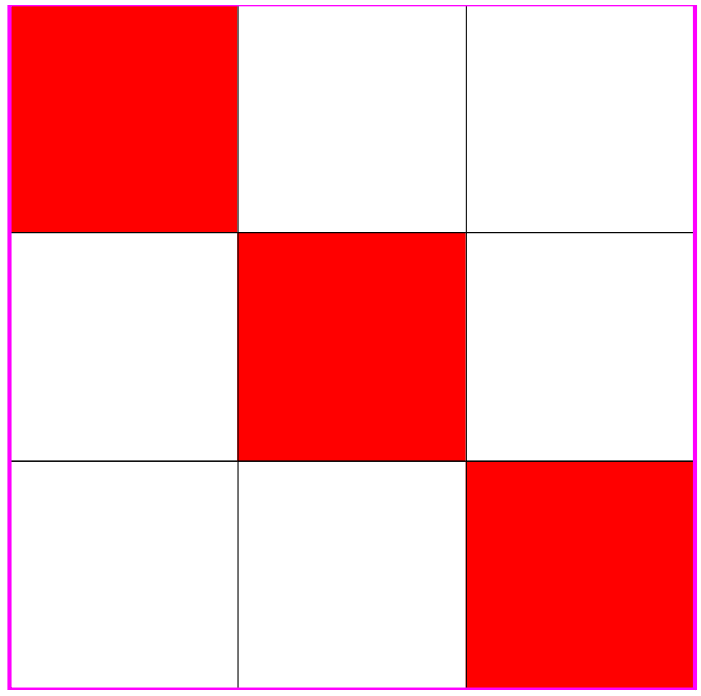
For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.



For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.



For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.



For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.

Observed Process

The observations are related to the latent points by a linear mapping matrix,

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \sigma^2\mathbf{I}\right)$$

Mapping from Latent Process to Observed

$$\begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W} \end{bmatrix} \times \begin{bmatrix} \mathbf{x}_{1,:} \\ \mathbf{x}_{2,:} \\ \mathbf{x}_{3,:} \end{bmatrix} = \begin{bmatrix} \mathbf{W}\mathbf{x}_{1,:} \\ \mathbf{W}\mathbf{x}_{2,:} \\ \mathbf{W}\mathbf{x}_{3,:} \end{bmatrix}$$

Output Covariance

This leads to a covariance of the form

$$(\mathbf{I} \otimes \mathbf{W})(\mathbf{K} \otimes \mathbf{I})(\mathbf{I} \otimes \mathbf{W}^\top) + \mathbf{I}\sigma^2$$

Using $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$ This leads to

$$\mathbf{K} \otimes \mathbf{W}\mathbf{W}^\top + \mathbf{I}\sigma^2$$

or

$$\mathbf{y} \sim \mathcal{N}\left(0, \mathbf{W}\mathbf{W}^\top \otimes \mathbf{K} + \mathbf{I}\sigma^2\right)$$

Foundations and Trends® in
Machine Learning
Vol. 4, No. 3 (2011) 195–266

DOI: 10.1561/2200000036

now
the essence of knowledge

Kernels for Vector-Valued Functions: A Review

By Mauricio A. Álvarez,
Lorenzo Rosasco and Neil D. Lawrence

Kronecker Structure GPs

- This Kronecker structure leads to several published models.

$$(\mathbf{K}(\mathbf{x}, \mathbf{x}'))_{j,j'} = k(\mathbf{x}, \mathbf{x}')k_T(j, j'),$$

  where k has $\mathbf{x}$ and k_T has i as inputs.
- Can think of multiple output covariance functions as covariances with augmented input.
- Alongside $\mathbf{x}$ we also input the j associated with the *output* of interest.

Separable Covariance Functions

- Taking $\mathbf{B} = \mathbf{W}\mathbf{W}^\top$ we have a matrix expression across outputs.

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}')\mathbf{B},$$

  where $\mathbf{B}$ is a $p \times p$ symmetric and positive semi-definite matrix.
- $\mathbf{B}$ is called the *coregionalization* matrix.
- We call this class of covariance functions *separable* due to their product structure.

Sum of Separable Covariance Functions

- In the same spirit a more general class of kernels is given by

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{q} k_j(\mathbf{x}, \mathbf{x}')\mathbf{B}_j.$$

- This can also be written as

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \sum_{j=1}^{q} \mathbf{B}_j \otimes k_j(\mathbf{X}, \mathbf{X}),$$

- This is like several Kalman filter-type models added together, but each one with a different set of latent functions.
- We call this class of kernels sum of separable kernels (SoS kernels).

Geostatistics

- Use of GPs in Geostatistics is called kriging.
- These multi-output GPs pioneered in geostatistics: prediction over vector-valued output data is known as *cokriging*.
- The model in geostatistics is known as the *linear model of coregionalization* (LMC, Journel and Huijbregts (1978); Goovaerts (1997)).
- Most machine learning multitask models can be placed in the context of the LMC model.

Weighted sum of Latent Functions

- In the linear model of coregionalization (LMC) outputs are expressed as linear combinations of independent random functions.
- In the LMC, each component f_j is expressed as a linear sum

$$f_j(\mathbf{x}) = \sum_{j=1}^{q} w_{j,j} u_j(\mathbf{x}).$$

  where the latent functions are independent and have covariance functions $k_j(\mathbf{x}, \mathbf{x}')$.
- The processes $\{f_j(\mathbf{x})\}_{j=1}^{q}$ are independent for $q \neq j'$.

Kalman Filter Special Case

- The Kalman filter is an example of the LMC where $u_i(\mathbf{x}) \rightarrow x_i(t)$.
- I.e. we've moved form time input to a more general input space.
- In matrix notation:
  1. Kalman filter
  $$\mathbf{F} = \mathbf{W}\mathbf{X}$$
  2. LMC
  $$\mathbf{F} = \mathbf{W}\mathbf{U}$$

  where the rows of these matrices $\mathbf{F}$, $\mathbf{X}$, $\mathbf{U}$ each contain q samples from their corresponding functions at a different time (Kalman filter) or spatial location (LMC).

Intrinsic Coregionalization Model

- If one covariance used for latent functions (like in Kalman filter).
- This is called the intrinsic coregionalization model (ICM, Goovaerts (1997)).
- The kernel matrix corresponding to a dataset $\mathbf{X}$ takes the form

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

Autokrigeability

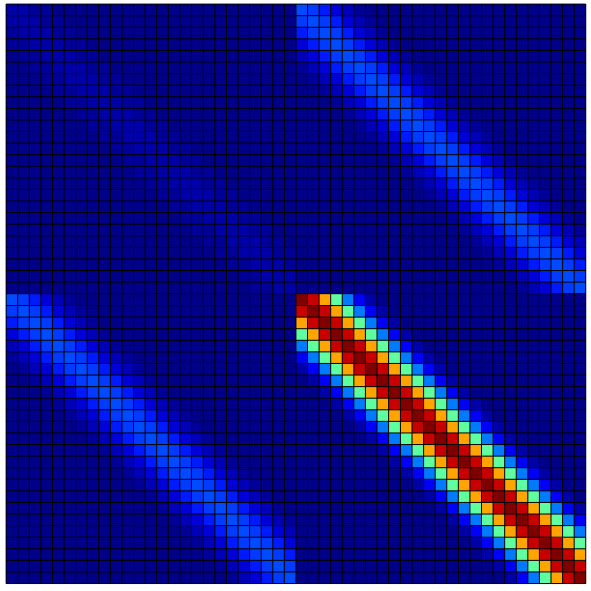
- If outputs are noise-free, maximum likelihood is equivalent to independent fits of $\mathbf{B}$ and $k(\mathbf{x}, \mathbf{x}')$ (Helterbrand and Cressie, 1994).
- In geostatistics this is known as autokrigeability (Wackernagel, 2003).
- In multitask learning its the cancellation of intertask transfer (Bonilla et al., 2008).

Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}\mathbf{w}^\top \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$

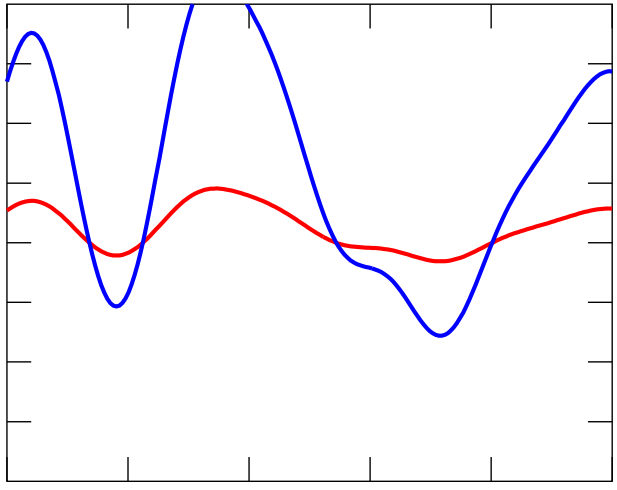


Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}\mathbf{w}^\top \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$

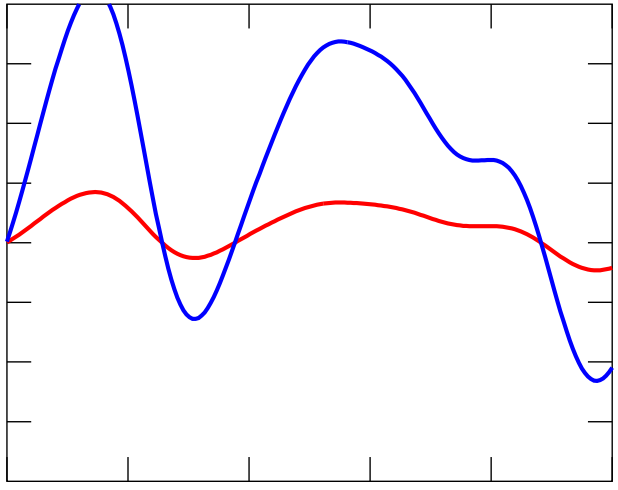


Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}\mathbf{w}^\top \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$

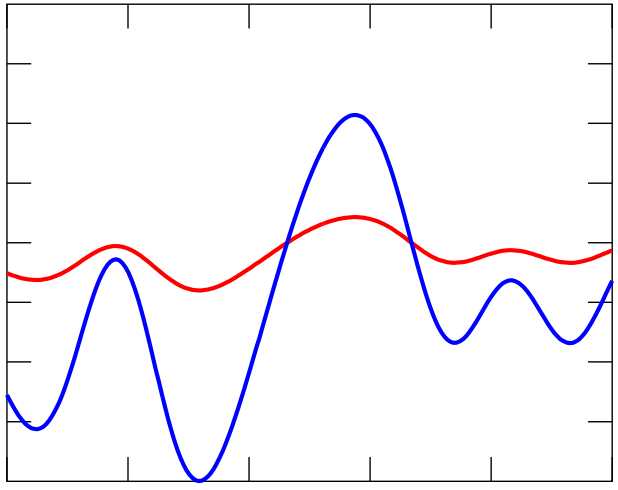


Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}\mathbf{w}^\top \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$

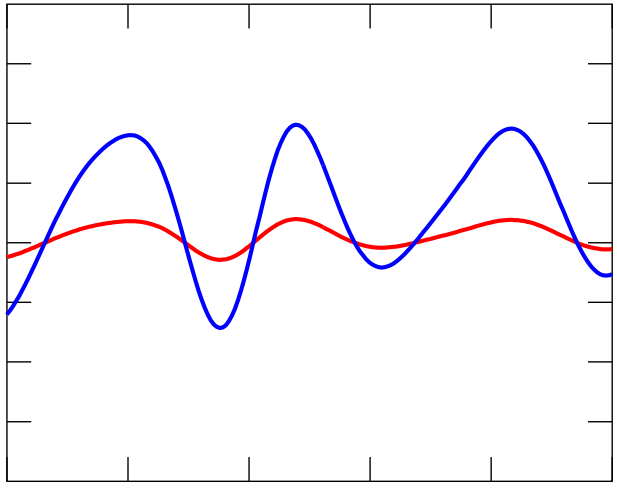


Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}\mathbf{w}^\top \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

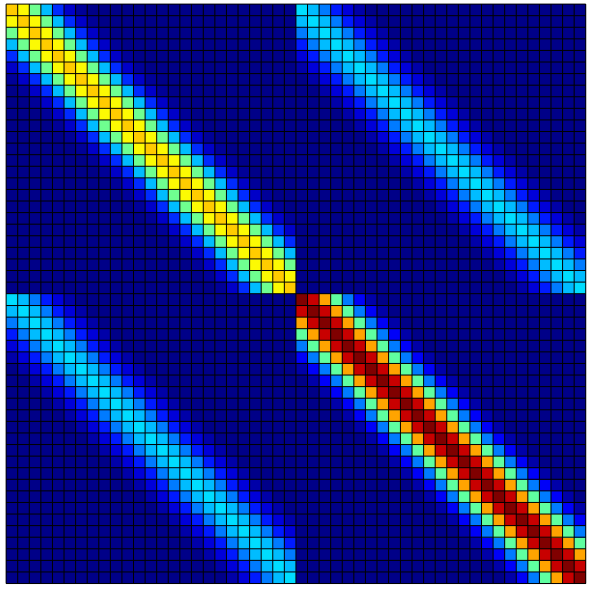
$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

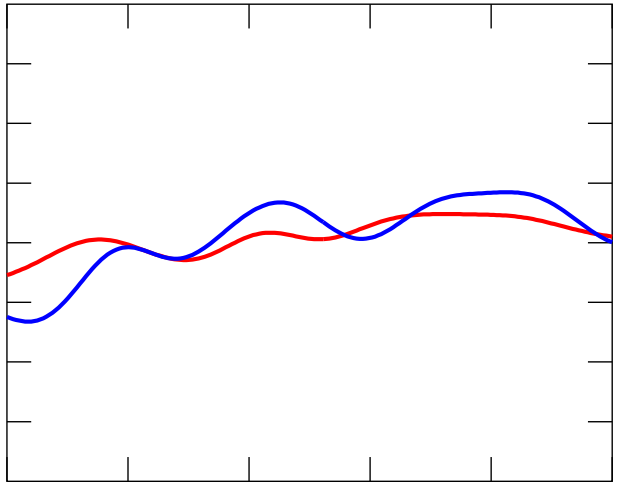
$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

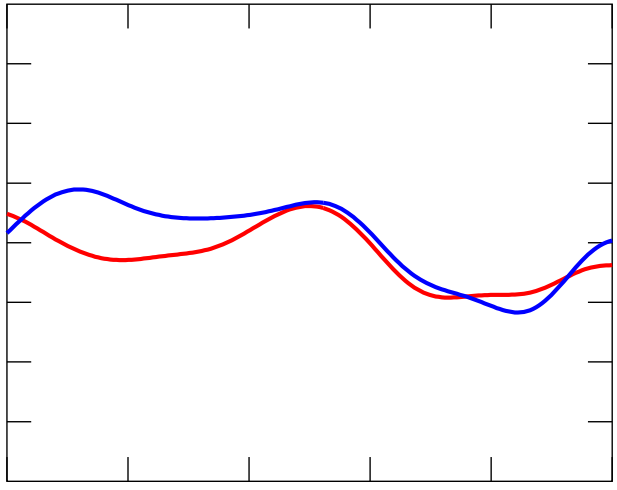
$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

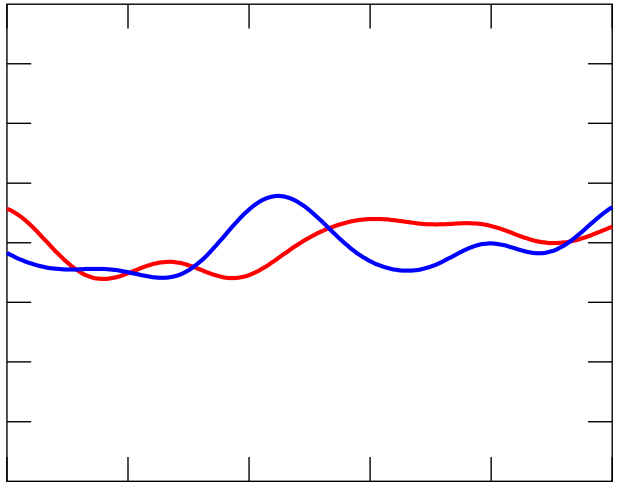
$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

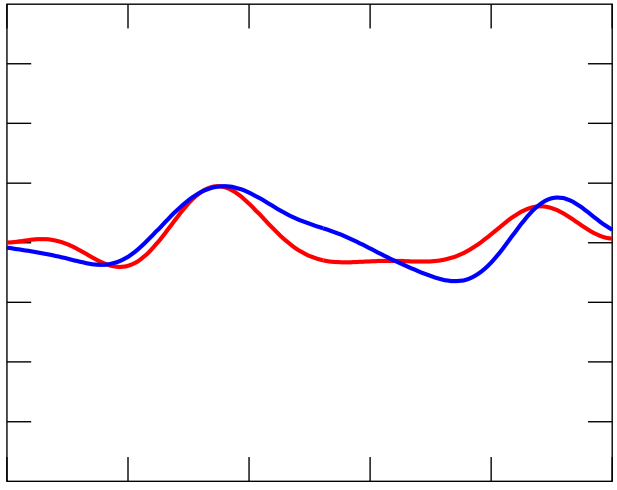
$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



LMC Samples

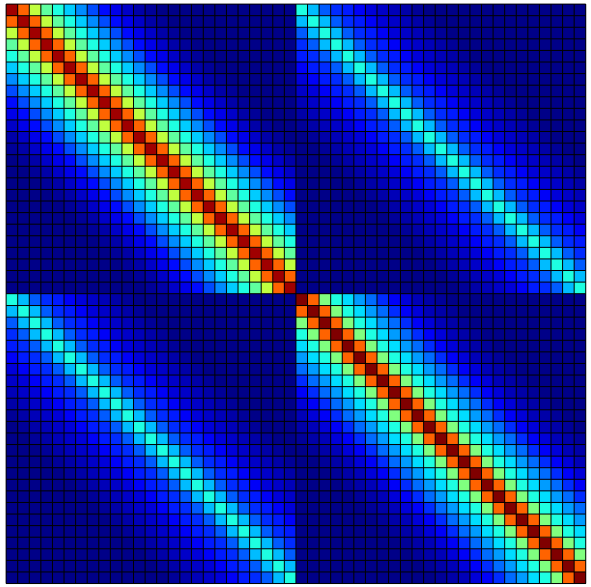
$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC Samples

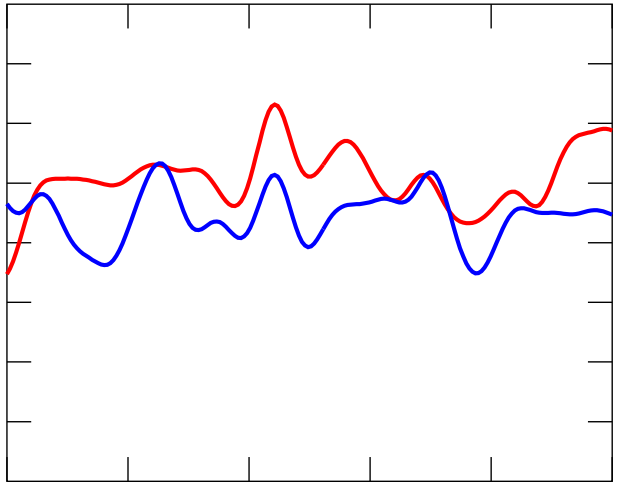
$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC Samples

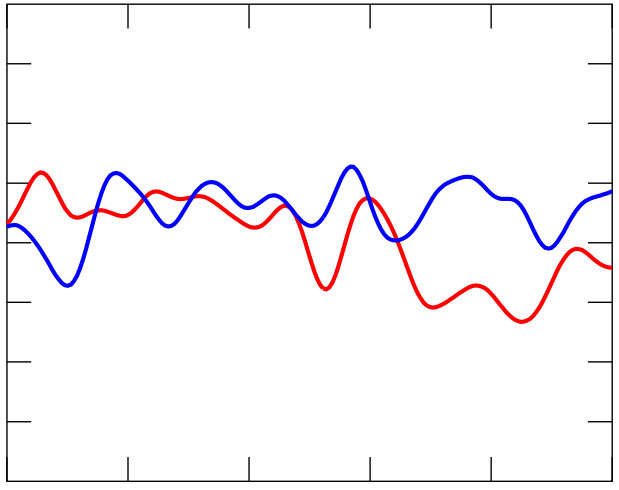
$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC Samples

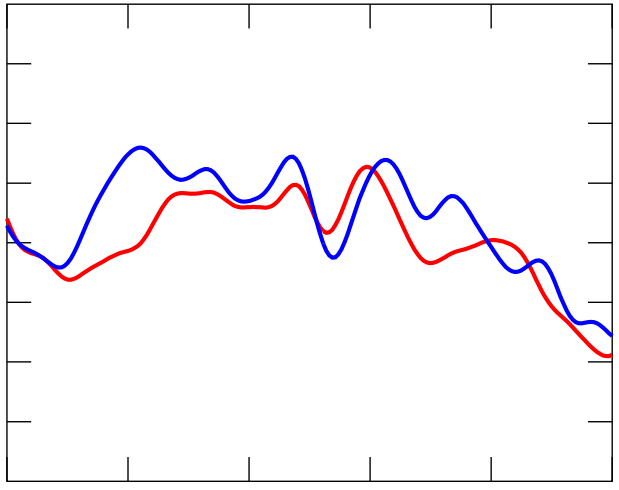
$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC Samples

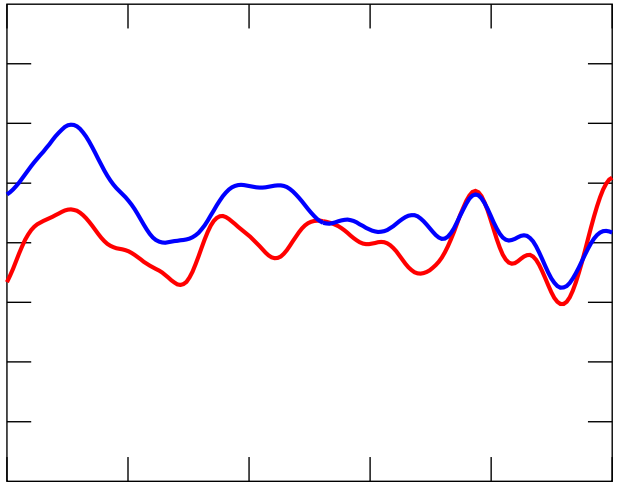
$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC in Machine Learning and Statistics

- Used in machine learning for GPs for multivariate regression and in statistics for computer emulation of expensive multivariate computer codes.
- Imposes the correlation of the outputs explicitly through the set of coregionalization matrices.
- Setting $\mathbf{B} = \mathbf{I}_p$ assumes outputs are conditionally independent given the parameters $\boldsymbol{\theta}$. (Minka and Picard, 1997; Lawrence and Platt, 2004; Yu et al., 2005).
- More recent approaches for multiple output modeling are different versions of the linear model of coregionalization.

Semiparametric Latent Factor Model

- Coregionalization matrices are rank 1 Teh et al. (2005). rewrite equation (**??**) as

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \sum_{j=1}^{q} \mathbf{w}_{:,j}\mathbf{w}_{:,j}^{\top} \otimes k_j(\mathbf{X}, \mathbf{X}).$$

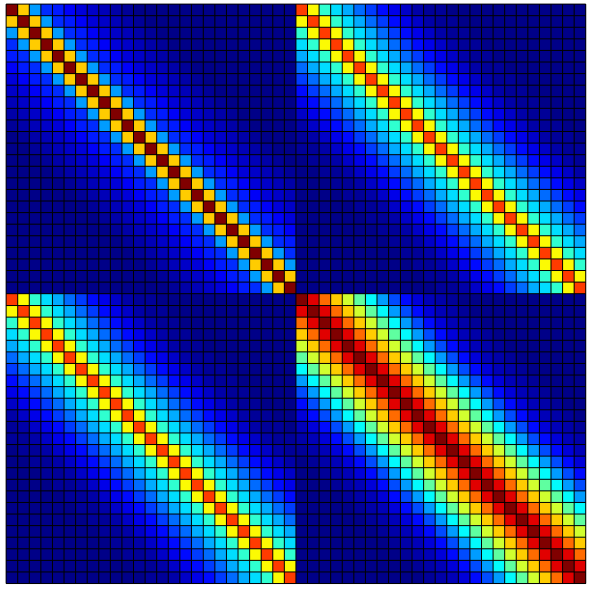
- Like the Kalman filter, but each latent function has a *different* covariance.
- Authors suggest using an exponentiated quadratic characteristic length-scale for each input dimension.

Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^{\top} \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^{\top} \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

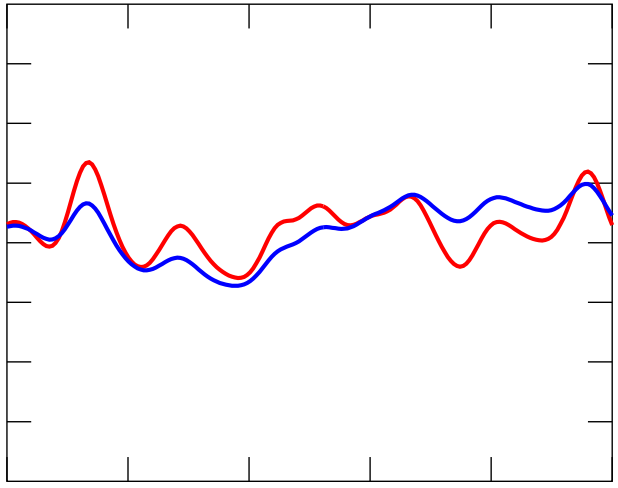


Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^{\top} \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^{\top} \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

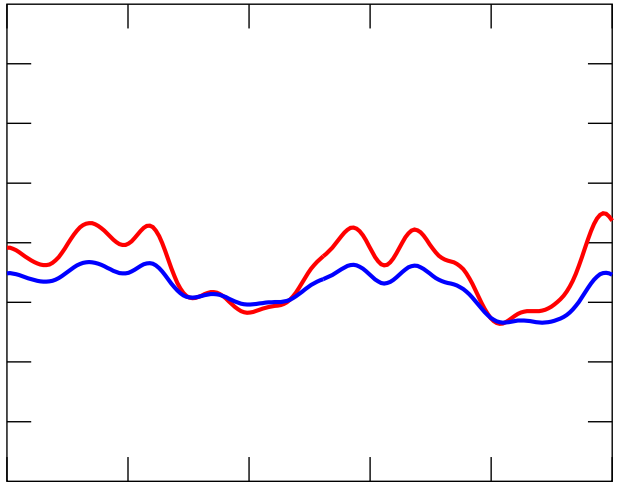


Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^{\top} \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^{\top} \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

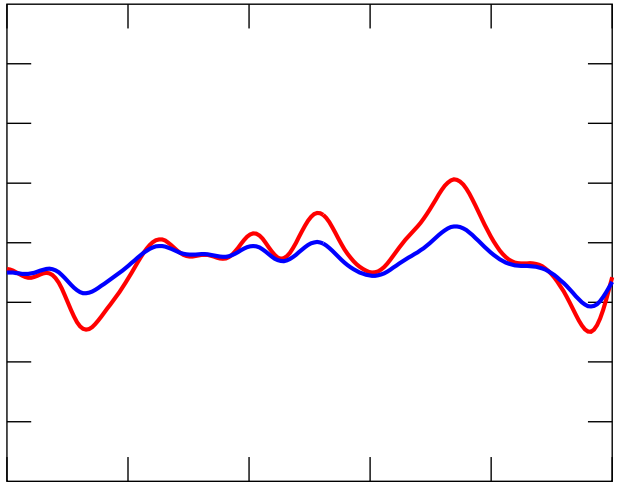


Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^{\top} \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^{\top} \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

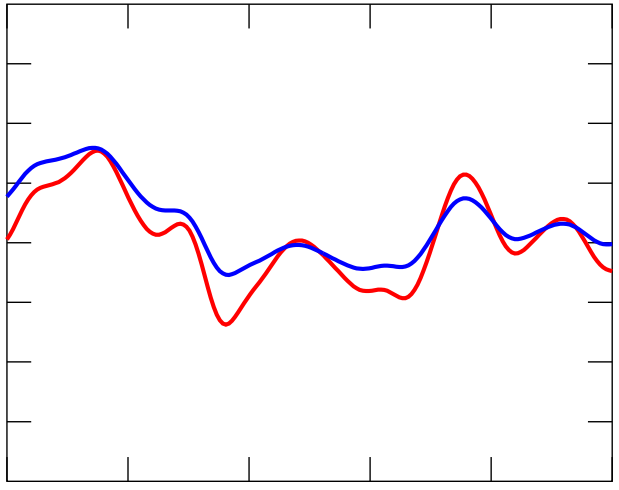


Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^{\top} \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^{\top} \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$



Gaussian processes for Multi-task, Multi-output and Multi-class

- Bonilla et al. (2008) suggest ICM for multitask learning.
- Use a PPCA form for $\mathbf{B}$: similar to our Kalman filter example.
- Refer to the autokrigeability effect as the cancellation of inter-task transfer.
- Also discuss the similarities between the multi-task GP and the ICM, and its relationship to the SLFM and the LMC.

Multitask Classification

- Mostly restricted to the case where the outputs are conditionally independent given the hyperparameters ϕ (Minka and Picard, 1997; Williams and Barber, 1998; Lawrence and Platt, 2004; Seeger and Jordan, 2004; Yu et al., 2005; Rasmussen and Williams, 2006).
- Intrinsic coregionalization model has been used in the multiclass scenario. Skolidis and Sanguinetti (2011) use the intrinsic coregionalization model for classification, by introducing a probit noise model as the likelihood.
- Posterior distribution is no longer analytically tractable: approximate inference is required.

Computer Emulation

- A statistical model used as a surrogate for a computationally expensive computer model.
- Higdon et al. (2008) use the linear model of coregionalization to model images representing the evolution of the implosion of steel cylinders.
- In Conti and O'Hagan (2009) use the ICM to model a vegetation model: called the Sheffield Dynamic Global Vegetation Model (Woodward et al., 1998).

Bochner's Theorem

Given a positive finite Borel measure μ on the real line $\mathbb{R}$, the Fourier transform Q of μ is the continuous function

$$Q(t) = \int_{\mathbb{R}} e^{-itx} \mathrm{d}\mu(x).$$

Q is continuous since for a fixed x, the function e^{-itx} is continuous and periodic. The function Q is a positive definite function, i.e. the kernel $k(x, x') = Q(x' - x)$ is positive definite.

Bochner's theorem says the converse is true, i.e. every positive definite function Q is the Fourier transform of a positive finite Borel measure. A proof can be sketched as follows.

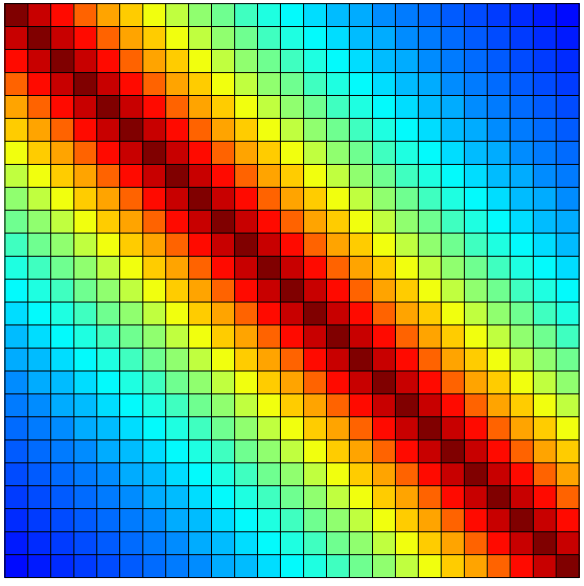
Covariance Functions

Where did this covariance matrix come from?

Ornstein-Uhlenbeck (stationary Gauss-Markov) covariance function

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{|\mathbf{x} - \mathbf{x}'|}{2\ell^2}\right)$$

- In one dimension arises from a stochastic differential equation. Brownian motion in a parabolic tube.
- In higher dimension a Fourier filter of the form $\frac{1}{\pi(1+x^2)}$.



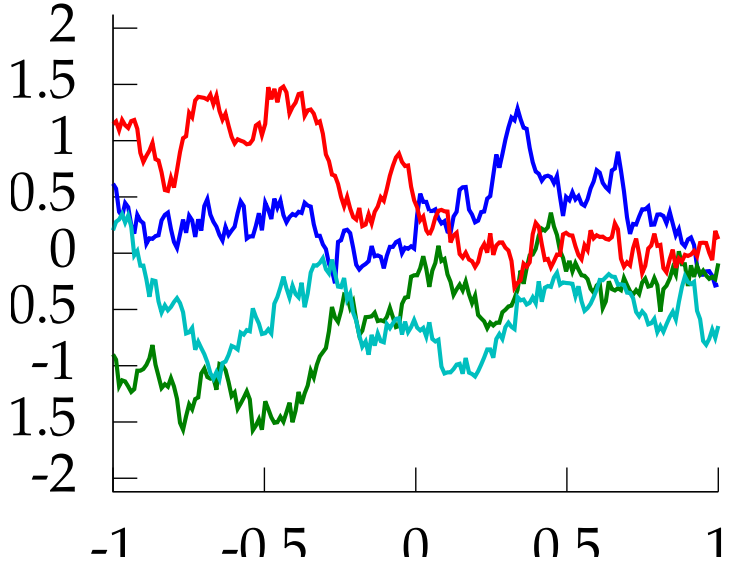
Covariance Functions

Where did this covariance matrix come from?

Ornstein-Uhlenbeck (stationary Gauss-Markov) covariance function

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{|\mathbf{x} - \mathbf{x}'|}{2\ell^2}\right)$$

- In one dimension arises from a stochastic differential equation. Brownian motion in a parabolic tube.
- In higher dimension a Fourier filter of the form $\frac{1}{\pi(1+x^2)}$.



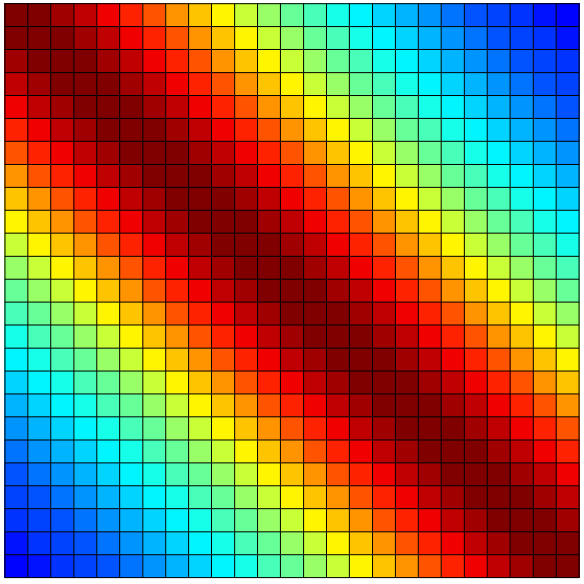
Covariance Functions

Where did this covariance matrix come from?

Matern 3/2 Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha\left(1 + \sqrt{3}r\right)\exp\left(-\sqrt{3}r\right) \quad \text{where} \quad r = \frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\ell}$$

- Matern 3/2 is a once differentiable covariance.
- Matern family constructed with Student-t filters in Fourier space.



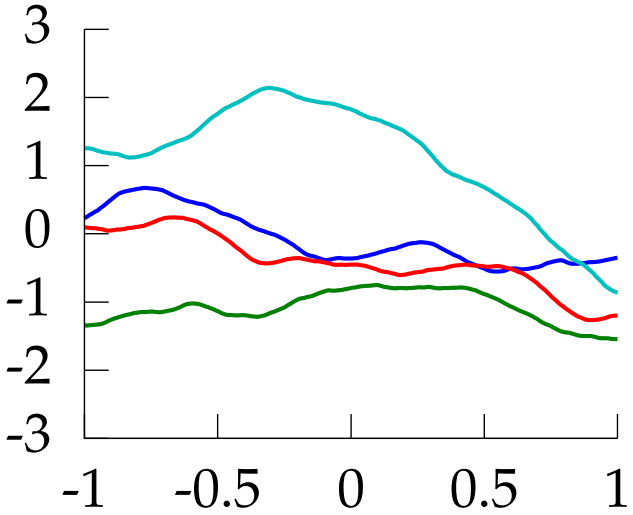
Covariance Functions

Where did this covariance matrix come from?

Matern 3/2 Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha\left(1 + \sqrt{3}r\right)\exp\left(-\sqrt{3}r\right) \quad \text{where} \quad r = \frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\ell}$$

- Matern 3/2 is a once differentiable covariance.
- Matern family constructed with Student-t filters in Fourier space.



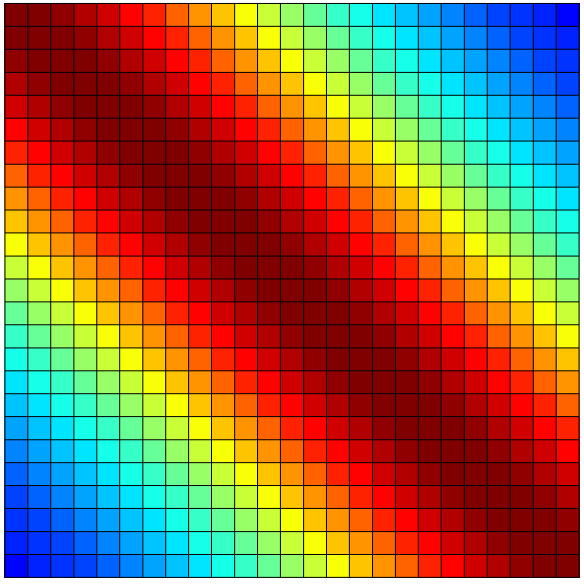
Covariance Functions

Where did this covariance matrix come from?

Matern 5/2 Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha\left(1 + \sqrt{5}r + \frac{5}{3}r^2\right)\exp\left(-\sqrt{5}r\right) \quad \text{where} \quad r = \frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\ell}$$

- Matern 5/2 is a twice differentiable covariance.
- Matern family constructed with Student-t filters in Fourier space.



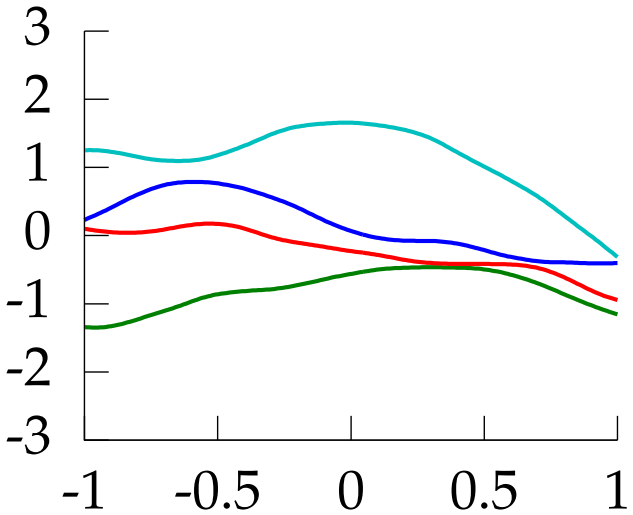
Covariance Functions

Where did this covariance matrix come from?

Matern 5/2 Covariance Function

$$k(\mathbf{x}, \mathbf{x}') = \alpha\left(1 + \sqrt{5}r + \frac{5}{3}r^2\right)\exp\left(-\sqrt{5}r\right) \quad \text{where} \quad r = \frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\ell}$$

- Matern 5/2 is a twice differentiable covariance.
- Matern family constructed with Student-t filters in Fourier space.



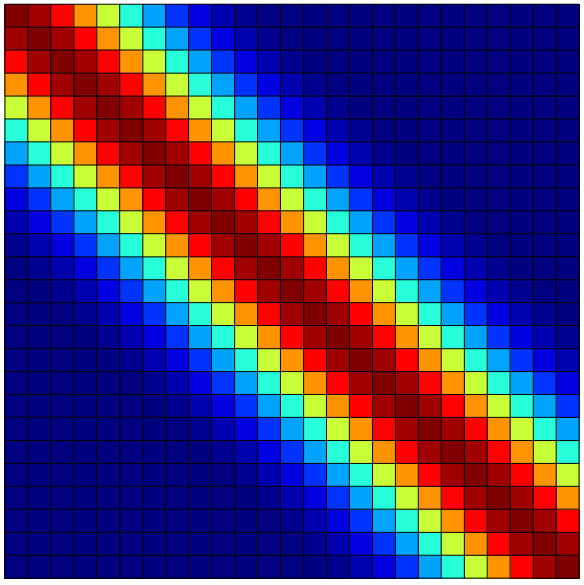
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



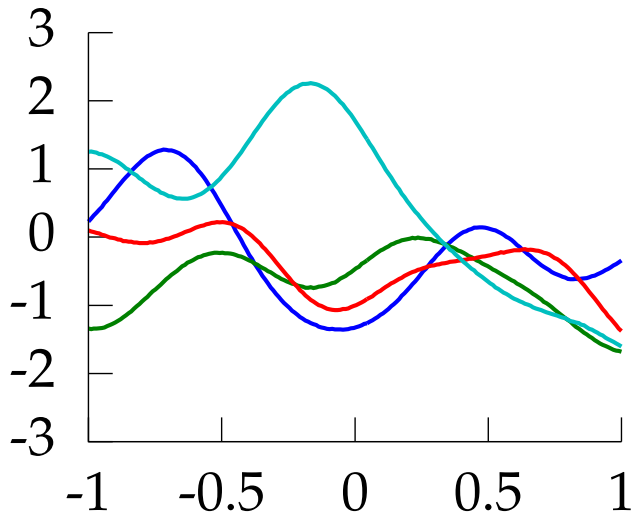
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



Outline

Motivation for Non-Linear Dimensionality Reduction

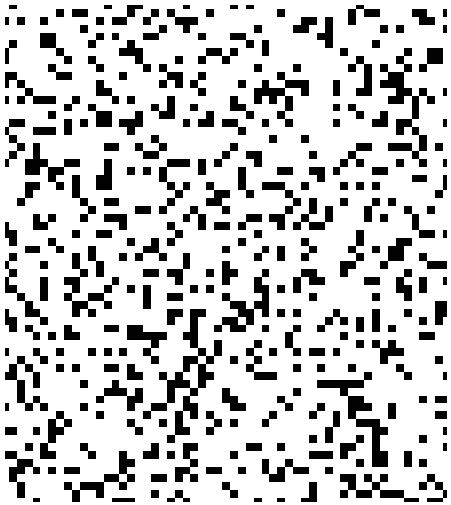
USPS Data Set Handwritten Digit

- 3648 Dimensions
  - 64 rows by 57 columns

Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

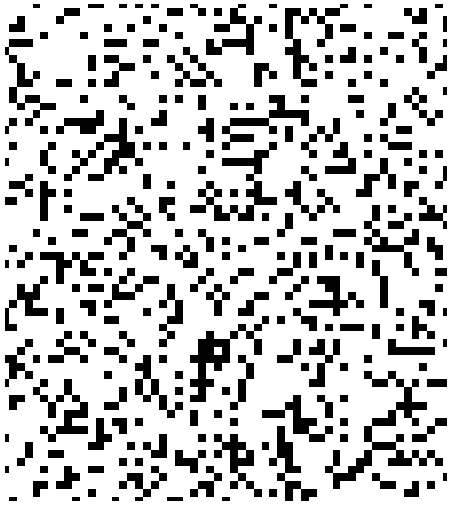
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.



Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

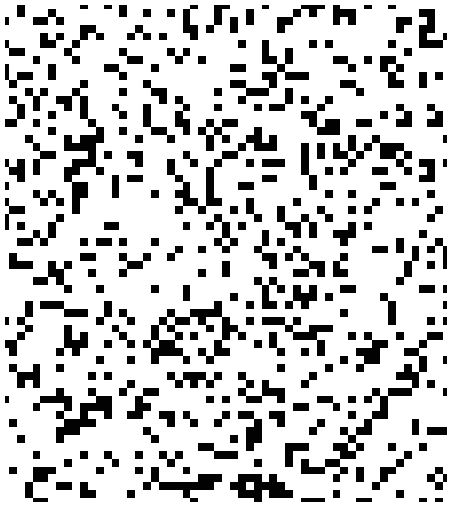
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

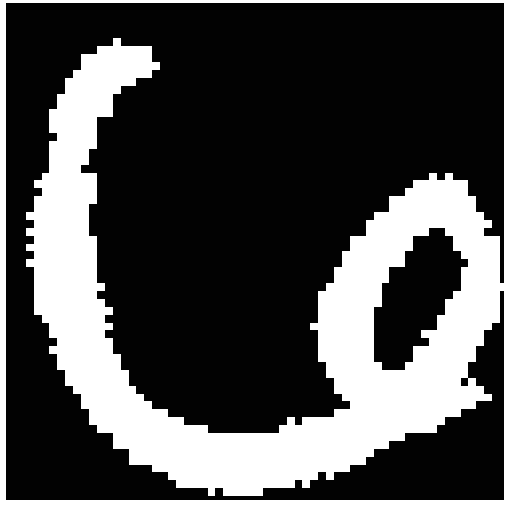
Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

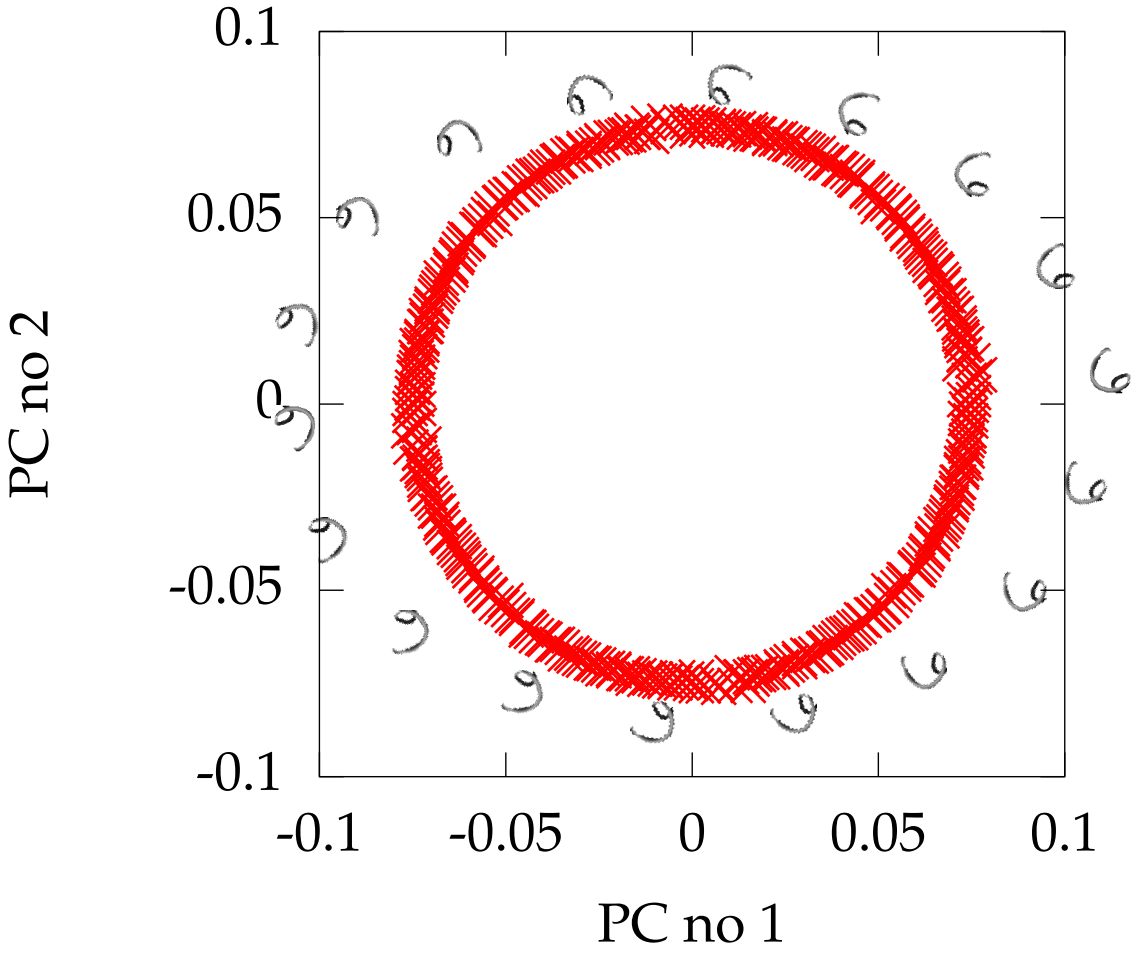


MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```

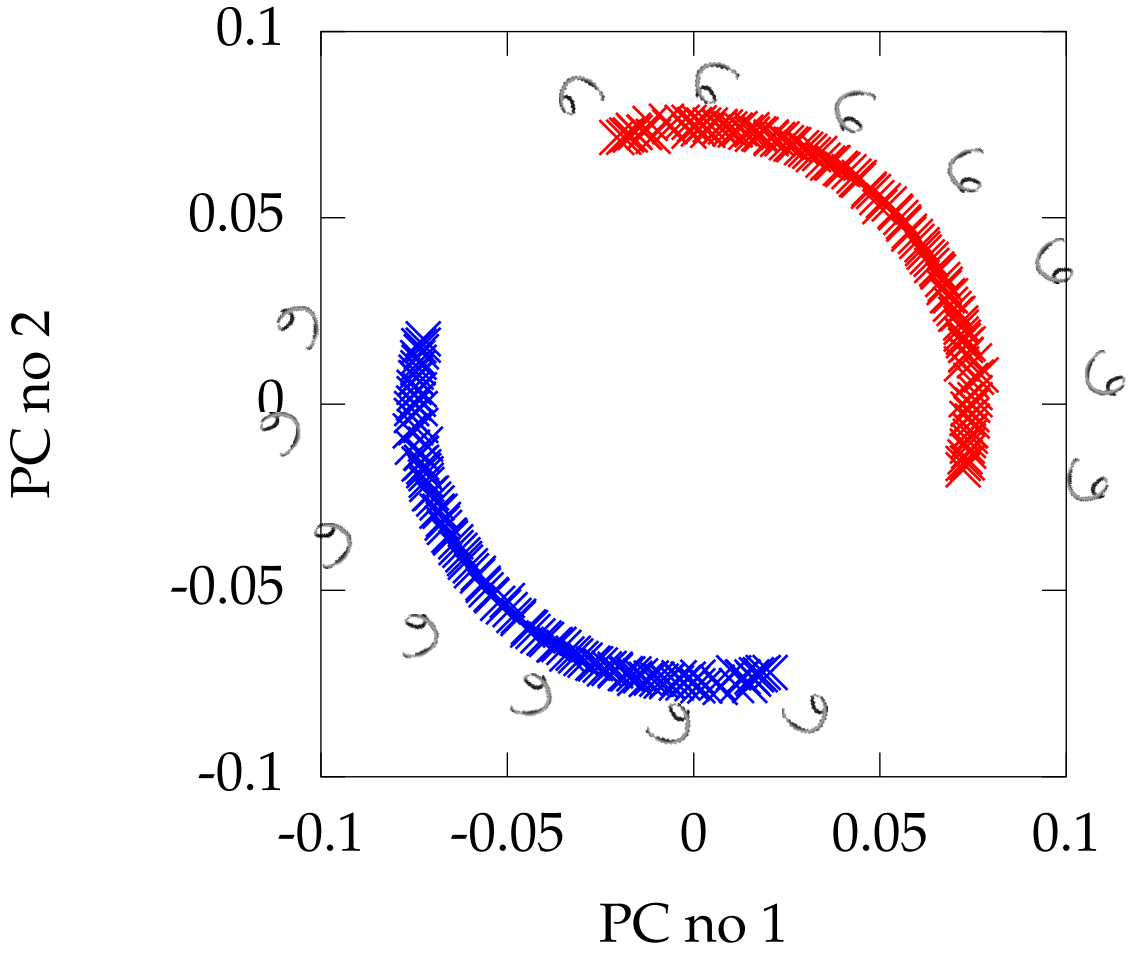
MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```



MATLAB Demo

```
demDigitsManifold([1 2], 'sixnine')
```



Low Dimensional Manifolds

Pure Rotation is too Simple

- In practice the data may undergo several distortions.
  - *e.g.* digits undergo 'thinning', translation and rotation.
- For data with 'structure':
  - we expect fewer distortions than dimensions;
  - we therefore expect the data to live on a lower dimensional manifold.
- Conclusion: deal with high dimensional data by looking for lower dimensional non-linear embedding.

Existing Methods

Spectral Approaches

- Classical Multidimensional Scaling (MDS) (Mardia et al., 1979).
  - Uses eigenvectors of similarity matrix.
    - Isomap (Tenenbaum et al., 2000) is MDS with a particular proximity measure.
  - Kernel PCA (Schölkopf et al., 1998)
    - Provides a representation and a mapping — dimensional expansion.
    - Mapping is implied throught he use of a kernel function as a similarity matrix.
  - Locally Linear Embedding (Roweis and Saul, 2000).
    - Looks to preserve locally linear relationships in a low dimensional space.

Existing Methods II

Iterative Methods

- Multidimensional Scaling (MDS)
  - Iterative optimisation of a stress function (Kruskal, 1964).
  - Sammon Mappings (Sammon, 1969).
    - Strictly speaking not a mapping — similar to iterative MDS.
- NeuroScale (Lowe and Tipping, 1997)
  - Augmentation of iterative MDS methods with a mapping.

Existing Methods III

Probabilistic Approaches

- Probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998)
  - A linear method.

Existing Methods III

Probabilistic Approaches

- Probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998)
  - A linear method.
- Density Networks (MacKay, 1995)
  - Use importance sampling and a multi-layer perceptron.

Existing Methods III

Probabilistic Approaches

- Probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998)
  - A linear method.
- Density Networks (MacKay, 1995)
  - Use importance sampling and a multi-layer perceptron.
- Generative Topographic Mapping (GTM) (Bishop et al., 1998)
  - Uses a grid based sample and an RBF network.

Existing Methods III

Probabilistic Approaches

- Probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998)
  - A linear method.
- Density Networks (MacKay, 1995)
  - Use importance sampling and a multi-layer perceptron.
- Generative Topographic Mapping (GTM) (Bishop et al., 1998)
  - Uses a grid based sample and an RBF network.

Difficulty for Probabilistic Approaches

- Propagate a probability distribution through a non-linear mapping.

The New Model

A Probabilistic Non-linear PCA

- PCA has a probabilistic interpretation (Tipping and Bishop, 1999; Roweis, 1998).
- It is difficult to 'non-linearise'.

Dual Probabilistic PCA

- We present a new probabilistic interpretation of PCA (Lawrence, 2005).
- This interpretation can be made non-linear.
- The result is non-linear probabilistic PCA.

Notation

q— dimension of latent/embedded space
p— dimension of data space
n— number of data points

centred data, $\mathbf{Y} = [\mathbf{y}_{1,:}, \ldots, \mathbf{y}_{n,:}]^\top = \left[\mathbf{y}_{:,1}, \ldots, \mathbf{y}_{:,p}\right] \in \Re^{n \times p}$
latent variables, $\mathbf{X} = [\mathbf{x}_{1,:}, \ldots, \mathbf{x}_{n,:}]^\top = \left[\mathbf{x}_{:,1}, \ldots, \mathbf{x}_{:,q}\right] \in \Re^{n \times q}$
mapping matrix, $\mathbf{W} \in \Re^{p \times q}$

$\mathbf{a}_{i,:}$ is a vector from the ith row of a given matrix $\mathbf{A}$
$\mathbf{a}_{:,j}$ is a vector from the jth row of a given matrix $\mathbf{A}$

Reading Notation

X and Y are *design matrices*

- Covariance given by $n^{-1}\mathbf{Y}^\top\mathbf{Y}$.
- Inner product matrix given by $\mathbf{Y}\mathbf{Y}^\top$.

Linear Dimensionality Reduction

Linear Latent Variable Model

- Represent data, $\mathbf{Y}$, with a lower dimensional set of latent variables $\mathbf{X}$.
- Assume a linear relationship of the form

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:},$$

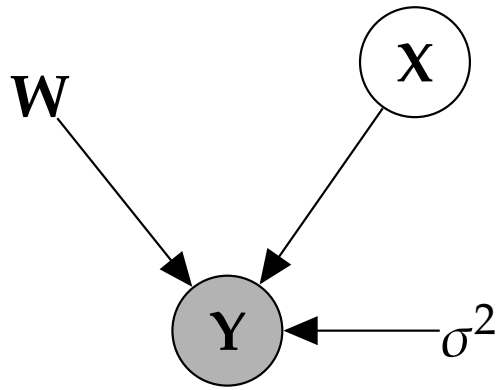
where

$$\boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right).$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.

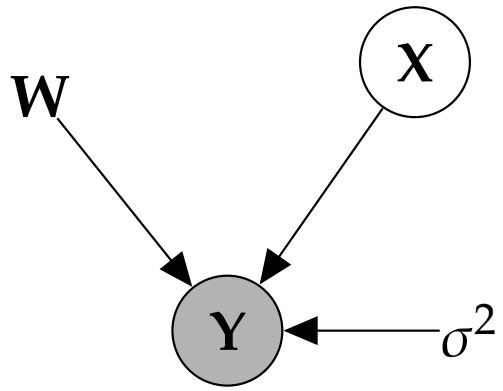


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:

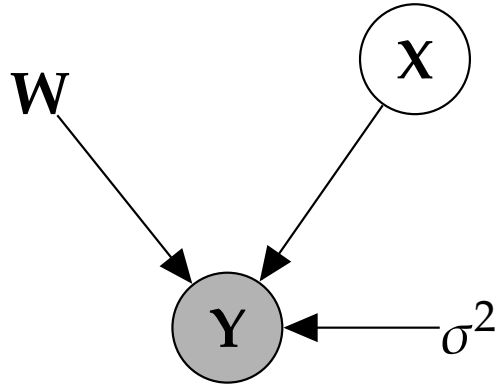


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.



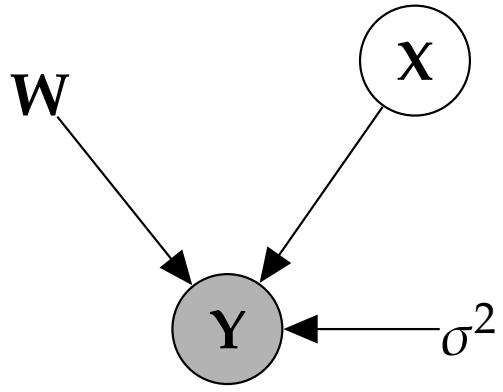
$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{X}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.



$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{X}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}, \quad \mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}, \quad \mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

$$\mathbf{W}\mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top),$$

Computation of the Marginal Likelihood

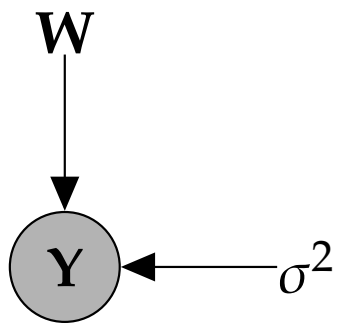
$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}, \quad \mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

$$\mathbf{W}\mathbf{x}_{i,:} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top),$$

$$\mathbf{W}\mathbf{x}_{i,:} + \epsilon_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)



$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

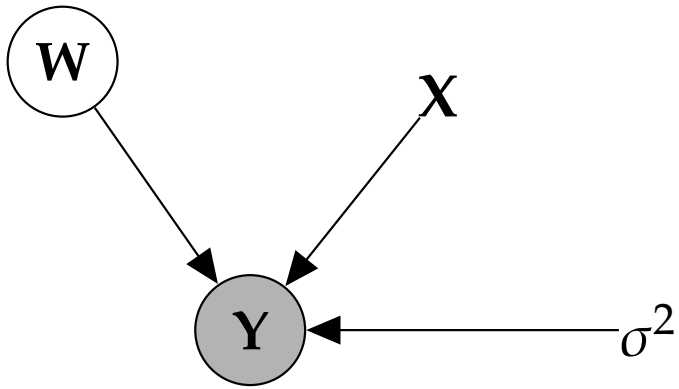
$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Linear Latent Variable Model III

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.

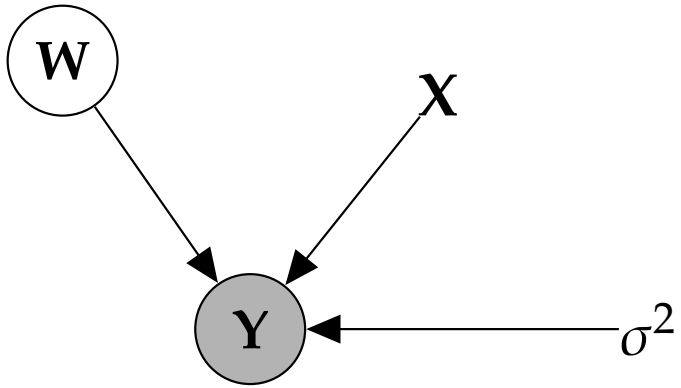


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach:

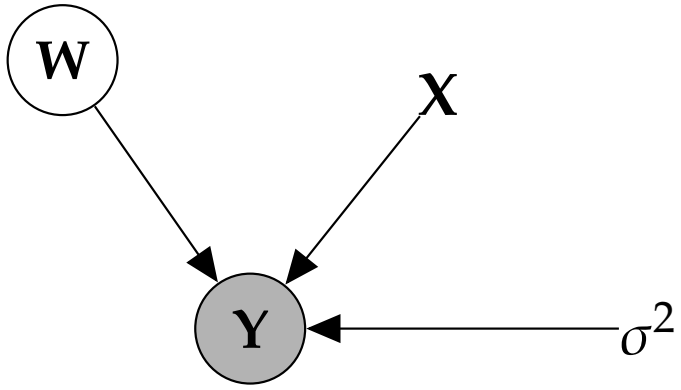


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach:
  - Define Gaussian prior over *parameters*, **W**.



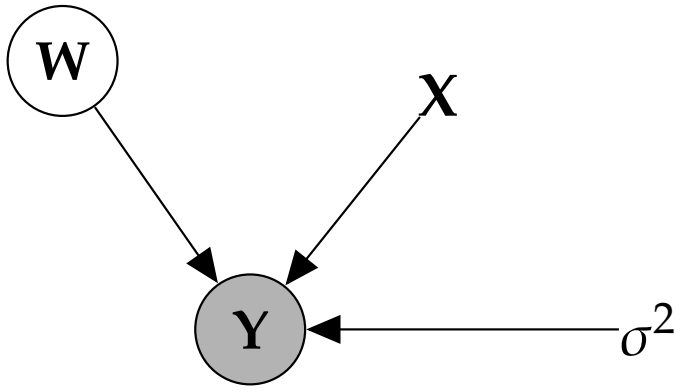
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{W}) = \prod_{i=1}^{p} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach:
  - Define Gaussian prior over *parameters*, $\mathbf{W}$.
  - Integrate out *parameters*.



$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{W}) = \prod_{i=1}^{p} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{:,j} = \mathbf{X}\mathbf{w}_{:,j} + \boldsymbol{\epsilon}_{:,j}, \quad \mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right)$$

Computation of the Marginal Likelihood

$$\mathbf{y}_{:,j} = \mathbf{X}\mathbf{w}_{:,j} + \boldsymbol{\epsilon}_{:,j}, \quad \mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right)$$

$$\mathbf{X}\mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{X}\mathbf{X}^\top),$$

Computation of the Marginal Likelihood

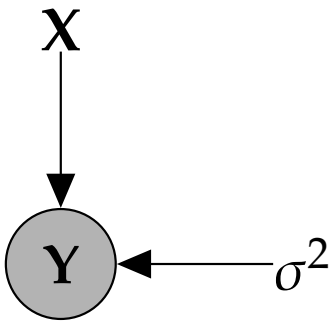
$$\mathbf{y}_{:,j} = \mathbf{X}\mathbf{w}_{:,j} + \boldsymbol{\epsilon}_{:,j}, \quad \mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\epsilon}_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$$

$$\mathbf{X}\mathbf{w}_{:,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{X}\mathbf{X}^\top),$$

$$\mathbf{X}\mathbf{w}_{:,j} + \epsilon_{:,j} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model IV

Dual Probabilistic PCA Max. Likelihood Soln (Lawrence, 2004, 2005)



$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model IV

Dual PPCA Max. Likelihood Soln (Lawrence, 2004, 2005)

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

Linear Latent Variable Model IV

PPCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{X}\right) = -\frac{p}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right) + \text{const.}$$

Linear Latent Variable Model IV

PPCA Max. Likelihood Soln

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{X}\right) = -\frac{p}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right) + \text{const.}$$

If $\mathbf{U}'_q$ are first q principal eigenvectors of $p^{-1}\mathbf{Y}\mathbf{Y}^{\top}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

Linear Latent Variable Model IV

PPCA Max. Likelihood Soln

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{X}\right) = -\frac{p}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right) + \text{const.}$$

If $\mathbf{U}'_q$ are first q principal eigenvectors of $p^{-1}\mathbf{Y}\mathbf{Y}^{\top}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

$$\mathbf{X} = \mathbf{U}'_q\mathbf{L}\mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Linear Latent Variable Model IV

Dual PPCA Max. Likelihood Soln (Lawrence, 2004, 2005)

$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right), \quad \mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{X}\right) = -\frac{p}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right) + \text{const.}$$

If $\mathbf{U}'_q$ are first q principal eigenvectors of $p^{-1}\mathbf{Y}\mathbf{Y}^{\top}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

$$\mathbf{X} = \mathbf{U}'_q\mathbf{L}\mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Linear Latent Variable Model IV

PPCA Max. Likelihood Soln (Tipping and Bishop, 1999)

$$p\left(\mathbf{Y}|\mathbf{W}\right) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0}, \mathbf{C}\right), \quad \mathbf{C} = \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}$$

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are $\mathbf{\Lambda}_q$,

$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{R}^{\top}, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ is an arbitrary rotation matrix.

Equivalence of Formulations

The Eigenvalue Problems are equivalent

- Solution for Probabilistic PCA (solves for the mapping)

$$\mathbf{Y}^\top \mathbf{Y}\mathbf{U}_q = \mathbf{U}_q \mathbf{\Lambda}_q \qquad \mathbf{W} = \mathbf{U}_q \mathbf{L}\mathbf{R}^\top$$

- Solution for Dual Probabilistic PCA (solves for the latent positions)

$$\mathbf{Y}\mathbf{Y}^\top \mathbf{U}'_q = \mathbf{U}'_q \mathbf{\Lambda}_q \qquad \mathbf{X} = \mathbf{U}'_q \mathbf{L}\mathbf{R}^\top$$

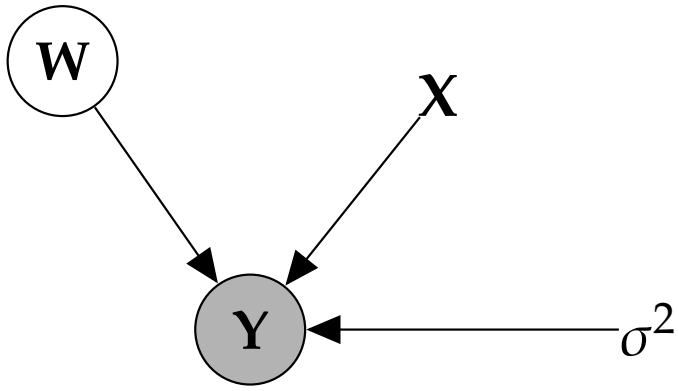
- Equivalence is from

$$\mathbf{U}_q = \mathbf{Y}^\top \mathbf{U}'_q \mathbf{\Lambda}_q^{-\frac{1}{2}}$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach:
  - Define Gaussian prior over *parameteters*, $\mathbf{W}$.
  - Integrate out *parameters*.



$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

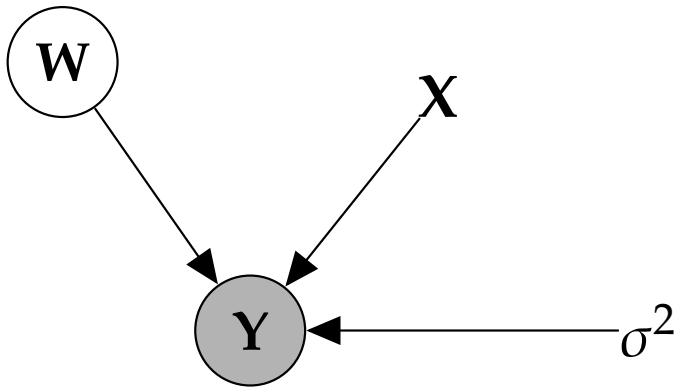
$$p(\mathbf{W}) = \prod_{i=1}^{p} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right)$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...

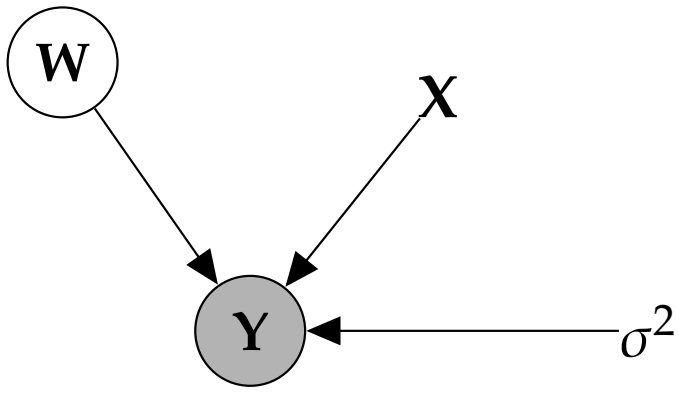


$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right)$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.



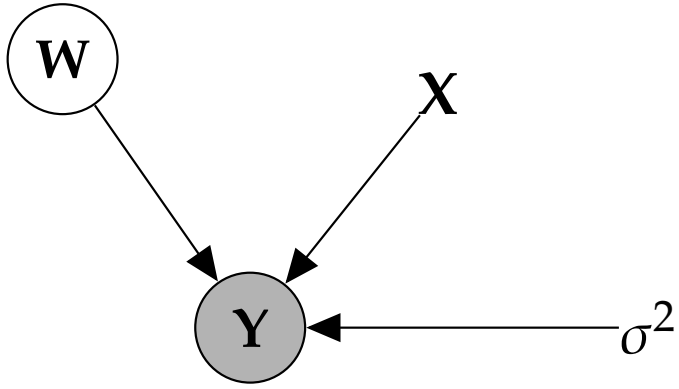
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$\mathbf{K} = \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.



$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

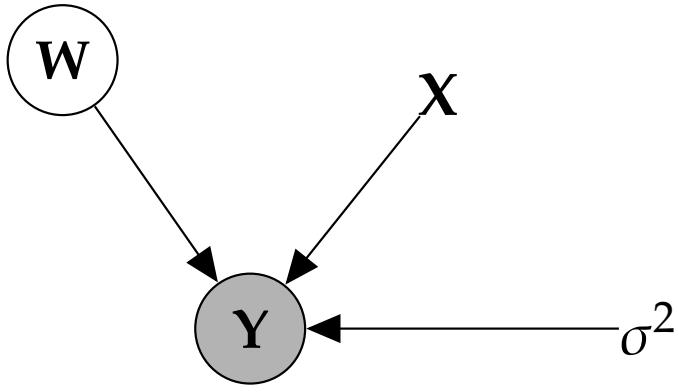
$$\mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

This is a product of Gaussian processes with linear kernels.

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.
  - We call this the Gaussian Process Latent Variable model (GP-LVM).



$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$\mathbf{K}$ =?

Replace linear kernel with non-linear kernel for non-linear model.

Non-linear Latent Variable Models

Exponentiated Quadratic (EQ) Covariance

- The EQ covariance has the form $k_{i,j} = k\left(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}\right)$, where

$$k\left(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}\right) = \alpha \exp\left(-\frac{\left\|\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\right\|_2^2}{2\ell^2}\right).$$

- No longer possible to optimise wrt $\mathbf{X}$ via an eigenvalue problem.
- Instead find gradients with respect to $\mathbf{X}$, α, ℓ and σ^2 and optimise using conjugate gradients.

Applications

Style Based Inverse Kinematics

- Facilitating animation through modeling human motion (Grochow et al., 2004)

Tracking

- Tracking using human motion models (Urtasun et al., 2005, 2006)

Assisted Animation

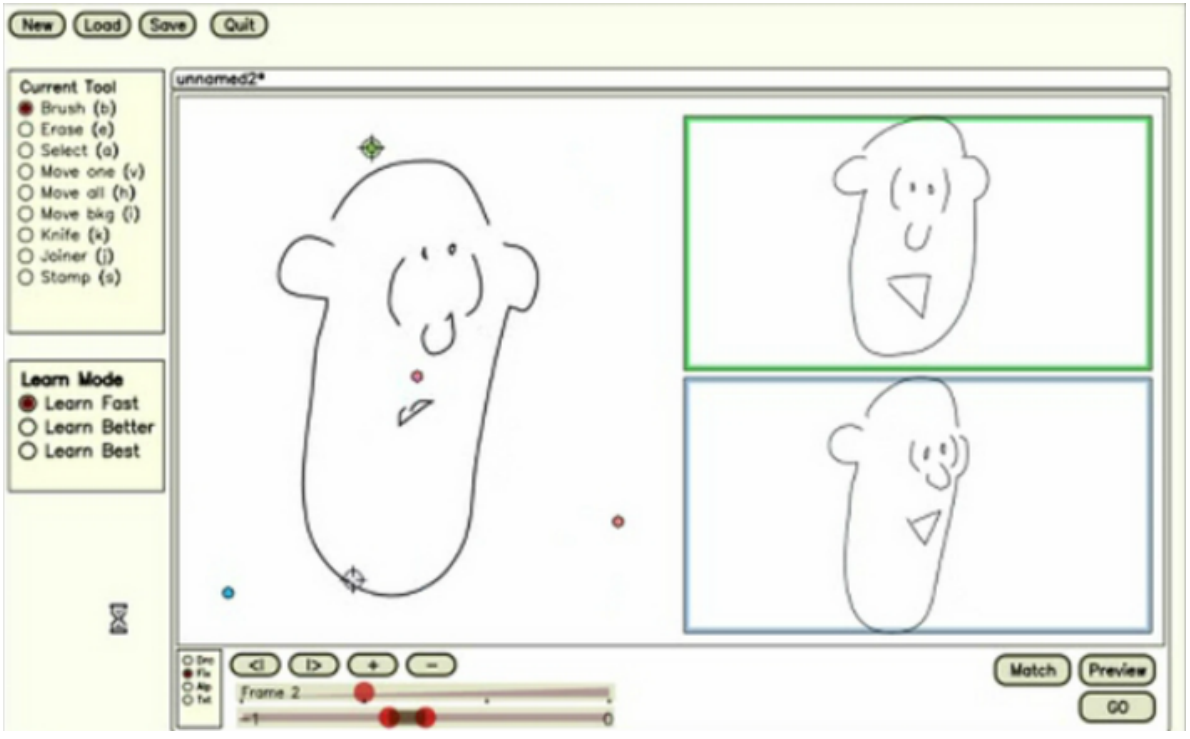
- Generalizing drawings for animation (Baxter and Anjyo, 2006)

Shape Models

- Inferring shape (e.g. pose from silhouette). (Ek et al., 2008b,a; Priacuriu and Reid, 2011a,b)

Example: Latent Doodle Space

(Baxter and Anjyo, 2006)



http://vimeo.com/3235882

Example: Latent Doodle Space

(Baxter and Anjyo, 2006)

Generalization with much less Data than Dimensions

- Powerful uncertainly handling of GPs leads to surprising properties.
- Non-linear models can be used where there are fewer data points than dimensions *without overfitting*.

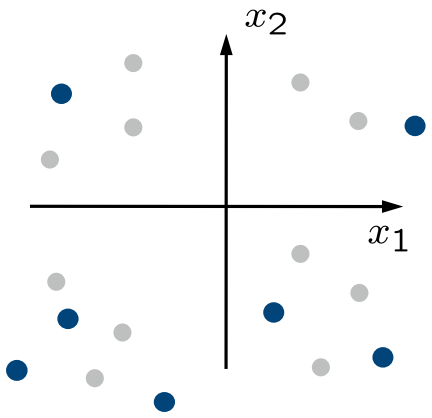
Prior for Supervised Learning

(Urtasun and Darrell, 2007)

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2}\text{tr}\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\},$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



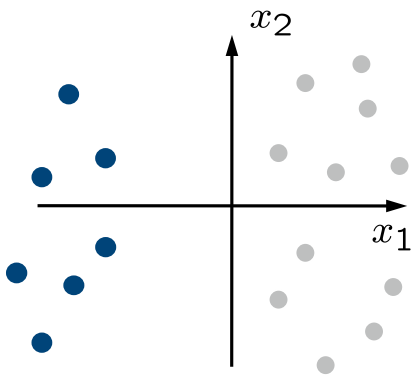
Prior for Supervised Learning

(Urtasun and Darrell, 2007)

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2}\mathrm{tr}\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\},$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



$$\mathbf{S}_w = \sum_{i=1}^{L} \frac{n_i}{n}(\mathbf{M}_i - \mathbf{M}_0)(\mathbf{M}_i - \mathbf{M}_0)^\top$$

where $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_{n_i}^{(i)}]$ are the n_i training points of class i, $\mathbf{M}_i$ is the mean of the elements of class i, and $\mathbf{M}_0$ is the mean of all the training points of all classes.

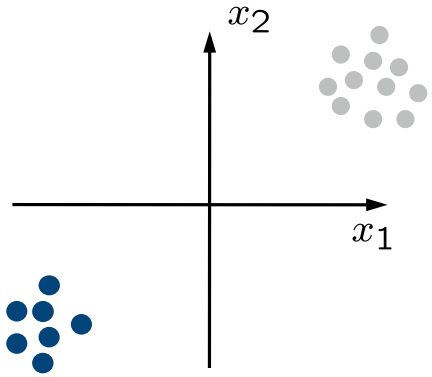
Prior for Supervised Learning

(Urtasun and Darrell, 2007)

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2}\text{tr}\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\},$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



$$\mathbf{S}_w = \sum_{i=1}^{L} \frac{n_i}{n}(\mathbf{M}_i - \mathbf{M}_0)(\mathbf{M}_i - \mathbf{M}_0)^\top$$

$$\mathbf{S}_b = \sum_{i=1}^{L} \frac{n_i}{n}\left[\frac{1}{n_i}\sum_{k=1}^{n_i}(\mathbf{x}_k^{(i)} - \mathbf{M}_i)(\mathbf{x}_k^{(i)} - \mathbf{M}_i)^\top\right]$$

where $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_{n_i}^{(i)}]$ are the n_i training points of class i, $\mathbf{M}_i$ is the mean of the elements of class i, and $\mathbf{M}_0$ is the

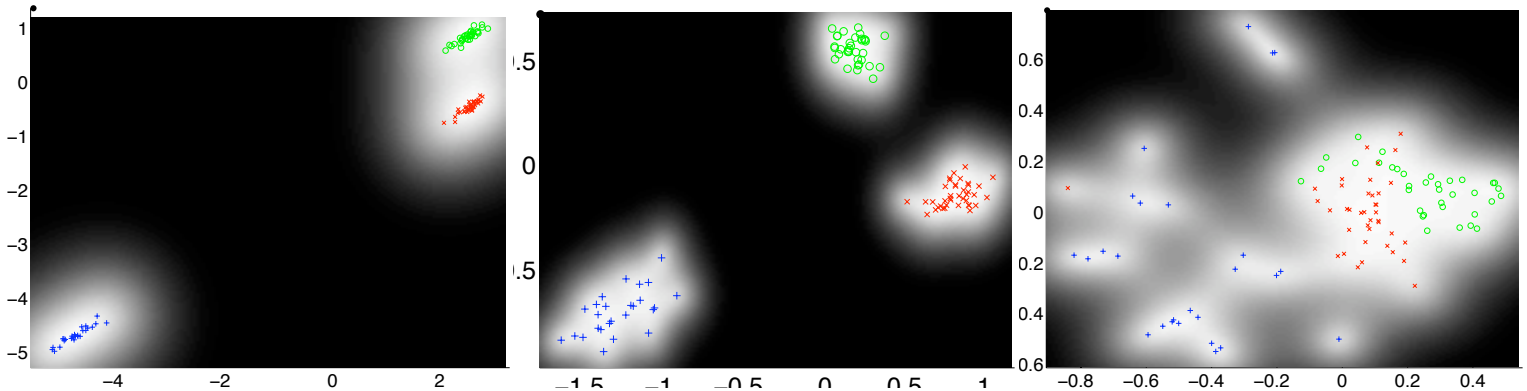
Prior for Supervised Learning

(Urtasun and Darrell, 2007)

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2}\text{tr}\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\},$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



GaussianFace

(Lu and Tang, 2014)

- First system to surpass human performance on cropped Learning Faces in Wild Data.
  `http://tinyurl.com/nkt9a38`
- Lots of feature engineering, followed by a Discriminative GP-LVM.

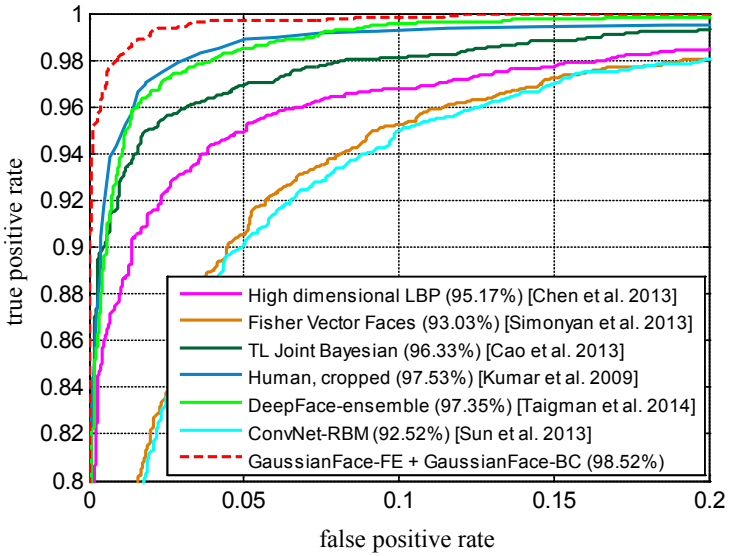


Figure 4: The ROC curve on LFW. Our method achieves the best performance, beating human-level performance.

Figure 5: The two rows present examples of matched and mismatched pairs respectively from LFW that were incorrectly classified by the GaussianFace model.

Conclusion and Future Work

Continuous Character Control

(Levine et al., 2012)

- Graph diffusion prior for enforcing connectivity between motions.

$$\log p(\mathbf{X}) = w_c \sum_{i,j} \log K^d_{ij}$$

with the graph diffusion kernel $\mathbf{K}^d$ obtain from

$$K^d_{ij} = \exp(\beta \mathbf{H}) \qquad \text{with} \qquad \mathbf{H} = -\mathbf{T}^{-1/2}\mathbf{L}\mathbf{T}^{-1/2}$$

the graph Laplacian, and $\mathbf{T}$ is a diagonal matrix with $T_{ii} = \sum_j w(\mathbf{x}_i, \mathbf{x}_j)$,

$$L_{ij} = \begin{cases} \sum_k w(\mathbf{x}_i, \mathbf{x}_k) & \text{if } i = j \\ -w(\mathbf{x}_i, \mathbf{x}_j) & \text{otherwise.} \end{cases}$$

and $w(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^{-p}$ measures similarity.

Character Control: Results

Other Topics

- Local distance preservation ▸ Details
- Dynamical models ▸ Details
- Hierarchical models ▸ Details
- Bayesian GP-LVM ▸ Details

Back Constraints I

Local Distance Preservation **(Lawrence and Quiñonero Candela, 2006)**

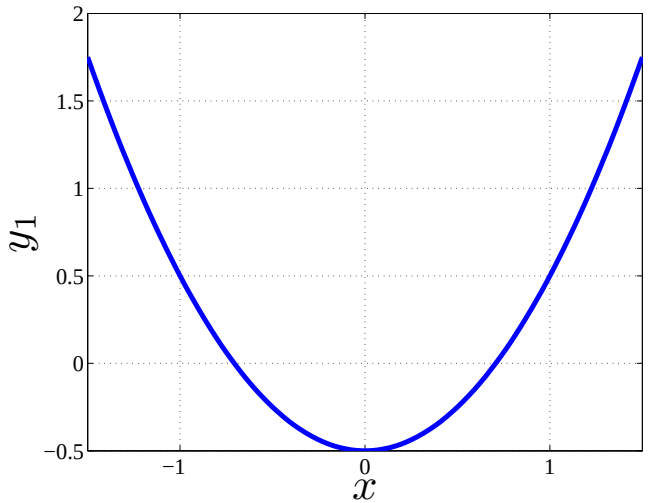
- Most dimensional reduction techniques preserve local distances.
- The GP-LVM does not.
- GP-LVM maps smoothly from latent to data space.
  - Points close in latent space are close in data space.
  - This does not imply points close in data space are close in latent space.
- Kernel PCA maps smoothly from data to latent space.
  - Points close in data space are close in latent space.
  - This does not imply points close in latent space are close in data space.

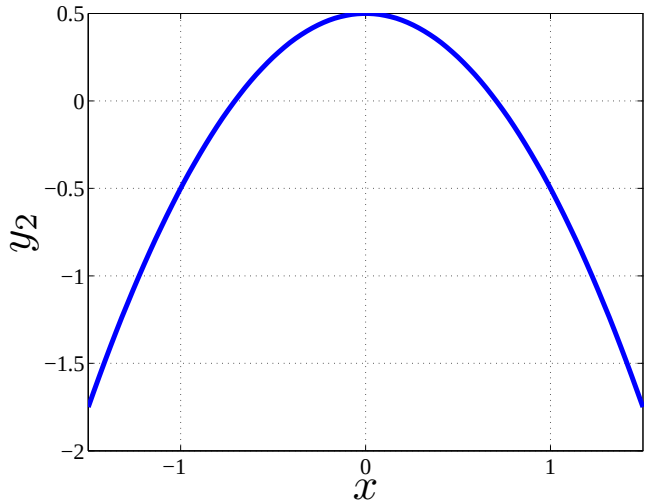
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 1-D latent space to 2-D data space.

$$y_1 = x^2 - 0.5, \quad y_2 = -x^2 + 0.5$$



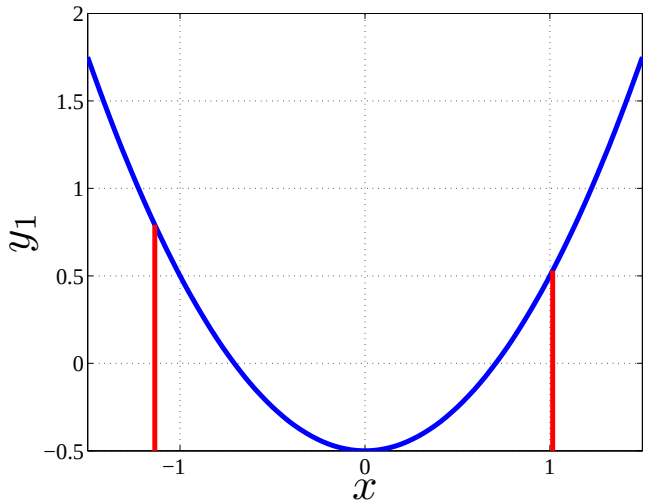


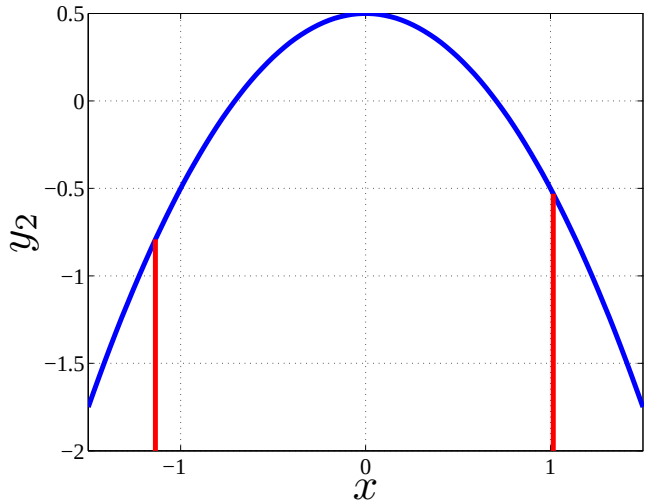
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 1-D latent space to 2-D data space.

$$y_1 = x^2 - 0.5, \quad y_2 = -x^2 + 0.5$$



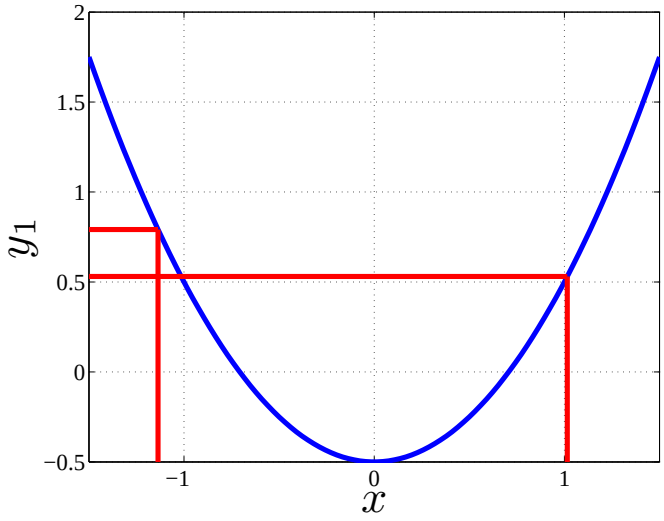


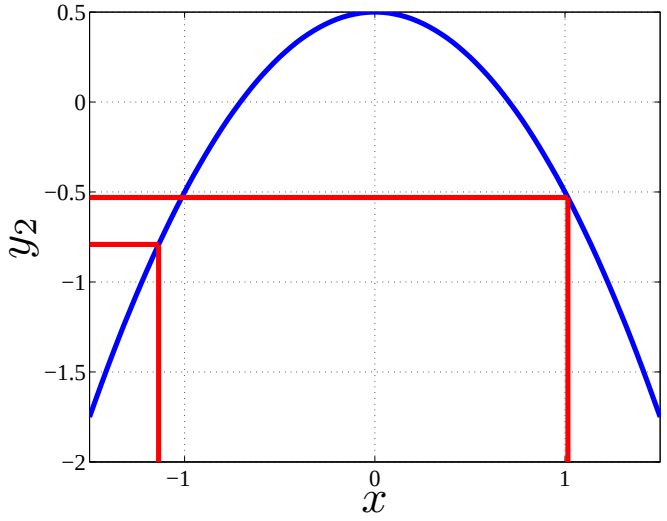
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 1-D latent space to 2-D data space.

$$y_1 = x^2 - 0.5, \quad y_2 = -x^2 + 0.5$$



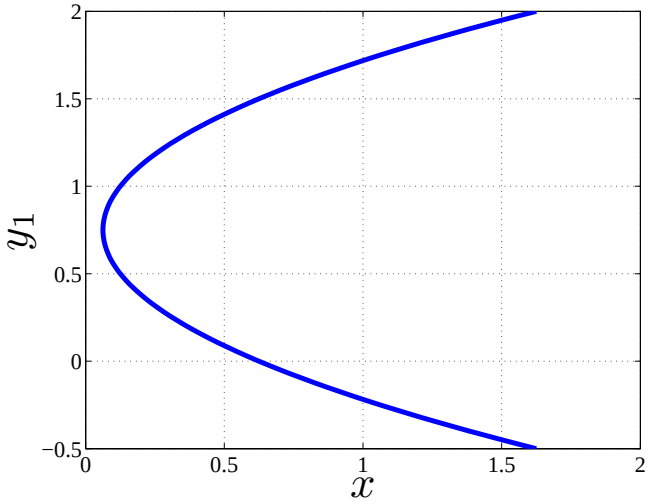


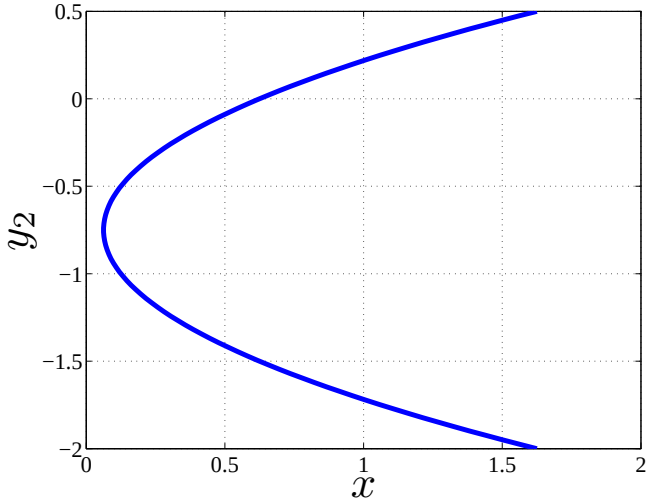
Back Constraints II

Backward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 2-D data space to 1-D latent.

$$x = 0.5\left(y_1^2 + y_2^2 + 1\right)$$



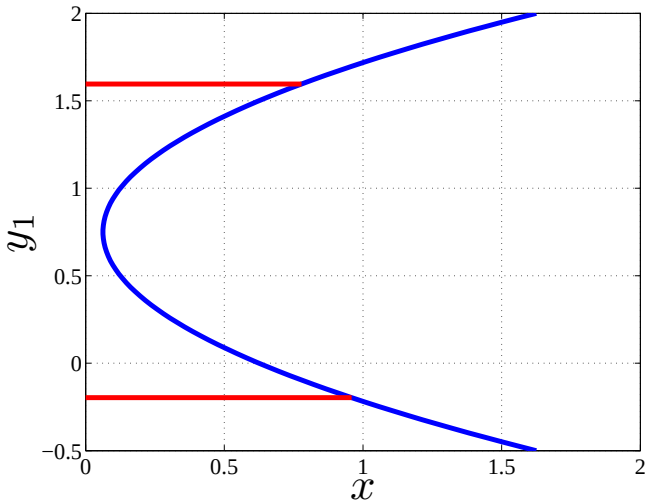


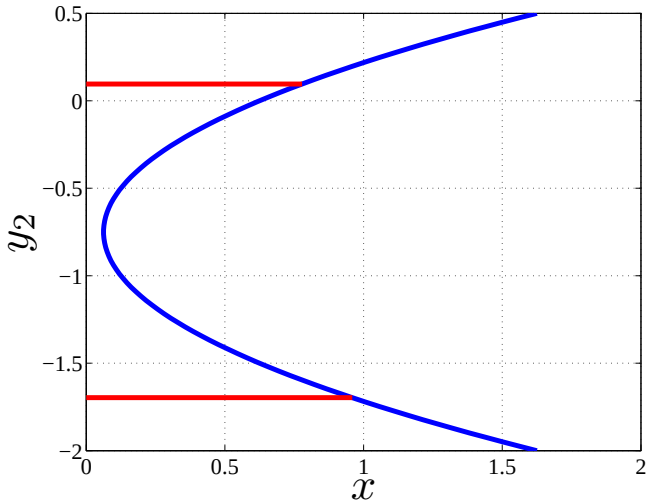
Back Constraints II

Backward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 2-D data space to 1-D latent.

$$x = 0.5\left(y_1^2 + y_2^2 + 1\right)$$



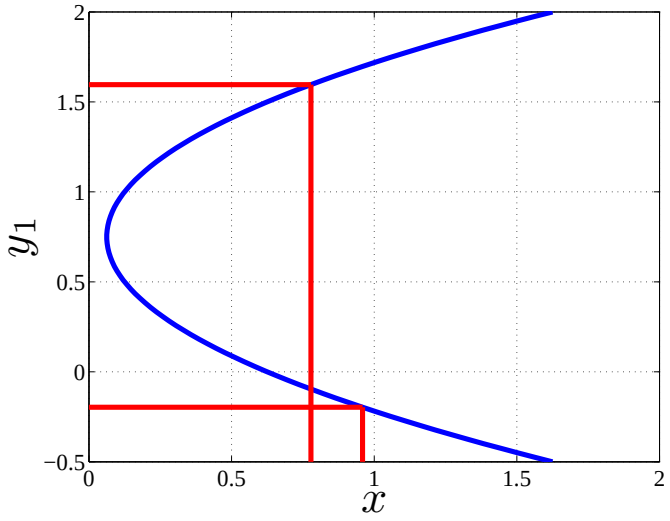


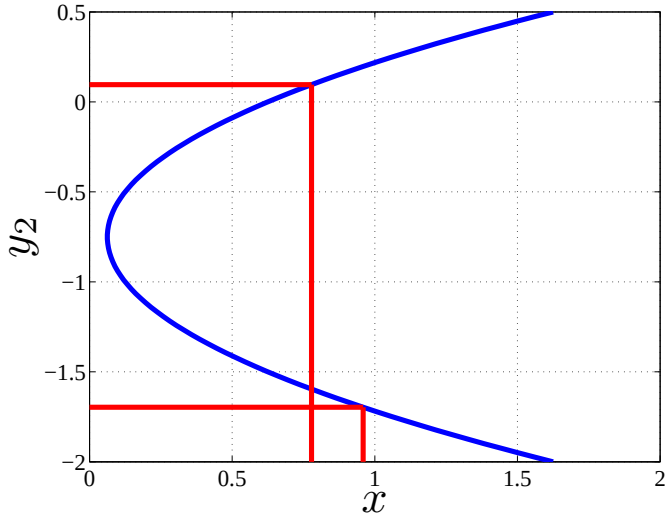
Back Constraints II

Backward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 2-D data space to 1-D latent.

$$x = 0.5\left(y_1^2 + y_2^2 + 1\right)$$





Multi-Dimensional Scaling with a Mapping

- Lowe and Tipping (1997) made latent positions a function of the data.

$$x_{i,j} = f_j(\mathbf{y}_{i,:}; \mathbf{v})$$

- Function was either multi-layer perceptron or a radial basis function network.
- Their motivation was different from ours:
  - They wanted to add the advantages of a true mapping to multi-dimensional scaling.

Back Constraints in the GP-LVM

Back Constraints

- We can use the same idea to force the GP-LVM to respect local distances.(Lawrence and Quiñonero Candela, 2006)
  - By constraining each $\mathbf{x}_i$ to be a 'smooth' mapping from $\mathbf{y}_i$ local distances can be respected.
- This works because in the GP-LVM we maximise wrt latent variables, we don't integrate out.
- Can use any 'smooth' function:
  1. Neural network.
  2. RBF Network.
  3. Kernel based mapping.

Optimising BC-GPLVM

Computing Gradients

- GP-LVM normally proceeds by optimising

$$L(\mathbf{X}) = \log p(\mathbf{Y}|\mathbf{X})$$

  with respect to $\mathbf{X}$ using $\frac{\mathrm{d}L}{\mathrm{d}\mathbf{X}}$.
- The back constraints are of the form

$$x_{i,j} = f_j(\mathbf{y}_{i,:}; \mathbf{v})$$

  where $\mathbf{v}$ are parameters.
- We can compute $\frac{\mathrm{d}L}{\mathrm{d}\mathbf{v}}$ via chain rule and optimise parameters of mapping.

Motion Capture Results

`demStick1` **and** `demStick3`

Figure: The latent space for the motion capture data with (*right*) and without (*left*) back constraints.

Motion Capture Results

`demStick1` **and** `demStick3`

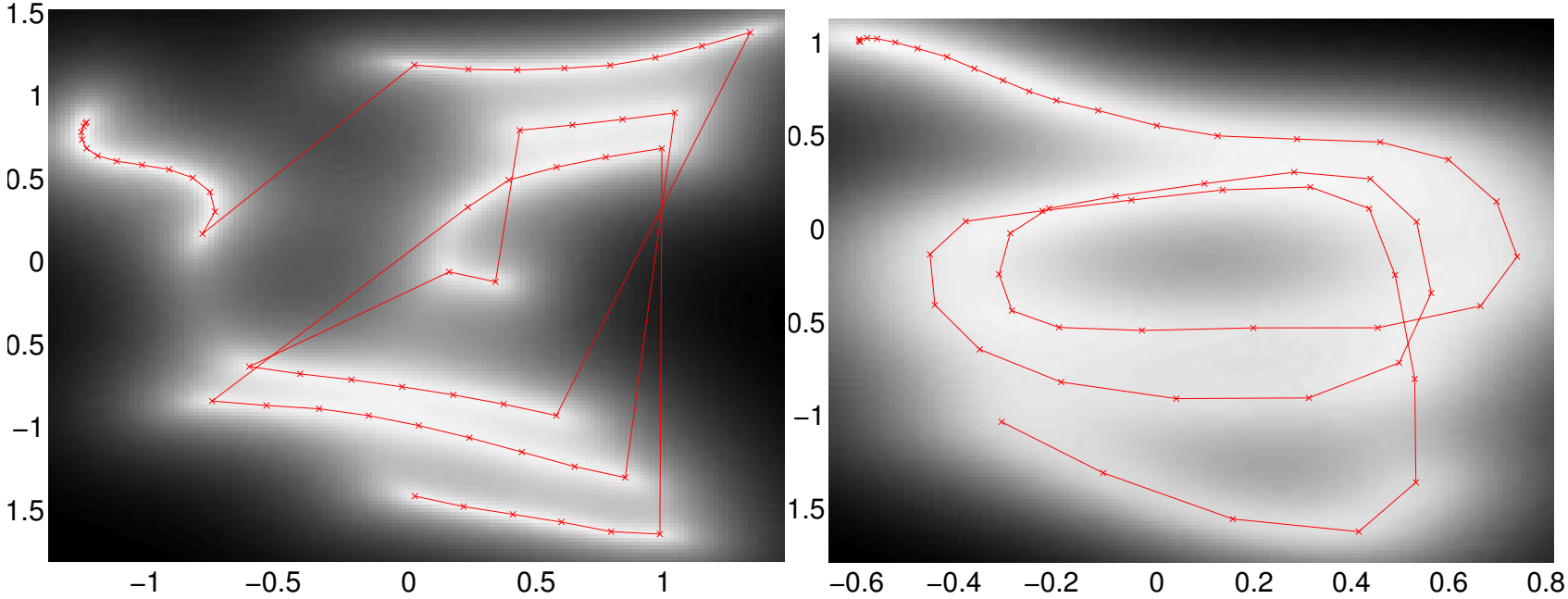
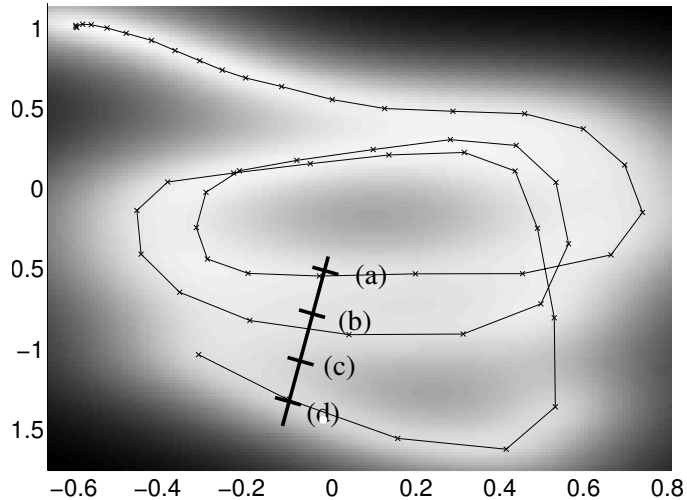


Figure: The latent space for the motion capture data with (*right*) and without (*left*) back constraints.

Stick Man Results

`demStickResults`



(a) (b) (c) (d)

Projection into data space from four points in the latent space. The inclination of the runner changes becoming more upright.

Adding Dynamics

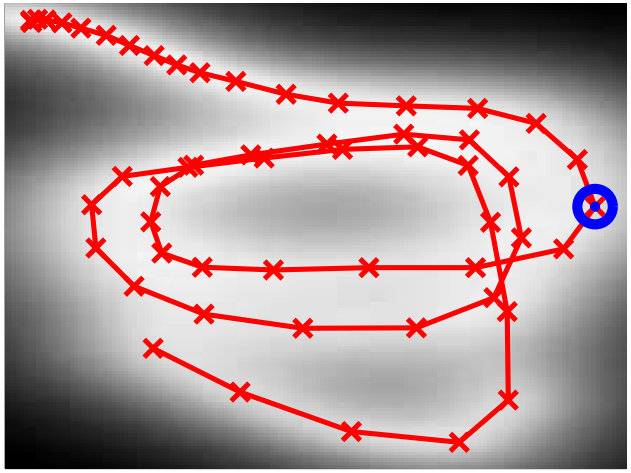
MAP Solutions for Dynamics Models

- Data often has a temporal ordering.
- Markov-based dynamics are often used.
- For the GP-LVM
  - Marginalising such dynamics is intractable.
  - But: MAP solutions are trivial to implement.
- Many choices: Kalman filter, Markov chains *etc*..
- Wang et al. (2006) suggest using a Gaussian Process.

Gaussian Process Dynamics

GP-LVM with Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.

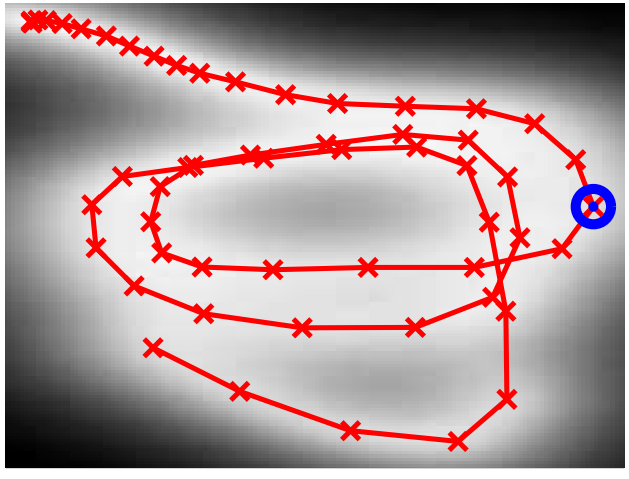


t

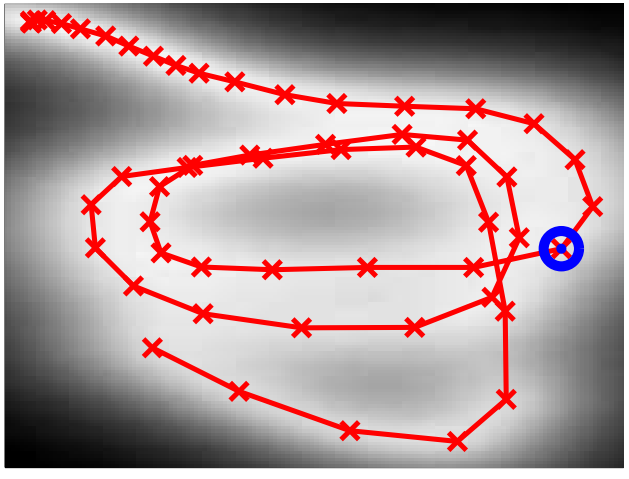
Gaussian Process Dynamics

GP-LVM with Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.



t

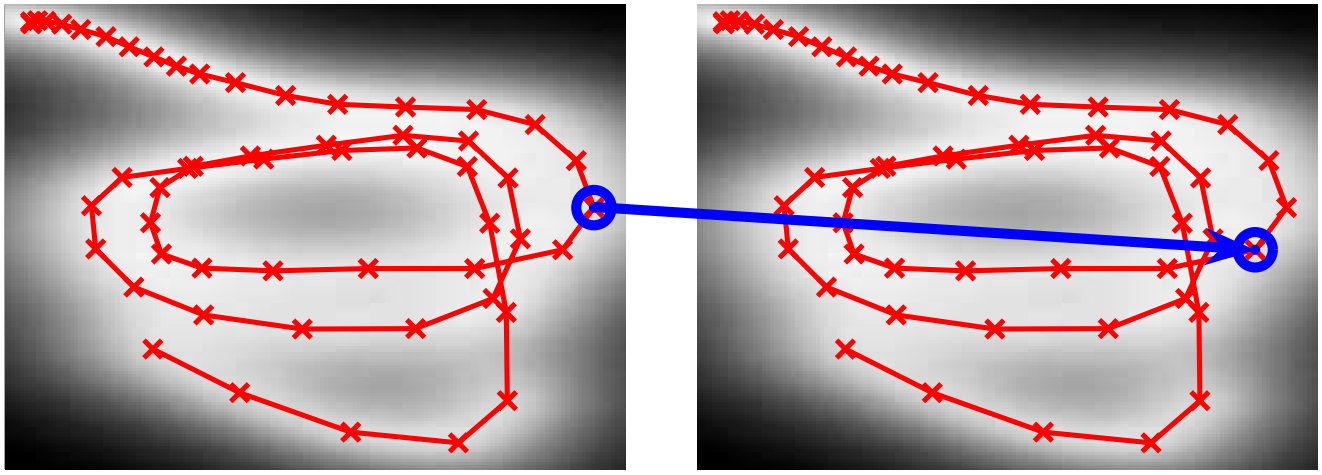


$t+1$

Gaussian Process Dynamics

GP-LVM with Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.



t $\quad$ $t+1$

Motion Capture Results

`demStick1` **and** `demStick2`

Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an exponentiated quadratic kernel.

Motion Capture Results

`demStick1` **and** `demStick2`

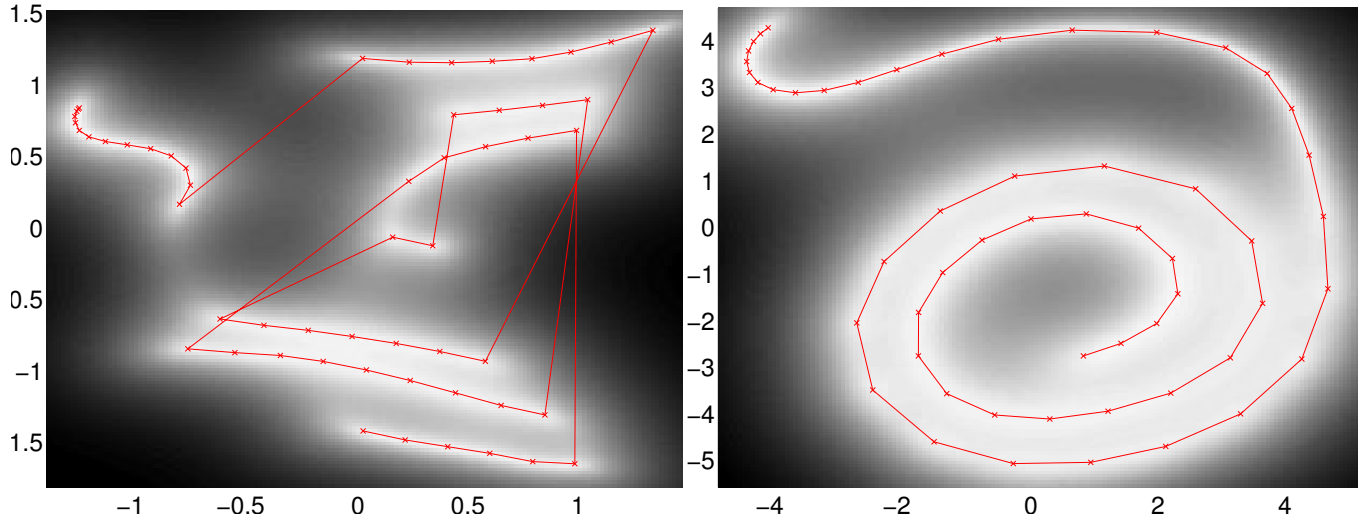
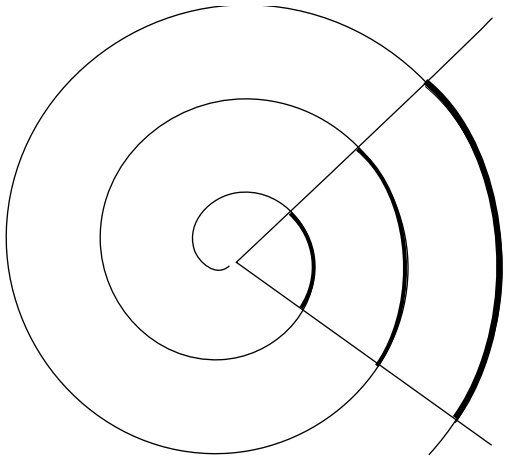


Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an exponentiated quadratic kernel.

Regressive Dynamics

Inner Groove Distortion

- Autoregressive unimodal dynamics, $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)$.
- Forces spiral visualisation.
- Poorer model due to inner groove distortion.



Regressive Dynamics

Direct use of Time Variable

- Instead of auto-regressive dynamics, consider regressive dynamics.
- Take $\mathbf{t}$ as an input, use a prior $p(\mathbf{X}|\mathbf{t})$.
- User a Gaussian process prior for $p(\mathbf{X}|\mathbf{t})$.
- Also allows us to consider variable sample rate data.

Motion Capture Results

`demStick1`, `demStick2` **and** `demStick5`

Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*middle*) and with regressive dynamics (*right*) based on an exponentiated quadratic kernel.

Motion Capture Results

`demStick1`, `demStick2` **and** `demStick5`

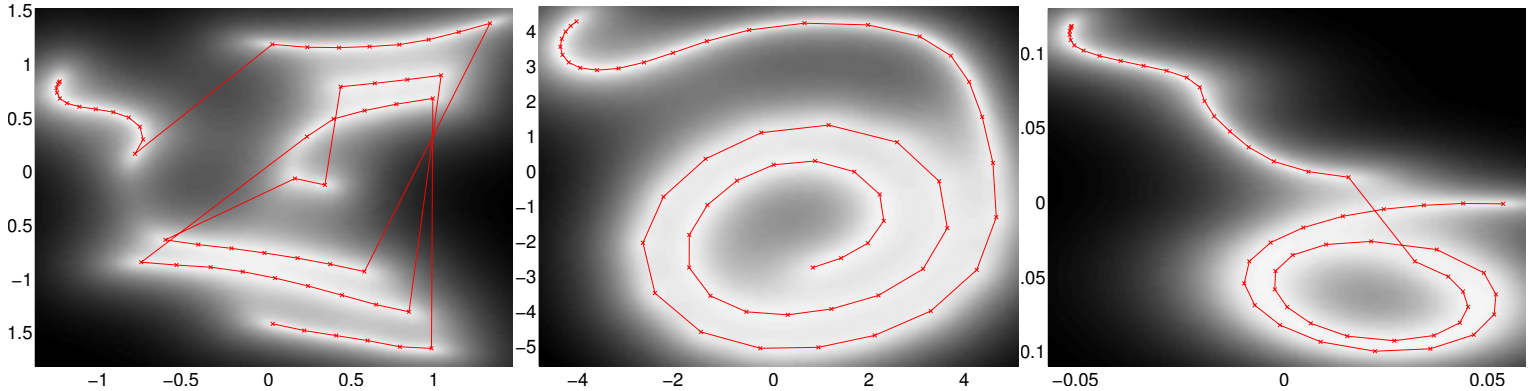


Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*middle*) and with regressive dynamics (*right*) based on an exponentiated quadratic kernel.

Return

Hierarchical GP-LVM

(Lawrence and Moore, 2007)

Stacking Gaussian Processes

- Regressive dynamics provides a simple hierarchy.
  - The input space of the GP is governed by another GP.
- By stacking GPs we can consider more complex hierarchies.
- Ideally we should marginalise latent spaces
  - In practice we seek MAP solutions.

Two Correlated Subjects

(Lawrence and Moore, 2007)

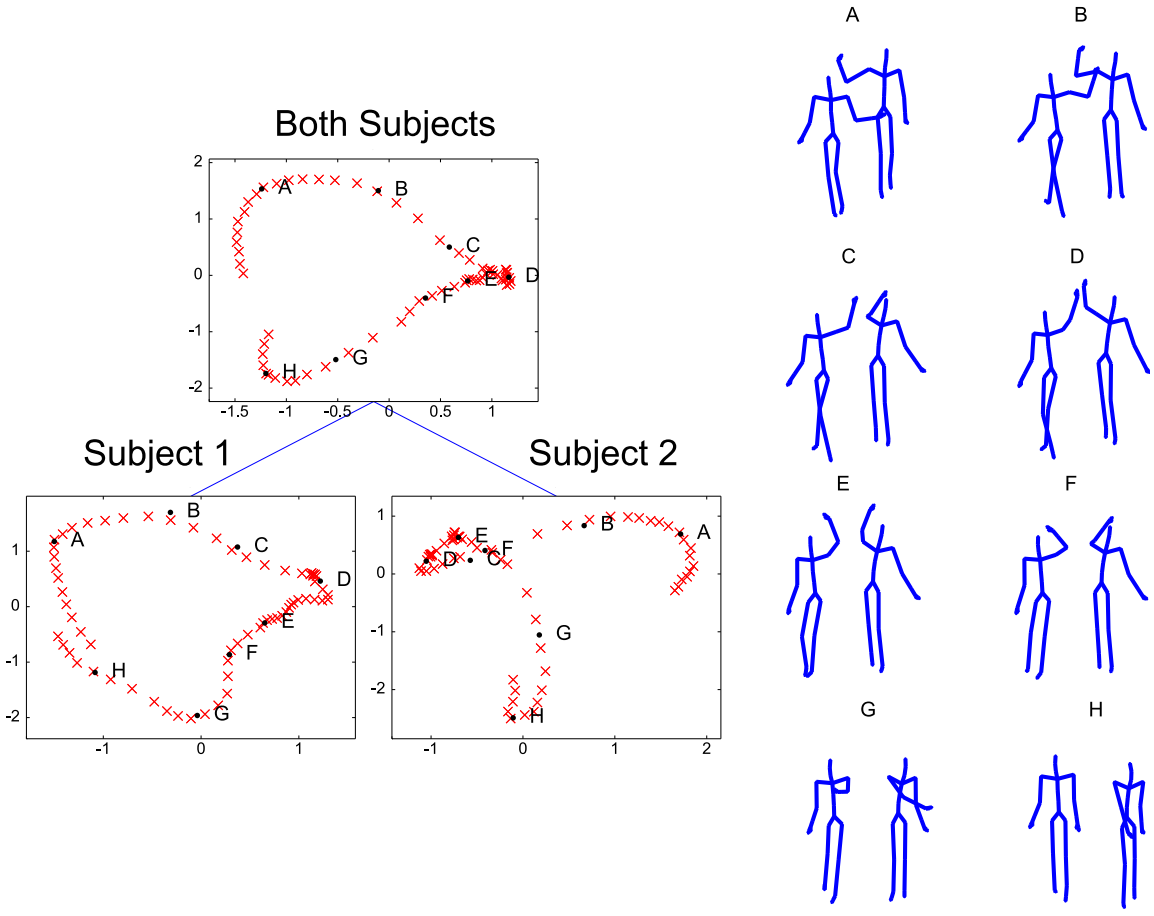


Figure: Hierarchical model of a 'high five'.

Within Subject Hierarchy

(Lawrence and Moore, 2007)

Decomposition of Body

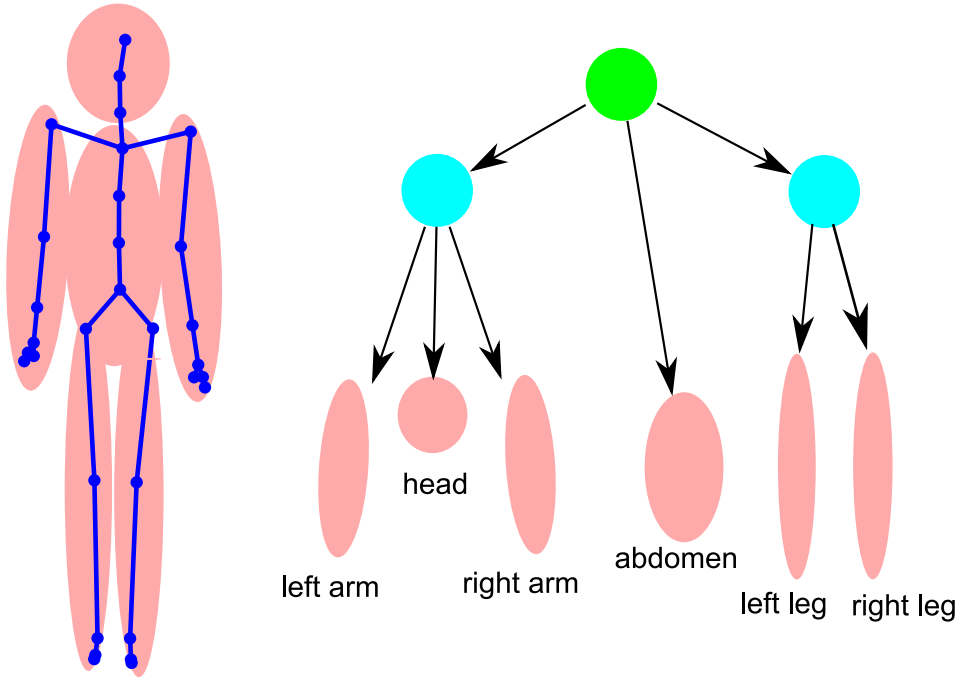


Figure: Decomposition of a subject.

Single Subject Run/Walk

(Lawrence and Moore, 2007)

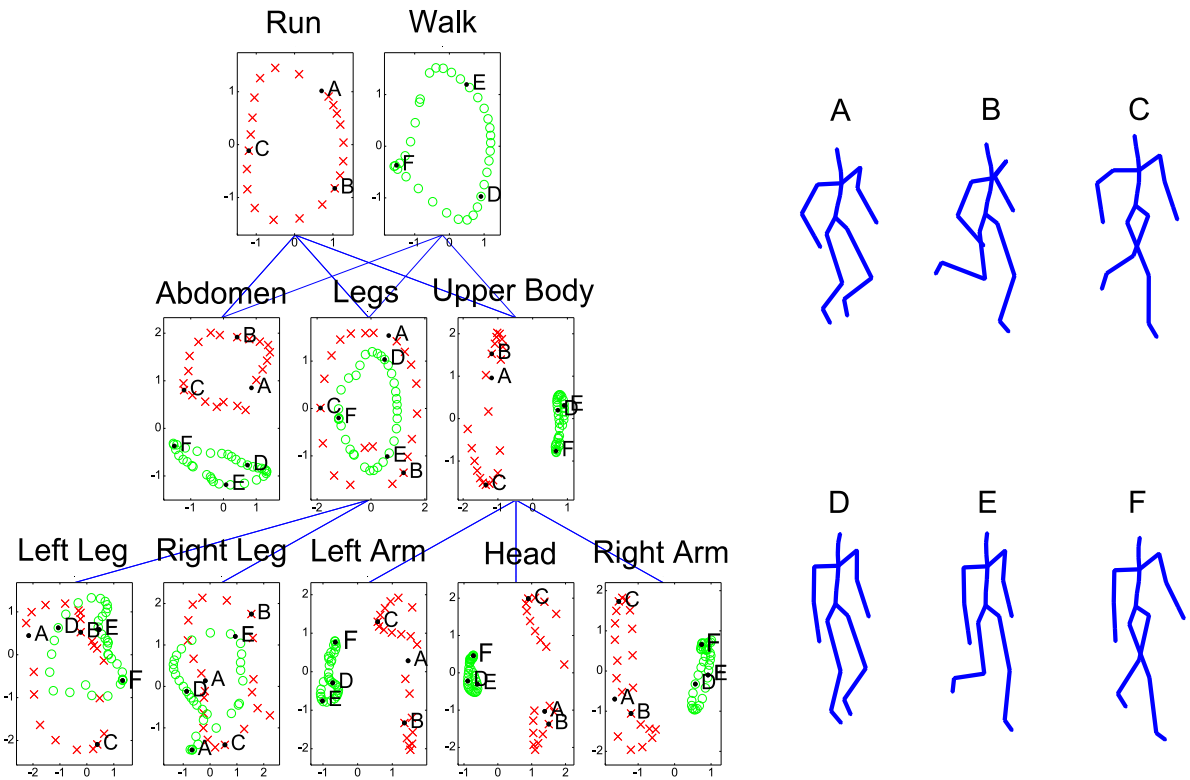


Figure: Hierarchical model of a walk and a run.

Return

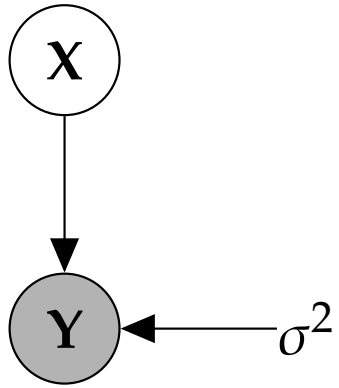
Selecting Data Dimensionality

- GP-LVM Provides probabilistic non-linear dimensionality reduction.
- How to select the dimensionality?
- Need to estimate marginal likelihood.
- In standard GP-LVM it increases with increasing q.

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.

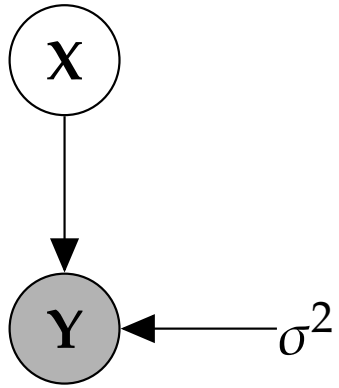


$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.

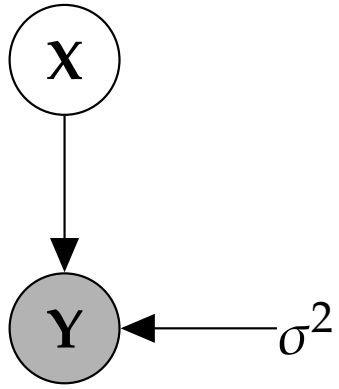


$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.



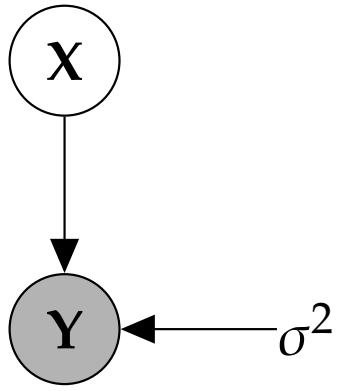
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \alpha_i^{-2}\mathbf{I}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.
  - Unfortunately integration is intractable.



$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \alpha_i^{-2}\mathbf{I}\right)$$

$$p(\mathbf{Y}|\boldsymbol{\alpha}) = ??$$

Standard Variational Approach Fails

- Standard variational bound has the form:

$$\mathcal{L} = \langle \log p(\mathbf{y}|\mathbf{X}) \rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \| p(\mathbf{X})\right)$$

Standard Variational Approach Fails

- Standard variational bound has the form:

$$\mathcal{L} = \left\langle \log p(\mathbf{y}|\mathbf{X}) \right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \,\|\, p(\mathbf{X})\right)$$

- Requires expectation of $\log p(\mathbf{y}|\mathbf{X})$ under $q(\mathbf{X})$.

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^\top \left(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right)^{-1} \mathbf{y} - \frac{1}{2}\log\left|\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right| - \frac{n}{2}\log 2\pi$$

Standard Variational Approach Fails

- Standard variational bound has the form:

$$\mathcal{L} = \left\langle \log p(\mathbf{y}|\mathbf{X}) \right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \,\|\, p(\mathbf{X})\right)$$

- Requires expectation of $\log p(\mathbf{y}|\mathbf{X})$ under $q(\mathbf{X})$.

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^\top \left(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y} - \frac{1}{2}\log\left|\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right| - \frac{n}{2}\log 2\pi$$

- Extremely difficult to compute because $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ is dependent on $\mathbf{X}$ and appears in the inverse.

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$p(\mathbf{y}) \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f} \right\rangle, \sigma^2 \mathbf{I}\right) p(\mathbf{u}) \mathrm{d}\mathbf{u}$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$p(\mathbf{y}|\mathbf{X}) \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{u})\mathrm{d}\mathbf{u}$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

- Apply variational lower bound to the inner integral.

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

- Apply variational lower bound to the inner integral.

$$\int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}$$

$$\geq \left\langle \sum_{i=1}^{n} \log c_i \right\rangle_{q(\mathbf{X})} + \left\langle \log \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right)\right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X})\,||\,p(\mathbf{X})\right)$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

- Apply variational lower bound to the inner integral.

$$\int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}$$

$$\geq \left\langle \sum_{i=1}^{n} \log c_i \right\rangle_{q(\mathbf{X})} + \left\langle \log \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right)\right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X})\,||\,p(\mathbf{X})\right)$$

- Which is analytically tractable for Gaussian $q(\mathbf{X})$ and some covariance functions.

Required Expectations

- Need expectations under $q(\mathbf{X})$ of:

$$\log c_i = \frac{1}{2\sigma^2}\left[k_{i,i} - \mathbf{k}_{i,\mathbf{u}}^\top \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{k}_{i,\mathbf{u}}\right]$$

and

$$\log \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{Y})}, \sigma^2\mathbf{I}\right) = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\left(y_i - \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}\right)^2$$

- This requires the expectations

$$\left\langle \mathbf{K}_{\mathbf{f},\mathbf{u}}\right\rangle_{q(\mathbf{X})}$$

and

$$\left\langle \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right\rangle_{q(\mathbf{X})}$$

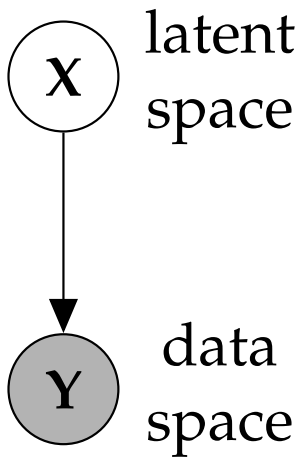
which can be computed analytically for some covariance functions.

Priors for Latent Space

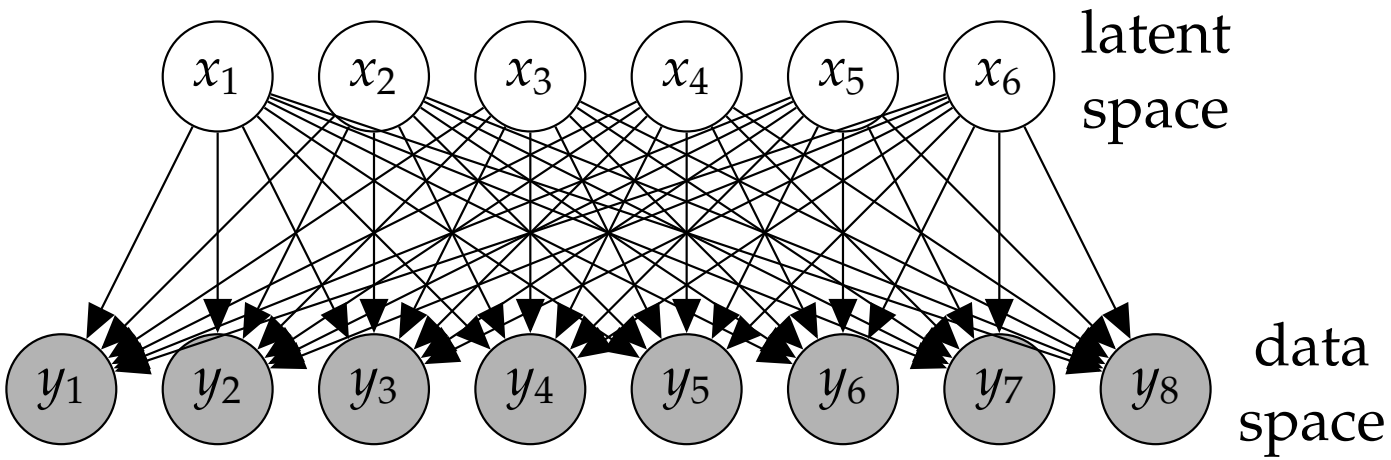
Titsias and Lawrence (2010)

- Variational marginalization of $\mathbf{X}$ allows us to learn parameters of $p(\mathbf{X})$.
- Standard GP-LVM where $\mathbf{X}$ learnt by MAP, this is not possible (see e.g. Wang et al., 2008).
- First example: learn the dimensionality of latent space.

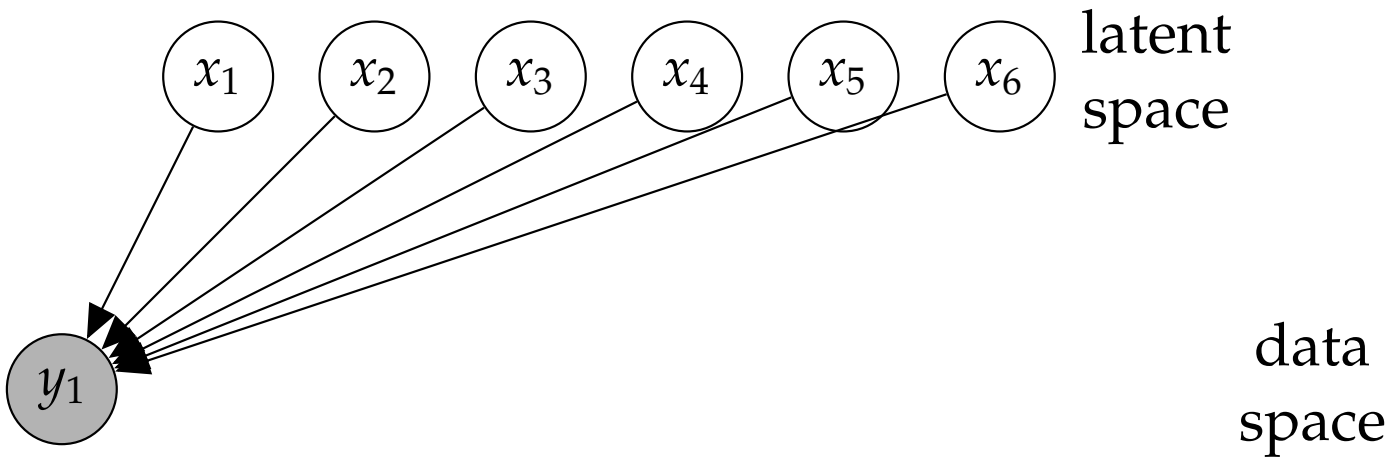
Graphical Representations of GP-LVM



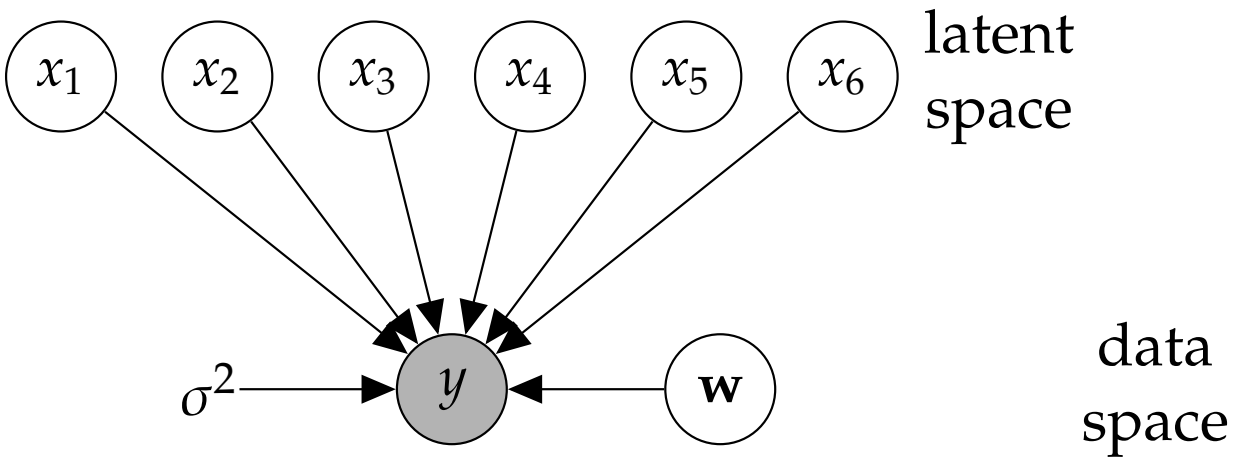
Graphical Representations of GP-LVM



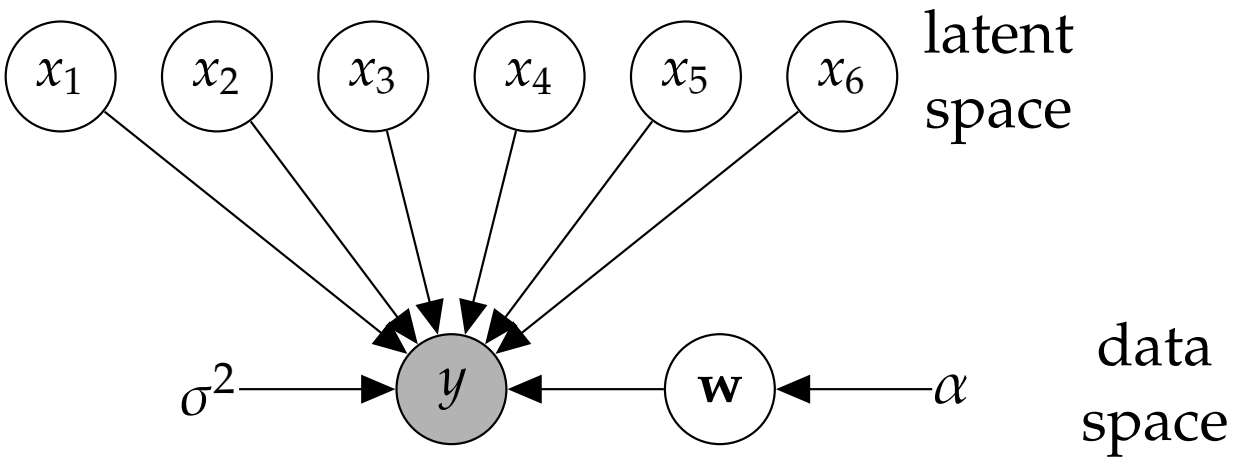
Graphical Representations of GP-LVM



Graphical Representations of GP-LVM



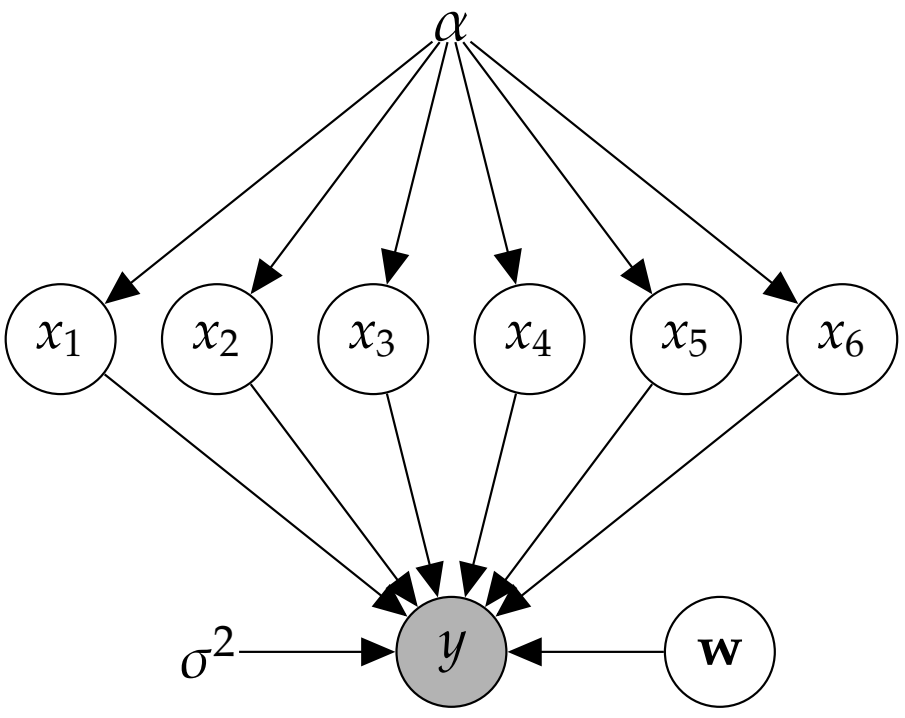
Graphical Representations of GP-LVM



$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \alpha\mathbf{I}) \quad \mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$$

$$y \sim \mathcal{N}\left(\mathbf{x}^\top \mathbf{w}, \sigma^2\right)$$

Graphical Representations of GP-LVM



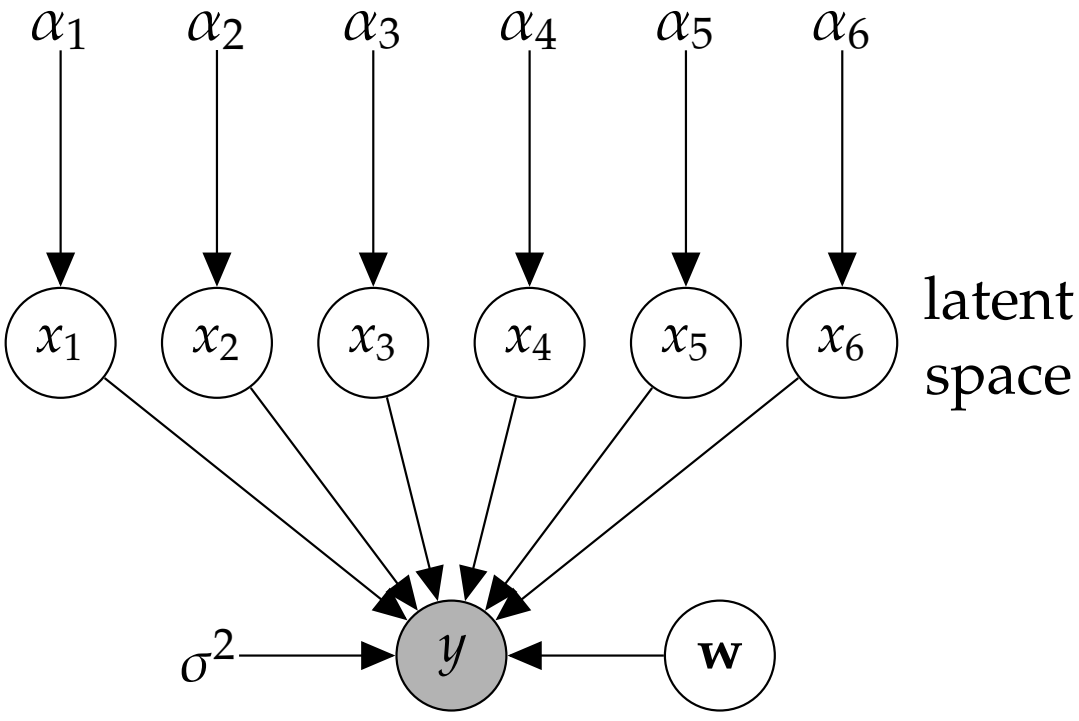
latent space

data space

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \mathbf{x} \sim \mathcal{N}(0, \alpha\mathbf{I})$$

$$y \sim \mathcal{N}\left(\mathbf{x}^\top \mathbf{w}, \sigma^2\right)$$

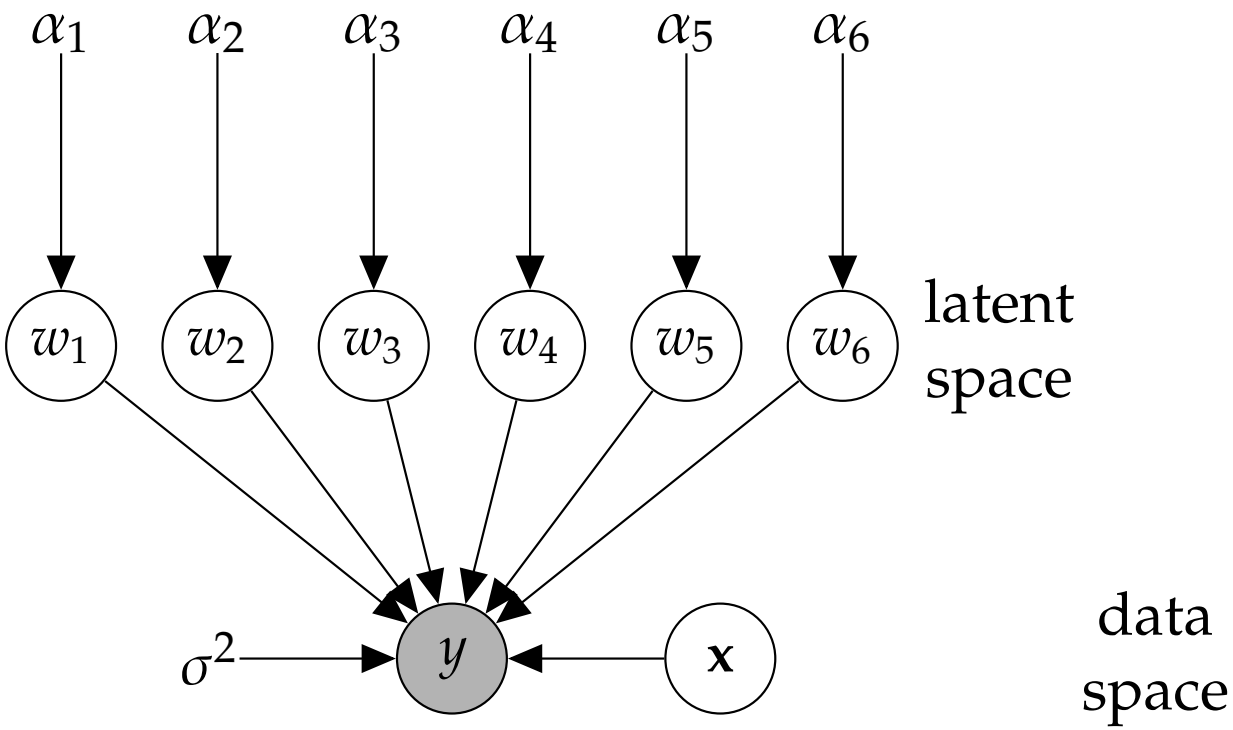
Graphical Representations of GP-LVM



$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad x_i \sim \mathcal{N}(0, \alpha_i)$$

$$y \sim \mathcal{N}\left(\mathbf{x}^\top \mathbf{w}, \sigma^2\right)$$

Graphical Representations of GP-LVM



$$w_i \sim \mathcal{N}(0, \alpha_i) \quad \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$y \sim \mathcal{N}\left(\mathbf{x}^\top \mathbf{w}, \sigma^2\right)$$

Non-linear $f(\mathbf{x})$

- In linear case equivalence because $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$

$$p(w_i) \sim \mathcal{N}(\mathbf{0}, \alpha_i)$$

- In non linear case, need to scale columns of $\mathbf{X}$ in prior for $f(\mathbf{x})$.
- This implies scaling columns of $\mathbf{X}$ in covariance function

$$k(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}) = \exp\left(-\frac{1}{2}(\mathbf{x}_{:,i} - \mathbf{x}_{:,j})^\top \mathbf{A}(\mathbf{x}_{:,i} - \mathbf{x}_{:,j})\right)$$

$\mathbf{A}$ is diagonal with elements α_i^2. Now keep prior spherical

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \mathbf{I}\right)$$

- Covariance functions of this type are known as ARD (see e.g. Neal, 1996; MacKay, 2003; Rasmussen and Williams, 2006).

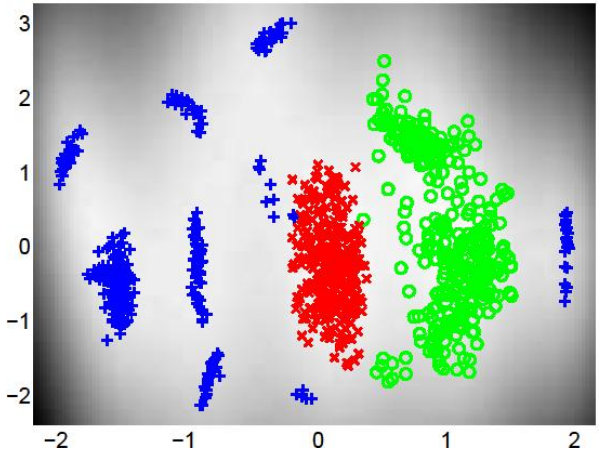
Automatic dimensionality detection

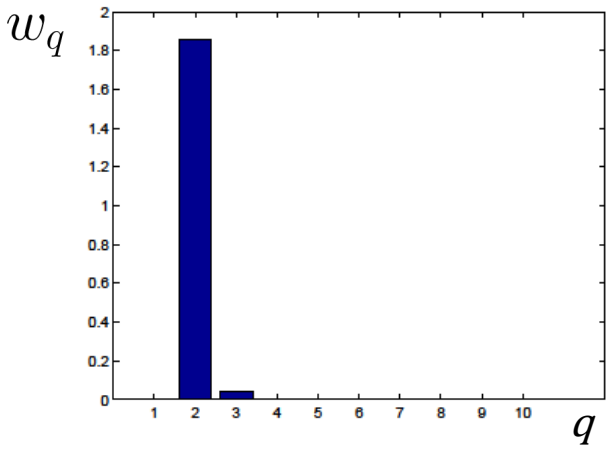
- Achieved by employing an *Automatic Relevance Determination (ARD)* covariance function for the prior on the GP mapping

- $f \sim GP(\mathbf{0}, k_f)$ with

$$k_f(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{i,q} - x_{j,q})^2\right)$$

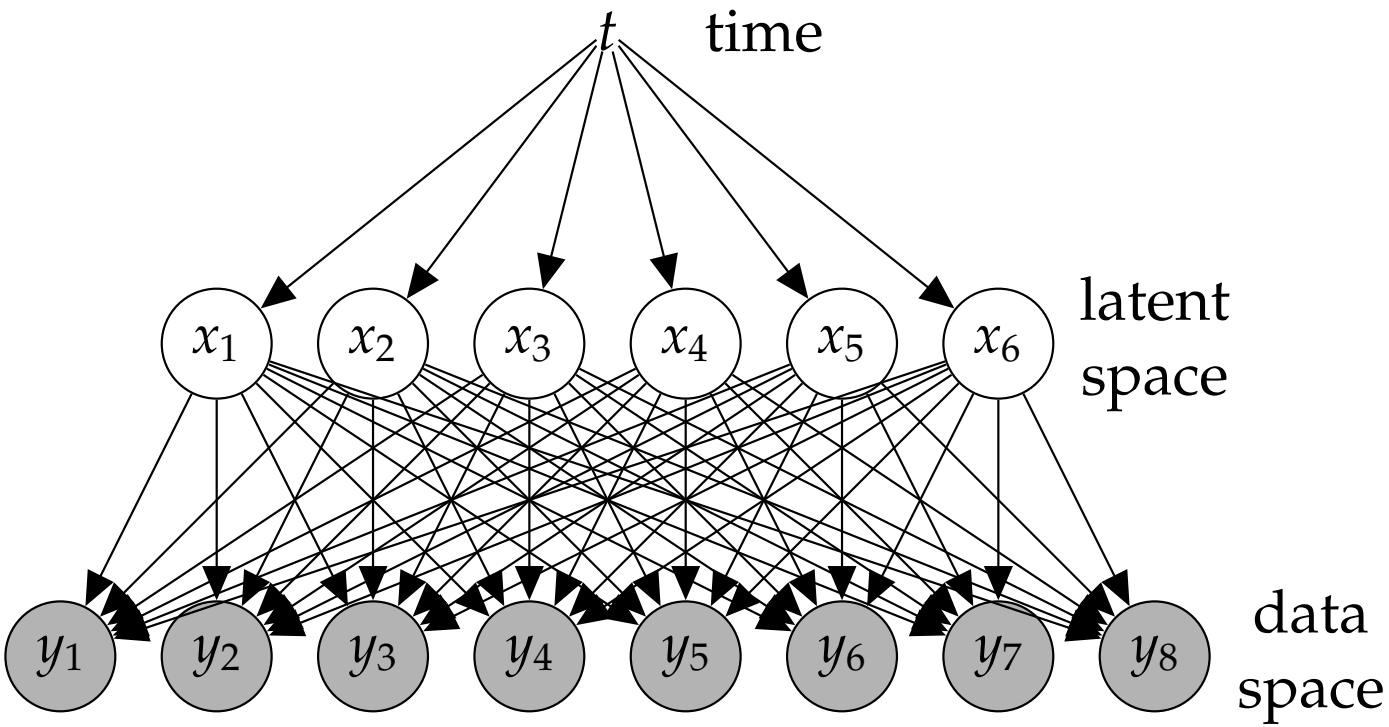
- Example





Gaussian Process Dynamical Systems

(Damianou et al., 2011)



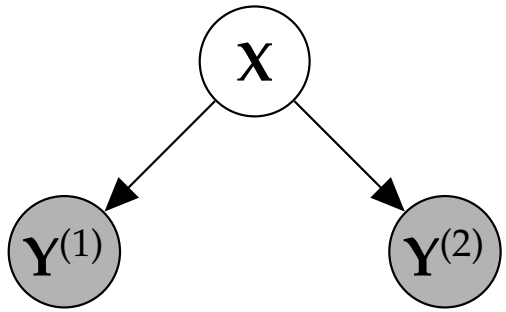
Gaussian Process over Latent Space

- Assume a GP prior for $p(\mathbf{X})$.
- Input to the process is time, $p(\mathbf{X}|t)$.

Interpolation of HD Video

Modeling Multiple 'Views'

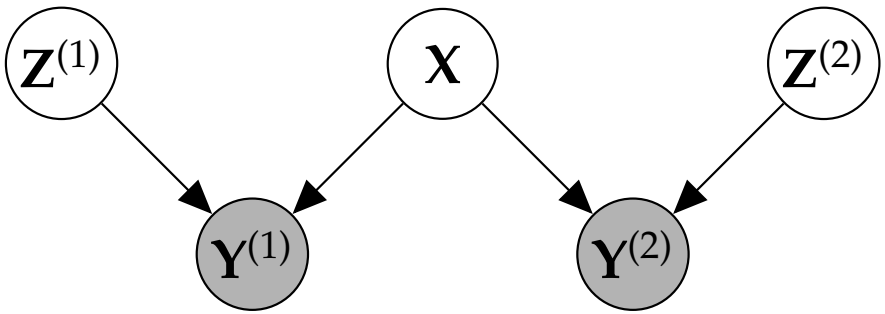
- Single space to model correlations between two different data sources, e.g., images & text, image & pose.
- Shared latent spaces: (Shon et al., 2006; Navaratnam et al., 2007; Ek et al., 2008b)



- Effective when the 'views' are correlated.
- But not all information is shared between both 'views'.
- PCA applied to concatenated data vs CCA applied to data.

Shared-Private Factorization

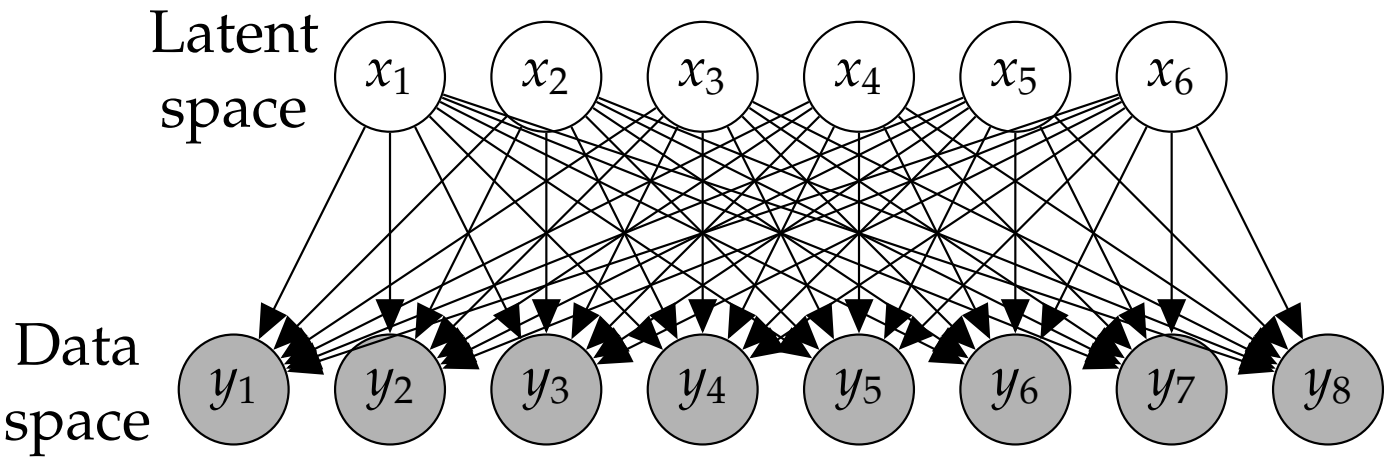
- In real scenarios, the 'views' are neither fully independent, nor fully correlated.
- Shared models
  - either allow information relevant to a single view to be mixed in the shared signal,
  - or are unable to model such private information.
- Solution: Model shared and private information (Virtanen et al., 2011; Ek et al., 2008a; Leen and Fyfe, 2006; Klami and Kaski, 2007, 2008; Tucker, 1958)



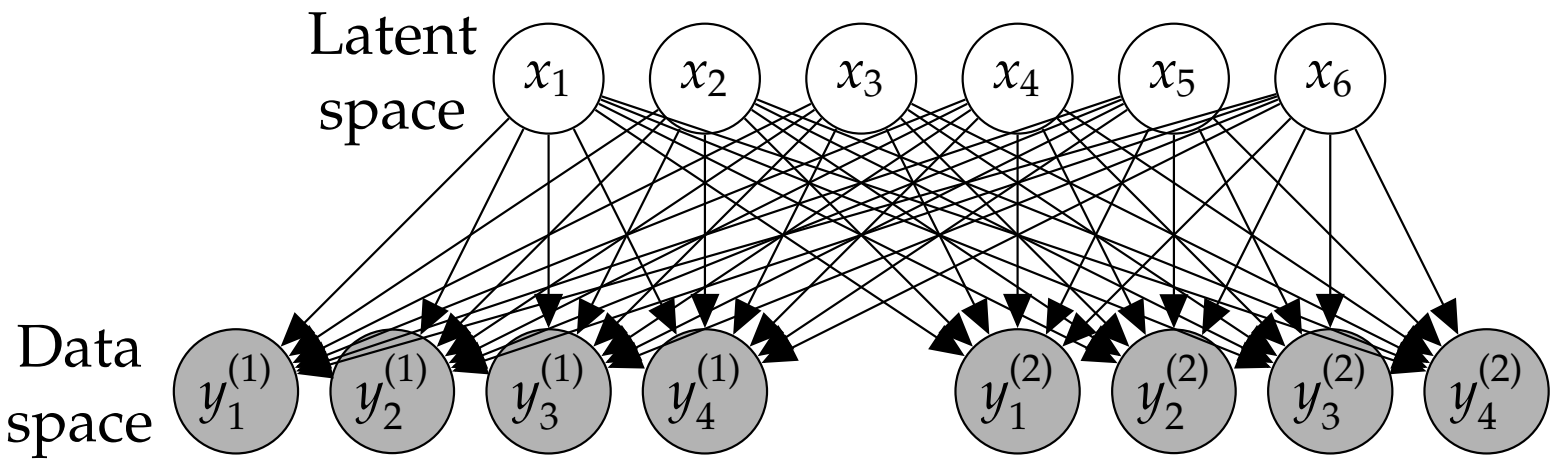
- Probabilistic CCA is case when dimensionality of $\mathbf{Z}$ matches $\mathbf{Y}^{(i)}$ (cf Inter Battery Factor Analysis (Tucker, 1958)).

Manifold Relevance Determination

Damianou et al. (2012)

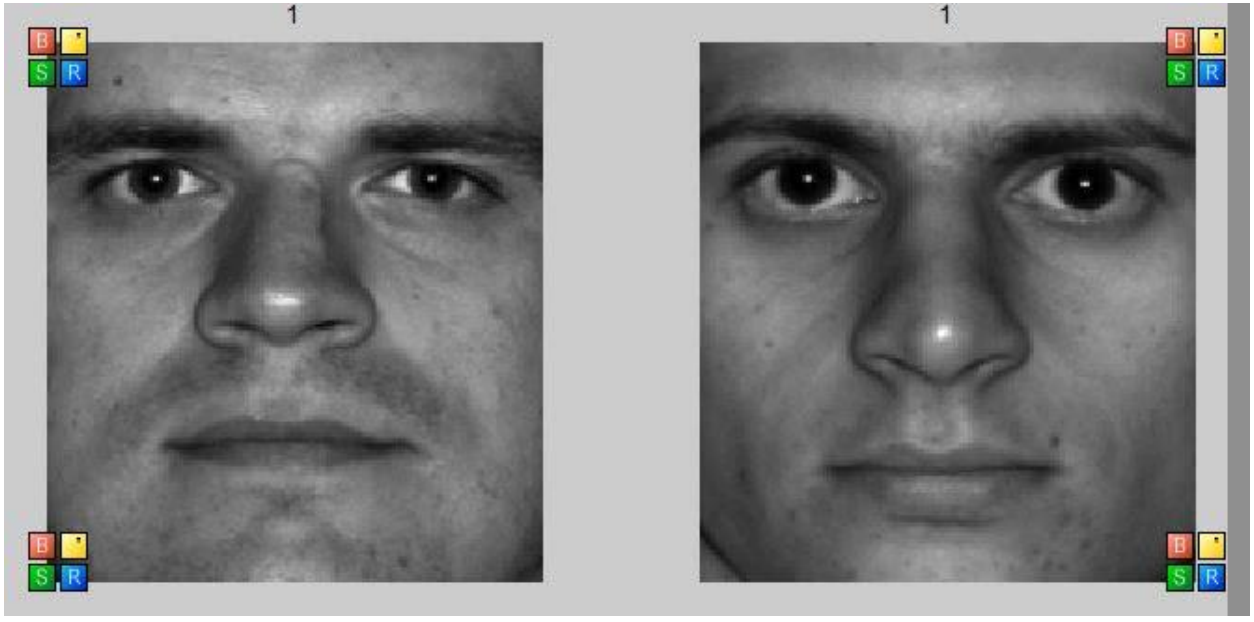


Shared GP-LVM



Separate ARD parameters for mappings to $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$.

Example: Yale faces



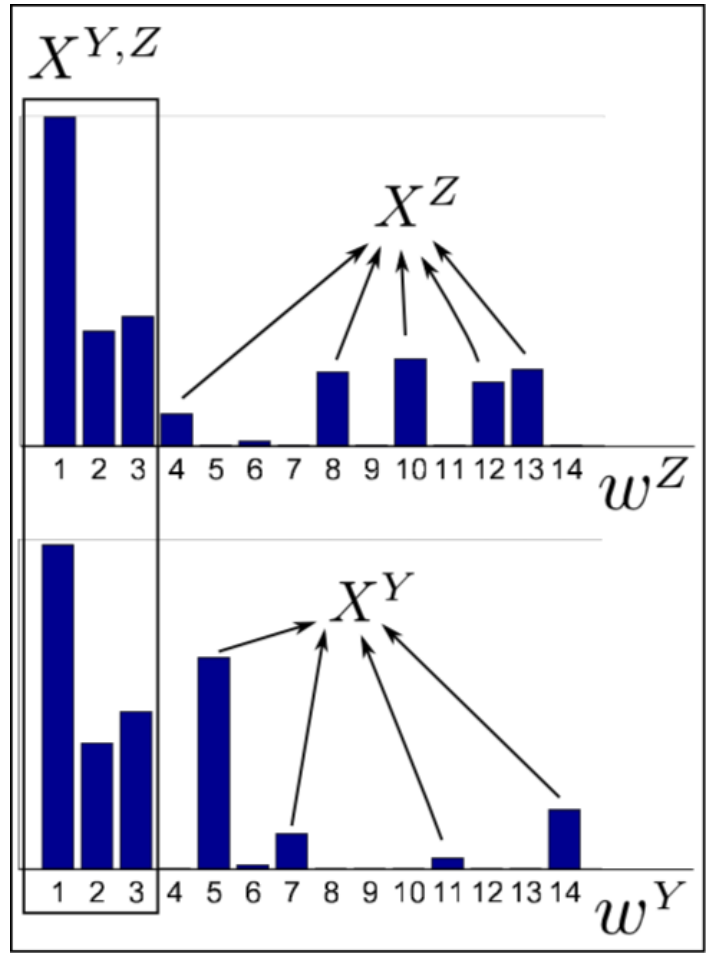
- Dataset Y: 3 persons under all illumination conditions
- Dataset Z: As above for 3 different persons
- Align datapoints $\mathbf{x}_n$ and $\mathbf{z}_n$ only based on the lighting direction

Results

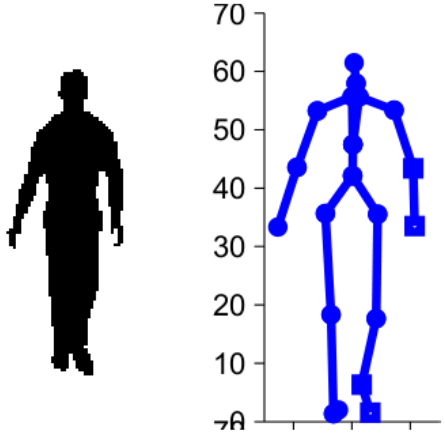
- Latent space X initialised with 14 dimensions

- Weights define a segmentation of X

- Video / demo…



Potential applications..?



Manifold Relevance Determination

References I

W. V. Baxter and K.-I. Anjyo. Latent doodle space. In *EUROGRAPHICS*, volume 25, pages 477–485, Vienna, Austria, September 4-8 2006.

C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: the Generative Topographic Mapping. *Neural Computation*, 10(1):215–234, 1998. [DOI].

E. V. Bonilla, K. M. Chai, and C. K. I. Williams. Multi-task Gaussian process prediction. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, Cambridge, MA, 2008. MIT Press.

S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140 (3):640–651, 2009. [DOI].

A. Damianou, C. H. Ek, M. K. Titsias, and N. D. Lawrence. Manifold relevance determination. In J. Langford and J. Pineau, editors, *Proceedings of the International Conference in Machine Learning*, volume 29, San Francisco, CA, 2012. Morgan Kauffman. [PDF].

A. Damianou, M. K. Titsias, and N. D. Lawrence. Variational Gaussian process dynamical systems. In P. Bartlett, F. Peirrera, C. Williams, and J. Lafferty, editors, *Advances in Neural Information Processing Systems*, volume 24, Cambridge, MA, 2011. MIT Press. [PDF].

References II

G. Della Gatta, M. Bansal, A. Ambesi-Impiombato, D. Antonini, C. Missero, and D. di Bernardo. Direct targets of the trp63 transcription factor revealed by a combination of gene expression profiling and reverse engineering. *Genome Research*, 18(6):939–948, Jun 2008. [URL]. [DOI].

C. H. Ek, J. Rihan, P. Torr, G. Rogez, and N. D. Lawrence. Ambiguity modeling in latent spaces. In A. Popescu-Belis and R. Stiefelhagen, editors, *Machine Learning for Multimodal Interaction (MLMI 2008)*, LNCS, pages 62–73. Springer-Verlag, 28–30 June 2008a. [PDF].

C. H. Ek, P. H. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In A. Popescu-Belis, S. Renals, and H. Bourlard, editors, *Machine Learning for Multimodal Interaction (MLMI 2007)*, volume 4892 of *LNCS*, pages 132–143, Brno, Czech Republic, 2008b. Springer-Verlag. [PDF].

Z. Ghahramani, editor. *Proceedings of the International Conference in Machine Learning*, volume 24, 2007. Omnipress. [Google Books] .

P. Goovaerts. *Geostatistics For Natural Resources Evaluation*. Oxford University Press, 1997. [Google Books] .

K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. In *ACM Transactions on Graphics (SIGGRAPH 2004)*, pages 522–531, 2004.

References III

J. D. Helterbrand and N. A. C. Cressie. Universal cokriging under intrinsic coregionalization. *Mathematical Geology*, 26(2):205–226, 1994.

D. M. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.

A. G. Journel and C. J. Huijbregts. *Mining Geostatistics*. Academic Press, London, 1978. [Google Books] .

A. A. Kalaitzis and N. D. Lawrence. A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression. *BMC Bioinformatics*, 12(180), 2011. [DOI].

A. Klami and S. Kaski. Local dependent components analysis. In Ghahramani (2007). [Google Books] .

A. Klami and S. Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72:39–46, 2008.

J. B. Kruskal. Multidimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–28, 1964. [DOI].

References IV

P. S. Laplace. *Essai philosophique sur les probabilités*. Courcier, Paris, 2nd edition, 1814. Sixth edition of 1840 translated and repreinted (1951) as *A Philosophical Essay on Probabilities*, New York: Dover; fifth edition of 1825 reprinted 1986 with notes by Bernard Bru, Paris: Christian Bourgois Éditeur, translated by Andrew Dale (1995) as *Philosophical Essay on Probabilities*, New York:Springer-Verlag.

N. D. Lawrence. Gaussian process models for visualisation of high dimensional data. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 329–336, Cambridge, MA, 2004. MIT Press.

N. D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 11 2005.

N. D. Lawrence and A. J. Moore. Hierarchical Gaussian process latent variable models. In Ghahramani (2007), pages 481–488. [Google Books] . [PDF].

N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In R. Greiner and D. Schuurmans, editors, *Proceedings of the International Conference in Machine Learning*, volume 21, pages 512–519. Omnipress, 2004. [PDF].

References V

N. D. Lawrence and J. Quiñonero Candela. Local distance preservation in the GP-LVM through back constraints. In W. Cohen and A. Moore, editors, *Proceedings of the International Conference in Machine Learning*, volume 23, pages 513–520. Omnipress, 2006. [Google Books] . [PDF].

G. Leen and C. Fyfe. A Gaussian process latent variable model formulation of canonical correlation analysis. Bruges (Belgium), 26-28 April 2006 2006.

S. Levine, J. M. Wang, A. Haraux, Z. Popović, and V. Koltun. Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics (SIGGRAPH 2012)*, 31(4), 2012.

D. Lowe and M. E. Tipping. Neuroscale: Novel topographic feature extraction with radial basis function networks. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 543–549, Cambridge, MA, 1997. MIT Press.

C. Lu and X. Tang. Surpassing human-level face verification performance on LFW with GaussianFace. Technical report,

D. J. C. MacKay. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research, A*, 354(1):73–80, 1995. [DOI].

D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, U.K., 2003. [Google Books] .

References VI

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic Press, London, 1979. [Google Books] .

T. P. Minka and R. W. Picard. Learning how to learn is learning with point sets. Available on-line., 1997. [URL]. Revised 1999, available at `http://www.stat.cmu.edu/~{}minka/`.

R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society Press, 2007.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996. Lecture Notes in Statistics 118.

J. Oakley and A. O'Hagan. Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, 89(4):769–784, 2002.

V. Priacuriu and I. D. Reid. Nonlinear shape manifolds as shape priors in level set segmentation and tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011a.

V. Priacuriu and I. D. Reid. Shared shape spaces. In *IEEE International Conference on Computer Vision (ICCV)*, 2011b.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. [Google Books] .

References VII

S. T. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 626–632, Cambridge, MA, 1998. MIT Press.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. [DOI].

J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969. [DOI].

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. [DOI].

M. Seeger and M. I. Jordan. Sparse Gaussian Process Classification With Multiple Classes. Technical Report 661, Department of Statistics, University of California at Berkeley,

A. P. Shon, K. Grochow, A. Hertzmann, and R. P. N. Rao. Learning shared latent structure for image synthesis and robotic imitation. In Weiss et al. (2006).

G. Skolidis and G. Sanguinetti. Bayesian multitask classification with Gaussian process priors. *IEEE Transactions on Neural Networks*, 22(12):2011 – 2021, 2011.

M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, 1999. [Google Books] .

References VIII

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 333–340, Barbados, 6-8 January 2005. Society for Artificial Intelligence and Statistics.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319–2323, 2000. [DOI].

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999. [PDF]. [DOI].

M. K. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In Y. W. Teh and D. M. Titterington, editors, *Proceedings of the Thirteenth International Workshop on Artificial Intelligence and Statistics*, volume 9, pages 844–851, Chia Laguna Resort, Sardinia, Italy, 13-16 May 2010. JMLR W&CP 9. [PDF].

L. R. Tucker. An inter-battery method of factor analysis. *Psychometrika*, 23(2): 111–136, 1958.

R. Urtasun and T. Darrell. Discriminative Gaussian process latent variable model for classification. In Ghahramani (2007). [Google Books] .

References IX

R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 238–245, New York, U.S.A., 17–22 Jun. 2006. IEEE Computer Society Press.

R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *IEEE International Conference on Computer Vision (ICCV)*, pages 403–410, Bejing, China, 17–21 Oct. 2005. IEEE Computer Society Press.

S. Virtanen, A. Klami, and S. Kaski. Bayesian CCA via group sparsity. In L. Getoor and T. Scheffer, editors, *Proceedings of the International Conference in Machine Learning*, volume 28, 2011.

H. Wackernagel. *Multivariate Geostatistics: An Introduction With Applications*. Springer-Verlag, 3rd edition, 2003. [Google Books] .

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In Weiss et al. (2006).

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008. ISSN 0162-8828. [DOI].

Y. Weiss, B. Schölkopf, and J. C. Platt, editors. *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

References X

C. K. Williams and D. Barber. Bayesian Classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (12):1342–1351, 1998.

C. K. I. Williams. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.

I. Woodward, M. R. Lomas, and R. A. Betts. Vegetation-climate feedbacks in a greenhouse world. *Philosophical Transactions: Biological Sciences*, 353(1365): 29–39, 1998.

K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 1012–1019, 2005.