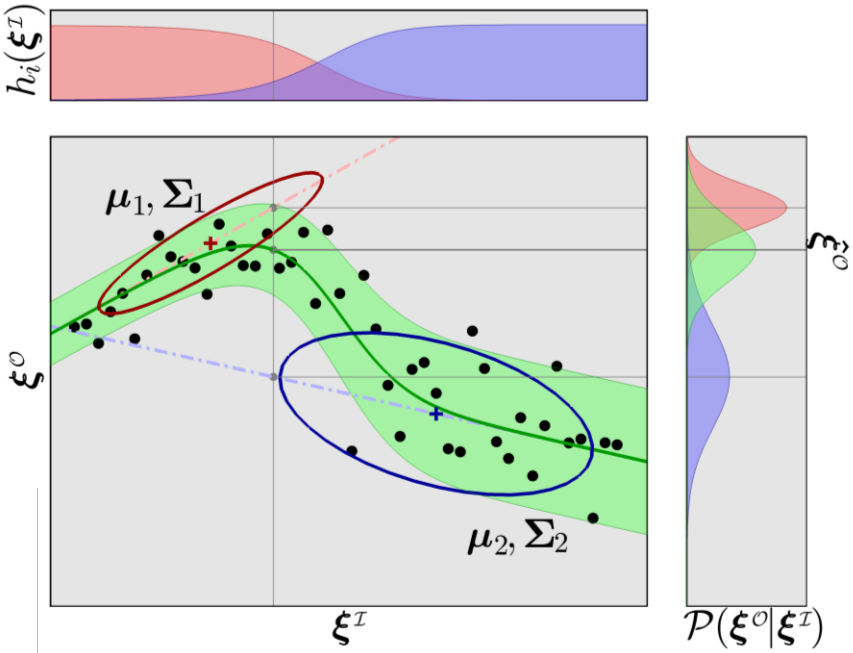
EE613
Machine Learning for Engineers

NONLINEAR REGRESSION II

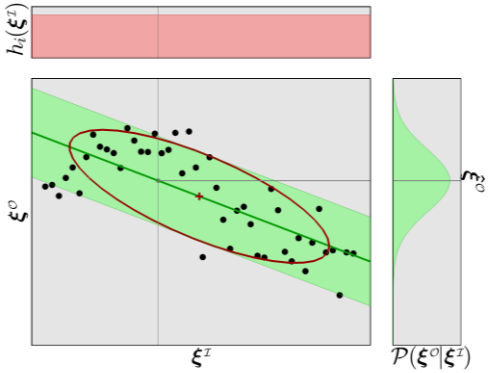
Sylvain Calinon
Robot Learning & Interaction Group
Idiap Research Institute
Dec. 20, 2017

First, let's recap some useful properties and approaches presented in previous lectures…

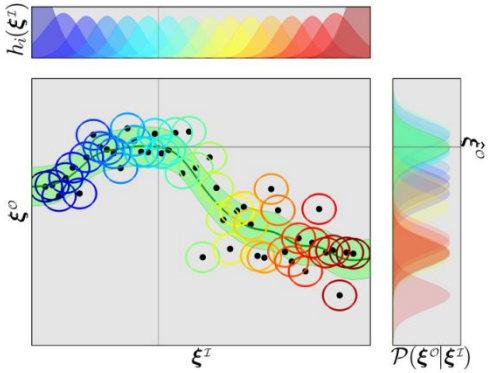
Gaussian mixture regression (GMR)



Least squares linear regression



Nadaraya-Watson kernel regression



GMR can cover a large spectrum of regression mechanisms

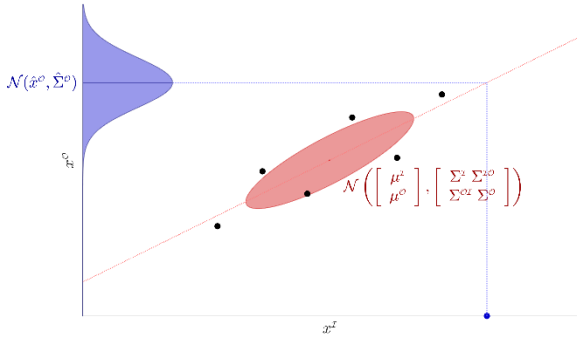
Both $\xi^{\mathcal{I}}$ and $\xi^{\mathcal{O}}$ can be multidimensional

$\mathcal{P}(\xi^{\mathcal{I}}, \xi^{\mathcal{O}})$ encoded in **Gaussian mixture model (GMM)**

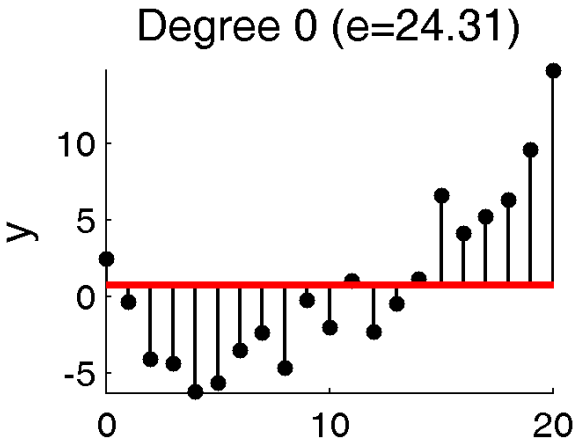
$\mathcal{P}(\xi^{\mathcal{O}}|\xi^{\mathcal{I}})$ retrieved by **Gaussian mixture regression (GMR)**

Conditioning and regression

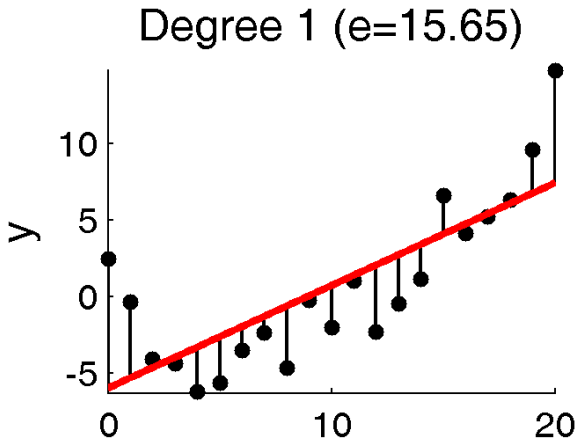
If $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we have that $\boldsymbol{x}^{\mathcal{O}}|\boldsymbol{x}^{\mathcal{I}} \sim \mathcal{N}\left(\hat{\boldsymbol{x}}^{\mathcal{O}}, \hat{\boldsymbol{\Sigma}}^{\mathcal{O}}\right)$, with parameters



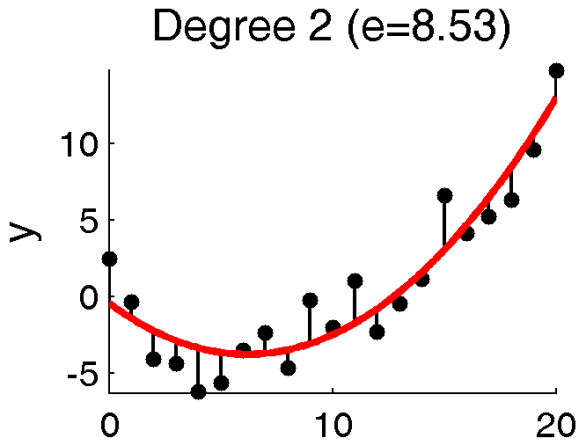
$$\hat{\boldsymbol{x}}^{\mathcal{O}} = \boldsymbol{\mu}^{\mathcal{O}} + \boldsymbol{\Sigma}^{\mathcal{OI}}\boldsymbol{\Sigma}^{\mathcal{I}-1}(\boldsymbol{x}^{\mathcal{I}} - \boldsymbol{\mu}^{\mathcal{I}}),$$
$$\hat{\boldsymbol{\Sigma}}^{\mathcal{O}} = \boldsymbol{\Sigma}^{\mathcal{O}} - \boldsymbol{\Sigma}^{\mathcal{OI}}\boldsymbol{\Sigma}^{\mathcal{I}-1}\boldsymbol{\Sigma}^{\mathcal{IO}}.$$



$\boldsymbol{X} = 1$



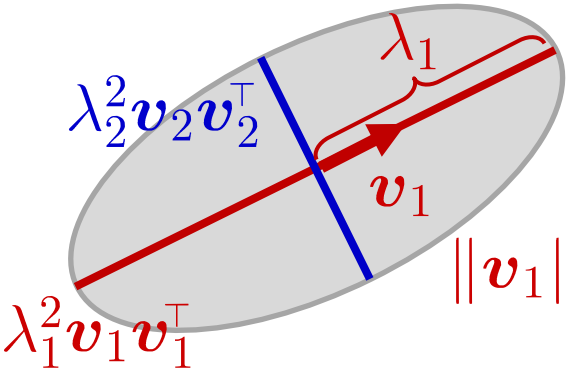
$\boldsymbol{X} = [\boldsymbol{x}, 1]$



$\boldsymbol{X} = [\boldsymbol{x}^2, \boldsymbol{x}, 1]$

Stochastic sampling with Gaussians

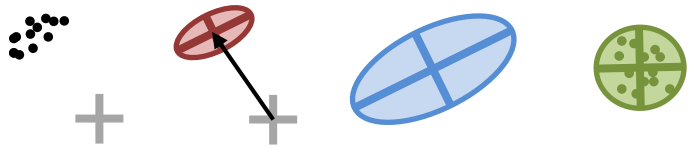
The eigendecomposition of $\boldsymbol{\Sigma}$ is expressed in a matrix form as



$$\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^\top = \sum_{j=1}^{D} \lambda_j^2 \boldsymbol{v}_j \boldsymbol{v}_j^\top$$

with $\boldsymbol{V} = \left[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D\right]$ and $\boldsymbol{D} = \begin{bmatrix} \lambda_1^2 & 0 & \cdots & 0 \\ 0 & \lambda_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_D^2 \end{bmatrix}$

By using this notation, datapoints can be stochastically generated with

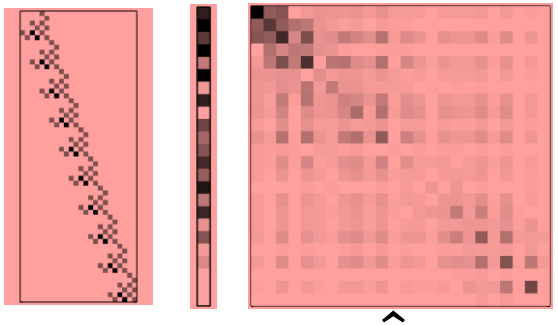


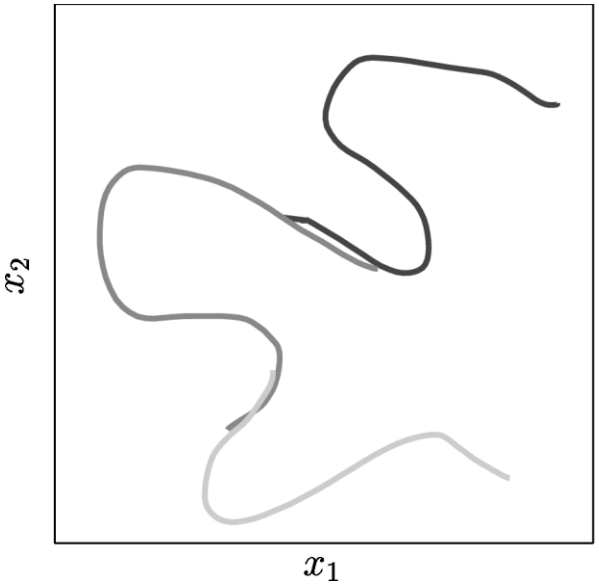
$$\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \Longleftrightarrow \quad \boldsymbol{\xi} \sim \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{D}^{\frac{1}{2}}\,\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

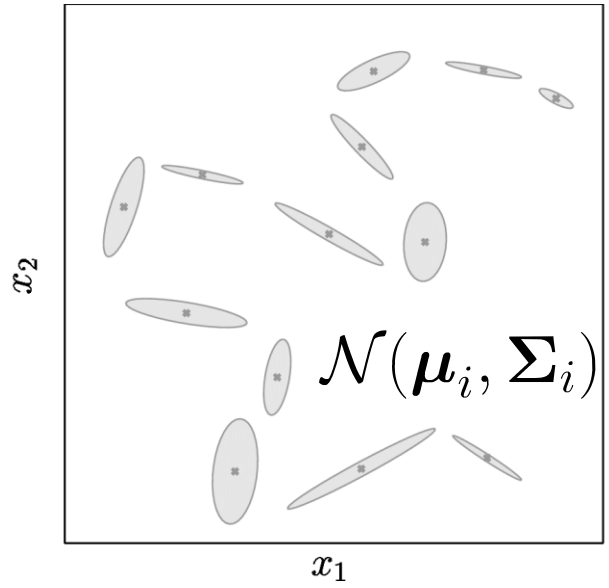
GMM/HMM with dynamic features

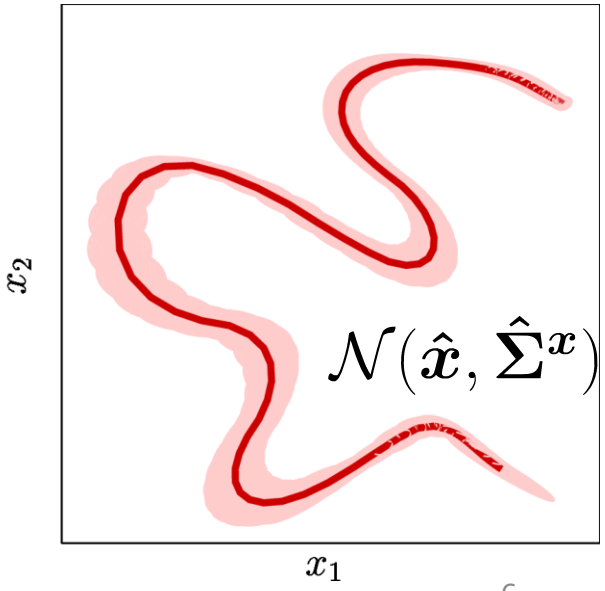
$$\mathcal{P}(\boldsymbol{\zeta}|\boldsymbol{s}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{\zeta}_t \,|\, \boldsymbol{\mu}_{s_t}, \boldsymbol{\Sigma}_{s_t}) \quad \rightarrow \quad \mathcal{P}(\boldsymbol{\Phi}\boldsymbol{x}|\boldsymbol{s}) = \mathcal{N}(\boldsymbol{\Phi}\boldsymbol{x} \,|\, \boldsymbol{\mu}_{\boldsymbol{s}}, \boldsymbol{\Sigma}_{\boldsymbol{s}})$$

with $\boldsymbol{\mu}_{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{\mu}_{s_1} \\ \boldsymbol{\mu}_{s_2} \\ \vdots \\ \boldsymbol{\mu}_{s_T} \end{bmatrix}$ and $\boldsymbol{\Sigma}_{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{\Sigma}_{s_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{s_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\Sigma}_{s_T} \end{bmatrix}$









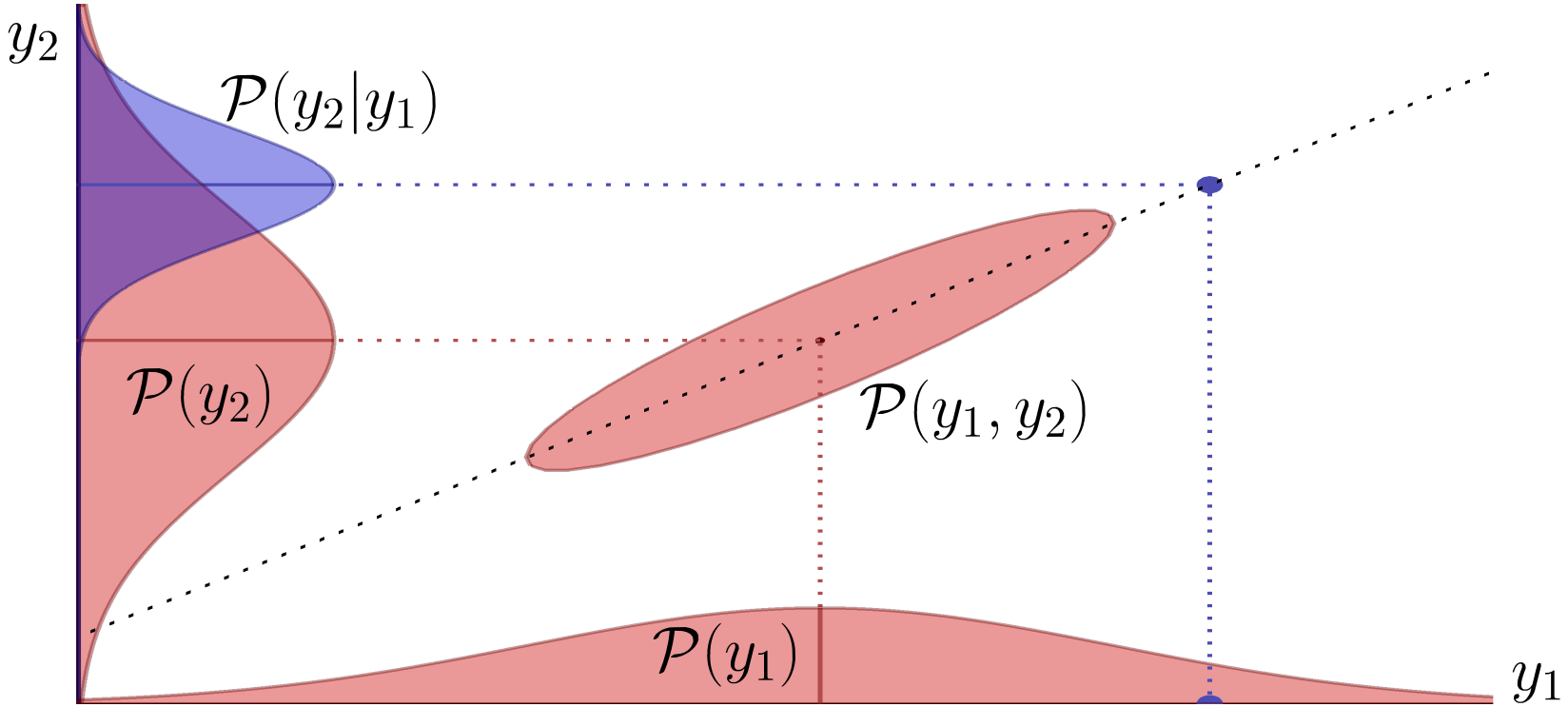
Gaussian process (GP)

[C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In Advances in Neural Information Processing Systems (NIPS), pages 514–520, 1996]

[S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. Philosophical Trans. of the Royal Society A, 371(1984):1–25, 2012]

Gaussian process - Informal interpretation

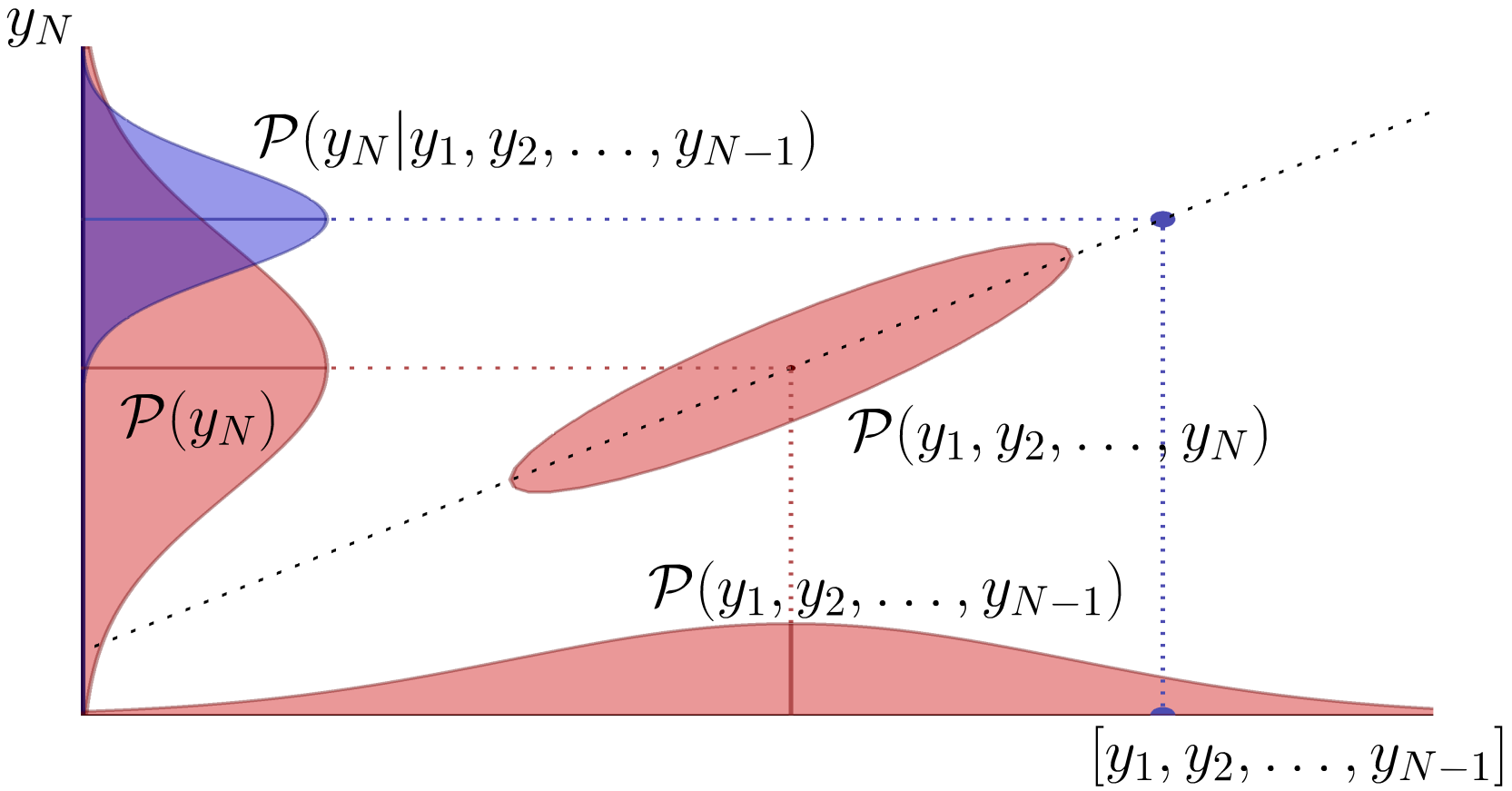
- A joint distribution represented by a bivariate Gaussian forms marginal distributions P(y_1) and P(y_2) that are unidimensional.
- Observing y_1 changes our belief about y_2, giving rise to a **conditional distribution**.
- Knowledge of the covariance lets us shrink uncertainty in one variable based on the observation of the other.



Gaussian process - Informal interpretation

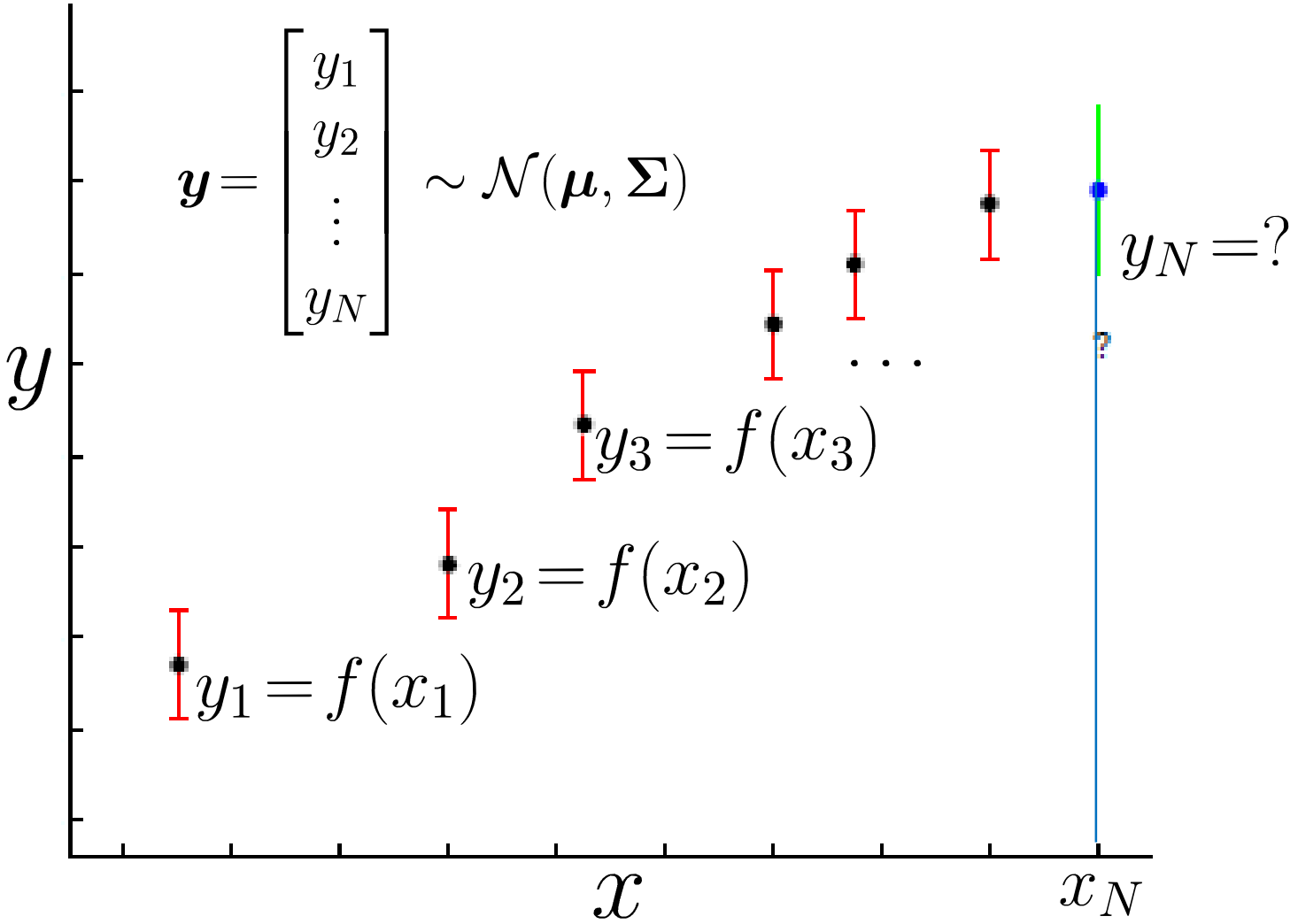
- This bivariate example can be extended to an arbitrarily large number of variables.
- Indeed, observations in an arbitrary dataset can always be imagined as a single point sampled from a multivariate Gaussian distribution.

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

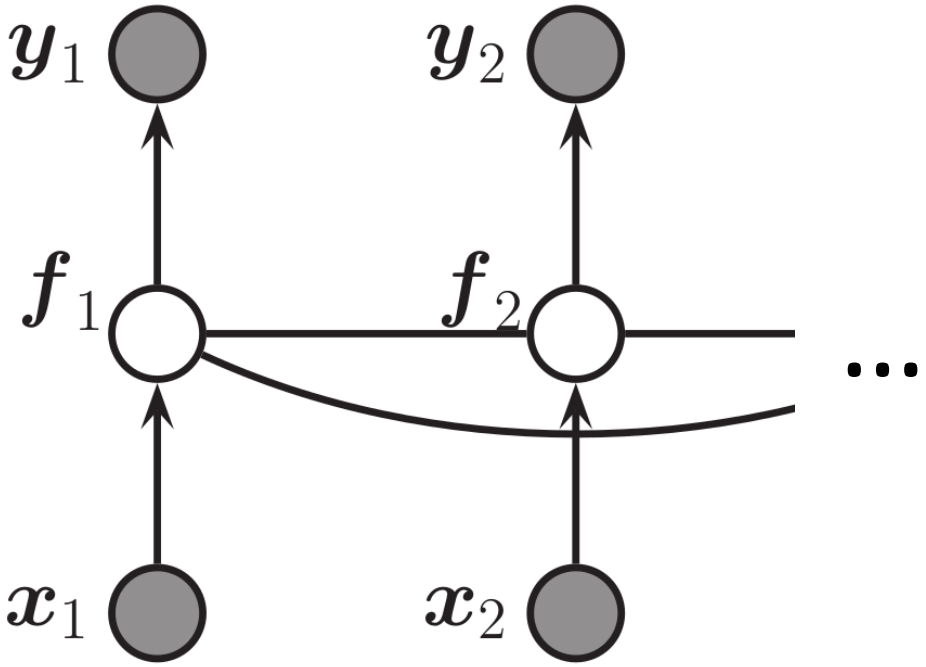


How to construct this joint distribution in GP?

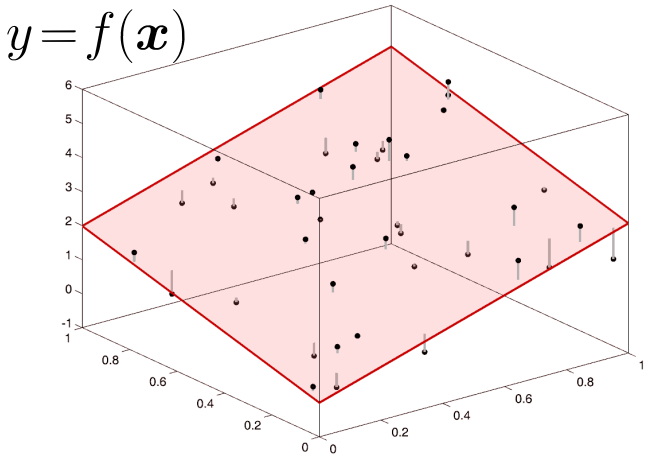
By looking at the similarities in the continuous **x** space, representing the locations at which we e aluate y = f(**x**)



Graphical model of a Gaussian process



$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$



x can be multivariate

Note that with GPs, we do not build joint distributions on {x_1,x_2,…x_N}!

Gaussian process (GP)

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

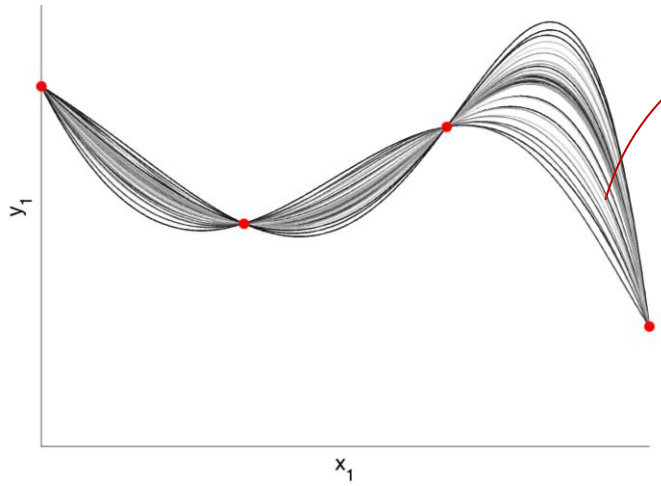
- Gaussian processes (GPs) can be seen as an infinite-dimensional generalization of multivariate normal distributions.
- The **infinite joint distribution** over all possible variables is equivalent to a **distribution over a function space** y = f(**x**).
- **x** can for be a vector or any object, but y is a scalar output.
- Although it might seem difficult to represent a distribution over a function, it turns out that we only need to be able to define a distribution over the function values at a finite, but arbitrary, set of points.
- To understand GPs, N observations of an arbitrary data set $y = \{y_1, ..., y_N\}$ should be imagined as a single point sampled from an N-variate Gaussian.

Gaussian process (GP)

- Gaussian processes are useful in statistical modelling, benefiting from properties inherited from multivariate normal distributions.
- When a random process is modelled as a Gaussian process, the distributions of various derived quantities can be obtained explicitly. Such quantities include the average value of the process over a range of times, and the error in estimating the average using sample values at a small set of times.

→ Usually more powerful than just selecting a model type, such as selecting the degree of a polynomial to fit a dataset, as we have seen in the lecture about linear regression.

Gaussian process - Informal interpretation



Polynomial fitting with least squares and nullspace optimization

$$\hat{\boldsymbol{A}} = \boldsymbol{X}^{\dagger}\boldsymbol{Y} + \overbrace{(\boldsymbol{I} - \boldsymbol{X}^{\dagger}\boldsymbol{X})}^{N}\boldsymbol{V}$$

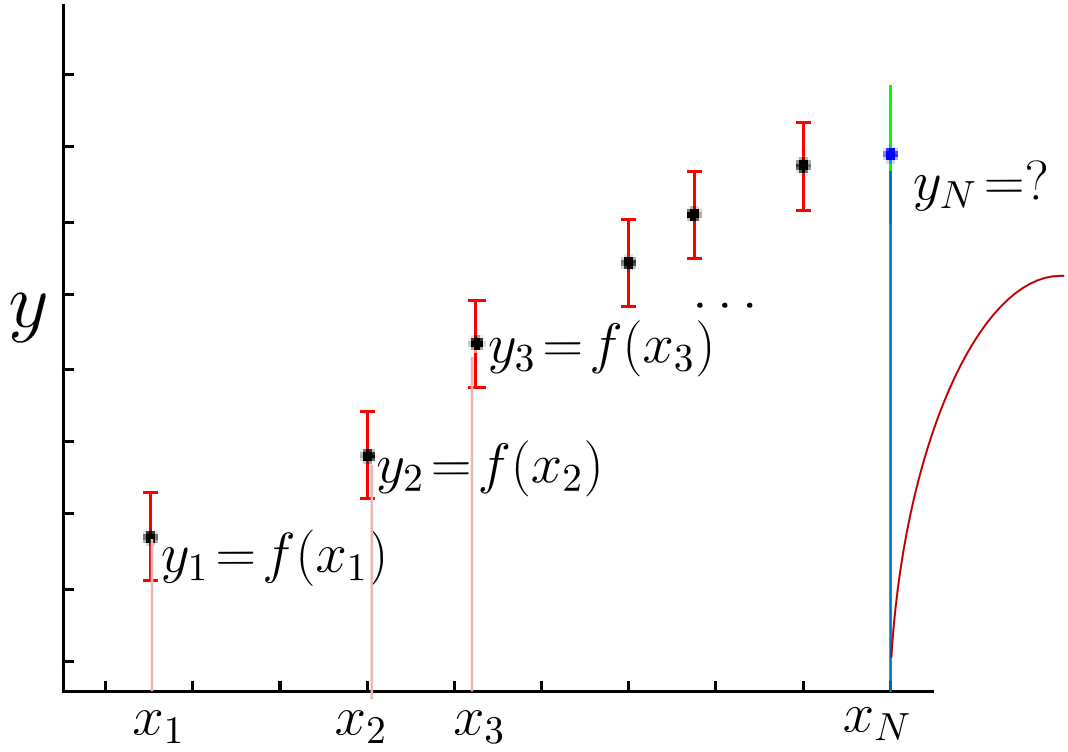
$$\text{with} \quad \boldsymbol{X} = [\boldsymbol{x}^6, \boldsymbol{x}^5, \ldots, \mathbf{1}]$$

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

- In the lecture about linear regression, we have seen polynomial fitting as an example of parametric modeling technique, where we provided the degree of the polynomial.
- We have seen that the nullspace projection operator could be used to generate multiple solutions of a fitting problem, thus obtaining a family of curves differing in regions where we have no observation.
- Now, we will treat this property as a distribution of curves, each offering a valid explanation for the observed data. Bayesian modeling will be the central tool to working with such **distribution over curves**.

Gaussian process (GP)

- The covariance lies at the core of Gaussian process, where a covariance over an arbitrarily large set of variables can be defined through the **covariance kernel function** $k(\mathbf{x}_i, \mathbf{x}_j)$, providing the covariance elements between any two sample locations $\mathbf{x}_i$ and $\mathbf{x}_j$.



If x_N is closer to x_3 than x_1, we also expect y_N to be closer to y_3 than y_1

Distribution over functions in GPs

For a set of spatial or temporal locations $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, the covariance matrix (also known as the Gram matrix) is defined as

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}) = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & k(\boldsymbol{x}_N, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}.$$

The entire function evaluation $y_n = f(\boldsymbol{x}_n)$ associated with the set of inputs $\boldsymbol{x}_n$ is a draw from a multivariate Gaussian distribution

$$\boldsymbol{y} \sim \mathcal{N}\Big(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})\Big),$$

specifying a distribution over functions.

How to choose $k(x_i,x_j)$?

- We may know that our observations are samples from an underlying process that is smooth, that is continuous, that has typical amplitude, or that the variations in the function take place over known time scales (e.g., within a typical dynamic range), etc.

  → We will work mathematically with the **infinite space of all functions** that have these characteristics.

- The underlying models still require hyperparameters to be inferred, but **these hyperparameters govern characteristics that are more generic such as the scale of a distribution** rather than acting explicitly on the structure or functional form of the signals.

How to choose k(xi,xj)?

$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{x}) = \begin{bmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & k(\boldsymbol{x}_1,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_1,\boldsymbol{x}_N) \\ k(\boldsymbol{x}_2,\boldsymbol{x}_1) & k(\boldsymbol{x}_2,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_2,\boldsymbol{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N,\boldsymbol{x}_1) & k(\boldsymbol{x}_N,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_N,\boldsymbol{x}_N) \end{bmatrix}$$

- The kernel function is chosen to express a property of similarity so that for points **x**i and **x**j that are similar, we expect the corresponding output of the functions **y**i and **y**j to be similar.
- The notion of similarity will depend on the application. Some of the basic aspects that can be defined through the covariance function *k* are the process **stationarity, isotropy, smoothness or periodicity**.
- When considering continuous time series, it can usually be assumed that past observations can be informative about current data as a function of how long ago they were observed.
- This corresponds to a **stationary** covariance, dependent on the Euclidean distance $|\mathbf{x}_i - \mathbf{x}_j|$.
- This process is also considered as **isotropic** if it does not depend on directions between **x**i and **x**j.
- A process that is both stationary and isotropic is **homogeneous**.

k(xi,xj) as squared exponential covariance

A very popular homogeneous covariance function is the squared exponential kernel, also known as **radial basis function (RBF)**

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\text{GP}} \exp\left(-\frac{1}{\Theta_2^{\text{GP}}}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)\right),$$

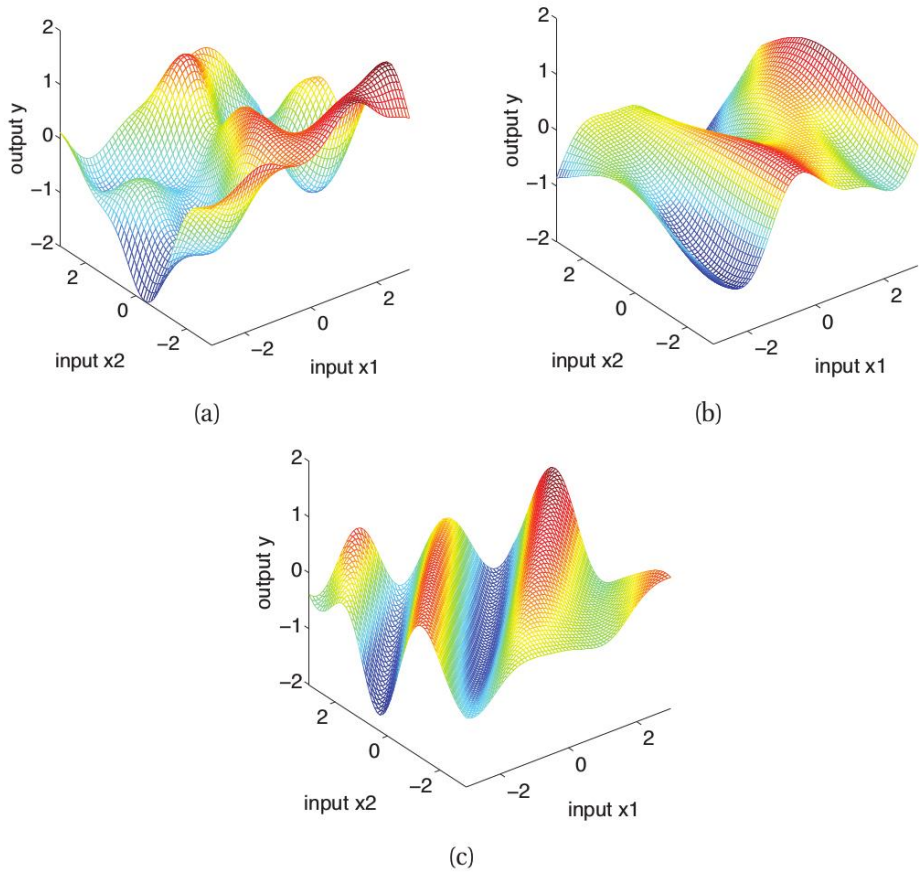
with two hyperparameters Θ_1^{GP} and Θ_2^{GP} corresponding respectively to **output and input scales** of the problem.

Θ_1^{GP} sets the maximum allowable covariance, which should then be high for functions covering a broad axis range.

The radial basis function is widely employed when it is expected that nearby inputs $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ will have their corresponding outputs $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ also nearby (**assumption of continuity**).

k(xi,xj) as squared exponential covariance

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\mathrm{GP}} \exp\left(-\frac{1}{\Theta_2^{\mathrm{GP}}}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{M}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right)$$



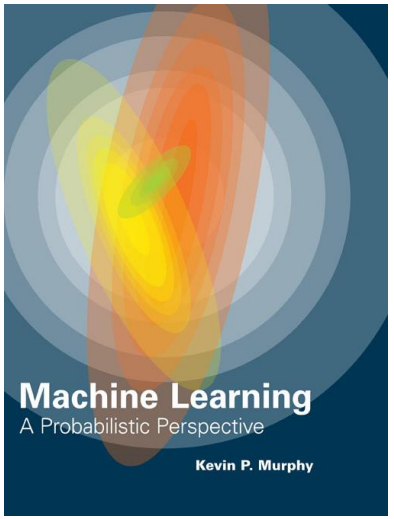


Figure 15.4 Some 2d functions sampled from a GP with an SE kernel but different hyper-parameters. The kernel has the form in Equation 15.20 where (a) $\mathbf{M} = \mathbf{I}$, (b) $\mathbf{M} = \mathrm{diag}(1, 3)^{-2}$, (c) $\mathbf{M} = (1, -1; -1, 1) + \mathrm{diag}(6, 6)^{-2}$. Based on Figure 5.1 of (Rasmussen and Williams 2006). Figure generated by `gprDemoArd`, written by Carl Rasmussen.

Modeling noise in the observed y_n

If we assume there is noise associated with the observed function values $\boldsymbol{y}_n = f(\boldsymbol{x}_n) + \boldsymbol{\eta}$, this noise term can also be modeled in the covariance.

This noise is most often assumed to be uncorrelated from sample to sample, meaning that the noise term is only added to the diagonal elements of $\boldsymbol{K}$, giving a modified covariance for noisy observations of the form

$$\tilde{\boldsymbol{K}}(\boldsymbol{x}, \boldsymbol{x}) = \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}) + \Theta_3^{\mathrm{GP}} \boldsymbol{I}$$

where $\boldsymbol{I}$ is the identity matrix and Θ_3^{GP} is a Gaussian process hyperparameter representing the noise variance.

Modeling noise in the observed y_n

In the example of a covariance defined as squared exponential function, if noisy observations $\boldsymbol{y}$ are assumed, the kernel can be directly defined as

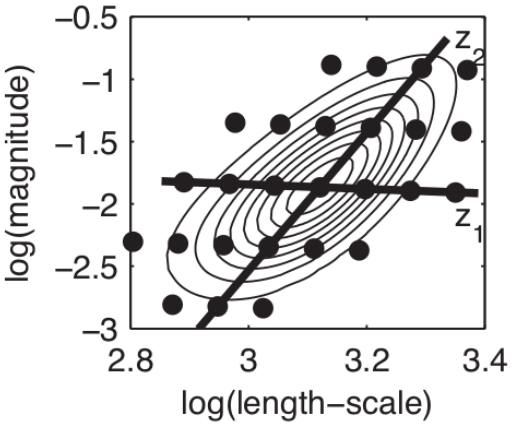
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\mathrm{GP}} \exp\left(-\frac{1}{\Theta_2^{\mathrm{GP}}}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)\right) + \Theta_3^{\mathrm{GP}}\delta_{i,j},$$

where $\delta_{i,j} = \mathbb{I}(i = j)$ is equal to one only when $i = j$ and is zero otherwise, resulting in a covariance matrix $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})$ with noise related to observations only present in the diagonal (noise uncorrelated from sample to sample).

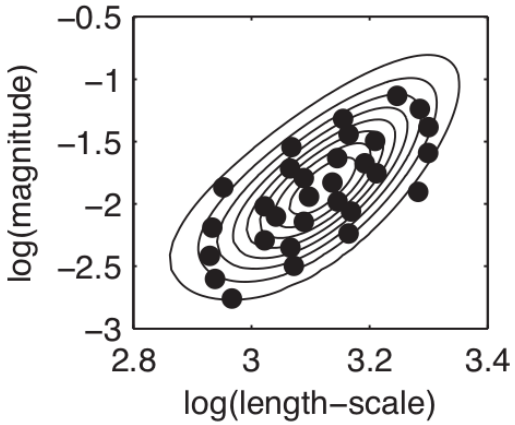
Learning the kernel function parameters

Several approaches exist to estimate the hyperparameters of the covariance function: Maximum Likelihood Estimation (MLE), cross-validation (CV), Bayesian approaches involving sampling algorithms such as MCMC, etc.

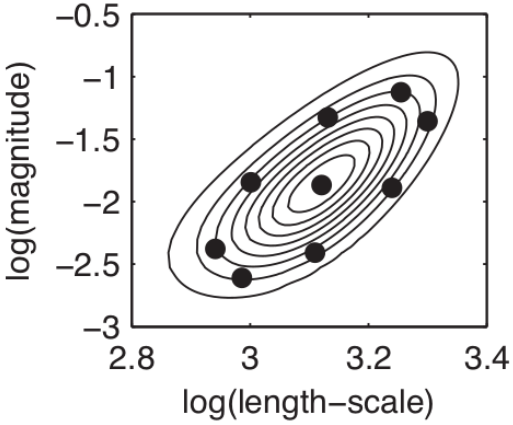
Grid-based



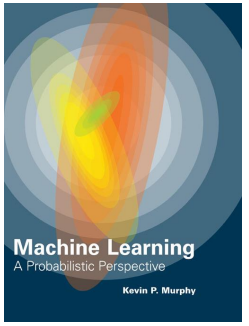
Monte-Carlo



2nd order model

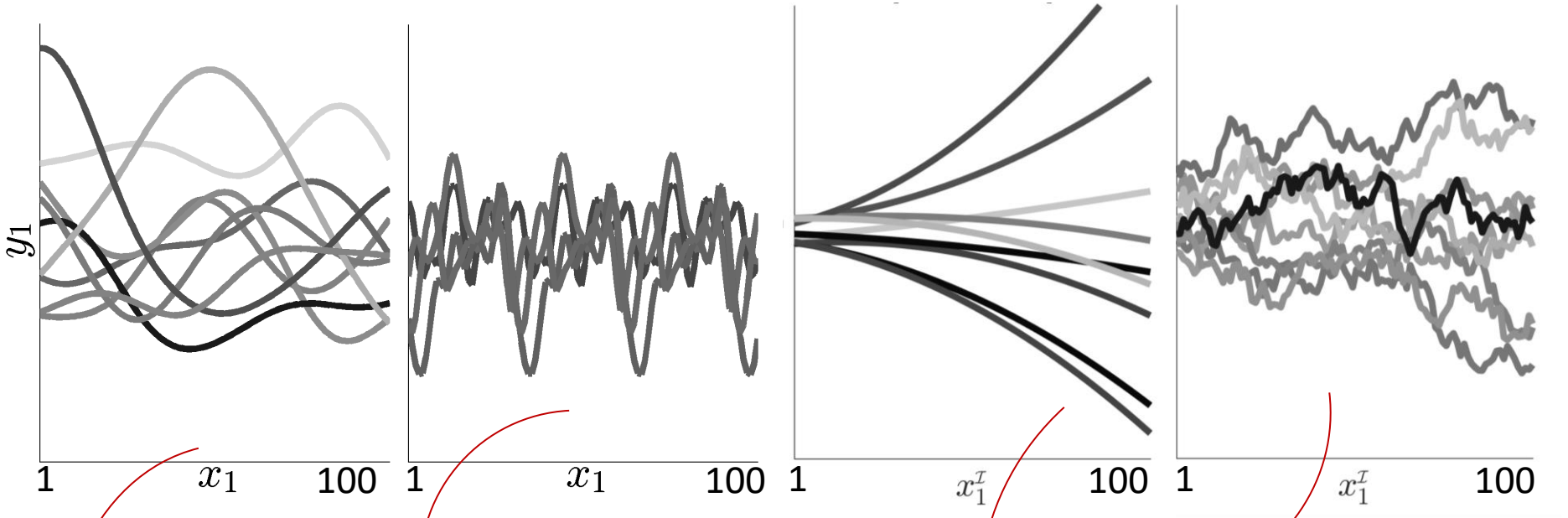


For example, given an expression for the log marginal likelihood and its derivative, we can estimate the kernel parameters using standard gradient-based optimizer. Note that since the objective is not convex, local minima can still be a problem.



Stochastic sampling from covariance matrix

$$\boldsymbol{y} \sim \mathcal{N}\big(\mathbf{0}, \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})\big) \qquad \boldsymbol{x} = [1, 2, \ldots, 100]^\top$$



$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\mathrm{GP}} \exp\left(-\frac{1}{\Theta_2^{\mathrm{GP}}} \sin^2\left(\Theta_4^{\mathrm{GP}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\right)\right)$$

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \min(\boldsymbol{x}_i, \boldsymbol{x}_j) + \Theta_1^{\mathrm{GP}}$$

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\mathrm{GP}} \exp\left(-\frac{1}{\Theta_2^{\mathrm{GP}}} (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\right)$$

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j + \Theta_1^{\mathrm{GP}})^2$$

Gaussian process regression (GPR)

a.k.a.

Kriging

Matlab code: demo_GPR01.m

[C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In Advances in Neural Information Processing Systems (NIPS), pages 514–520, 1996]

[S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. Philosophical Trans. of the Royal Society A, 371(1984):1–25, 2012]

Kriging, Gaussian process regression (GPR)

We are interested in the **posterior distribution** of $\boldsymbol{y}^*$ to be computed at some location(s) $\boldsymbol{x}^*$.

The **joint distribution** of the already observed $\boldsymbol{y}$ (at location $\boldsymbol{x}$) augmented by $\boldsymbol{y}^*$ (at location $\boldsymbol{x}^*$) is

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\boldsymbol{x}) \\ \boldsymbol{\mu}(\boldsymbol{x}^*) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{x},\boldsymbol{x}) & \boldsymbol{K}(\boldsymbol{x},\boldsymbol{x}^*) \\ \boldsymbol{K}(\boldsymbol{x}^*,\boldsymbol{x}) & \boldsymbol{K}(\boldsymbol{x}^*,\boldsymbol{x}^*) \end{bmatrix} \right)$$

We can use the conditional probability property of Gaussians to evaluate the posterior distribution over $\boldsymbol{y}^*$, yielding a Gaussian

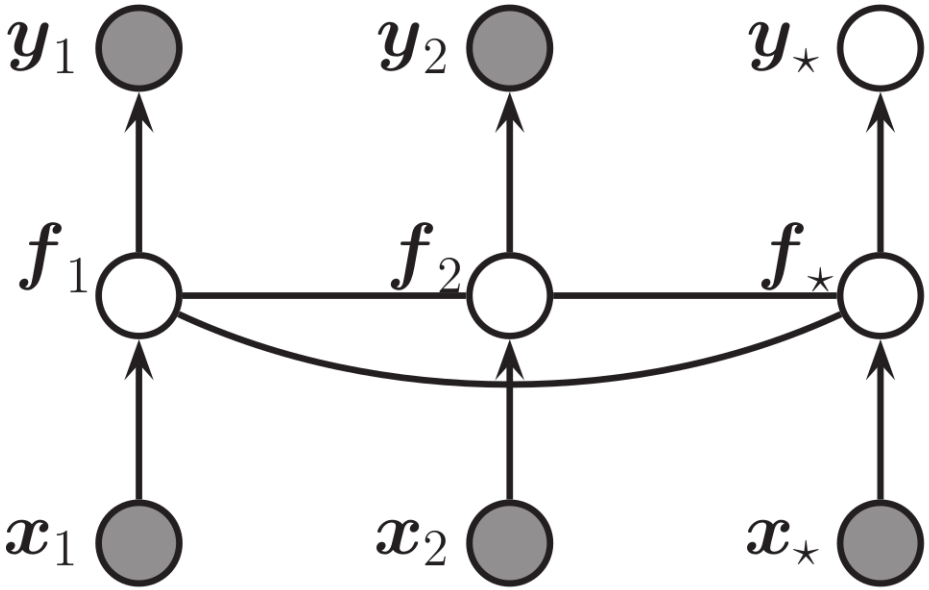
$$\begin{aligned} \boldsymbol{y}^* | \boldsymbol{y} &\sim \mathcal{N}\left(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*\right) \\ \text{with} \quad \boldsymbol{\mu}^* &= \boldsymbol{\mu}(\boldsymbol{x}^*) + \boldsymbol{K}(\boldsymbol{x}^*,\boldsymbol{x})\, \boldsymbol{K}(\boldsymbol{x},\boldsymbol{x})^{-1} \left(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{x})\right) \\ \boldsymbol{\Sigma}^* &= \boldsymbol{K}(\boldsymbol{x}^*,\boldsymbol{x}^*) - \boldsymbol{K}(\boldsymbol{x}^*,\boldsymbol{x})\, \boldsymbol{K}(\boldsymbol{x},\boldsymbol{x})^{-1} \boldsymbol{K}(\boldsymbol{x},\boldsymbol{x}^*) \end{aligned}$$

Kriging, Gaussian process regression (GPR)

Kriging can also be understood as a form of Bayesian inference:

It starts with a prior distribution over functions, that takes the form of a Gaussian process.

Namely, N samples from a function will be normally distributed, where the covariance between any two samples is the covariance function (or kernel) of the Gaussian process evaluated at the spatial location of two points.



$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ y^* \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

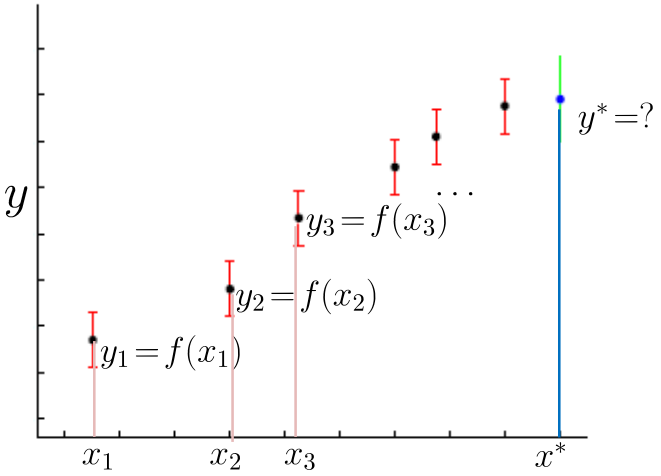
Kriging, Gaussian process regression (GPR)

- **Kriging** or **Gaussian process regression (GPR)** can be viewed as a method of interpolation for which the interpolated values are modeled by a Gaussian process governed by prior covariances.
- Under suitable assumptions on the priors, kriging gives the best linear unbiased prediction of the intermediate values.
- The method originates from geostatistics: the name comes from the Master's thesis of Danie G. Krige, a South African statistician and mining engineer.
- Kriging can also be seen as a spline in a reproducing kernel Hilbert space (RKHS), with the reproducing kernel given by the covariance function.

  Interpretation: the spline is motivated by a minimum norm interpolation based on a Hilbert space structure, while standard kriging is motivated by an expected squared prediction error based on a stochastic model.

Kriging, Gaussian process regression (GPR)

$$\boldsymbol{y}^*|\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*\right)$$
$$\boldsymbol{\mu}^* = \boldsymbol{\mu}(\boldsymbol{x}^*) + \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x})\ \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})^{-1}\ \left(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{x})\right)$$
$$\boldsymbol{\Sigma}^* = \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x})\ \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})^{-1} \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}^*)$$



Interpretation:

A set of values y is first observed, each value associated with a spatial/temporal location **x**.

Now, a new value y* can be predicted at any new spatial/temporal location **x***, by combining the Gaussian prior with a Gaussian likelihood function for each of the observed values.

The resulting posterior distribution is also Gaussian, with a mean and covariance that can be simply computed from the observed values, their variance, and the kernel matrix derived from the prior.

Kriging, Gaussian process regression (GPR)

It is also often assumed in practice that $\begin{bmatrix} \boldsymbol{\mu}(\boldsymbol{x}) \\ \boldsymbol{\mu}(\boldsymbol{x}^*) \end{bmatrix} = \mathbf{0}$.

In this case, Gaussian processes can be completely defined by second-order statistics, where the Gram matrix $\boldsymbol{K}$ is a positive semi-definite covariance.

Note that $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})^{-1}$ can be pre-computed so that the posterior distribution can be compted faster

$$\boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*\right)$$

$$\text{with} \quad \boldsymbol{\mu}^* = \boldsymbol{\mu}(\boldsymbol{x}^*) + \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x})\, \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})^{-1} \left(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{x})\right)$$

$$\boldsymbol{\Sigma}^* = \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x})\, \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})^{-1} \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}^*)$$

Kriging, Gaussian process regression (GPR)

Algorithm 15.1: GP regression

1 $\mathbf{L} = \text{cholesky}(\mathbf{K} + \sigma_y^2 \mathbf{I})$;

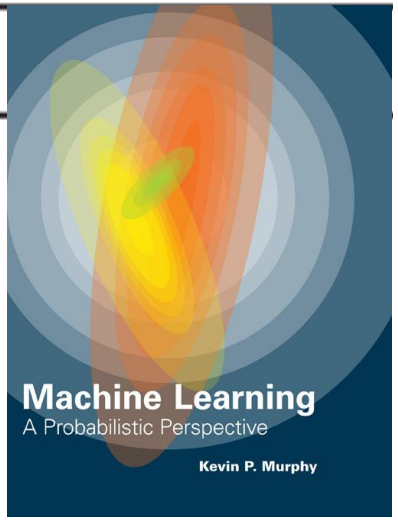
2 $\boldsymbol{\alpha} = \mathbf{L}^T \setminus (\mathbf{L} \setminus \mathbf{y})$;

3 $\mathbb{E}[f_*] = \mathbf{k}_*^T \boldsymbol{\alpha}$;

4 $\mathbf{v} = \mathbf{L} \setminus \mathbf{k}_*$;

5 $\text{var}[f_*] = \kappa(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^T \mathbf{v}$;

6 $\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{N}{2} \log(2\pi)$

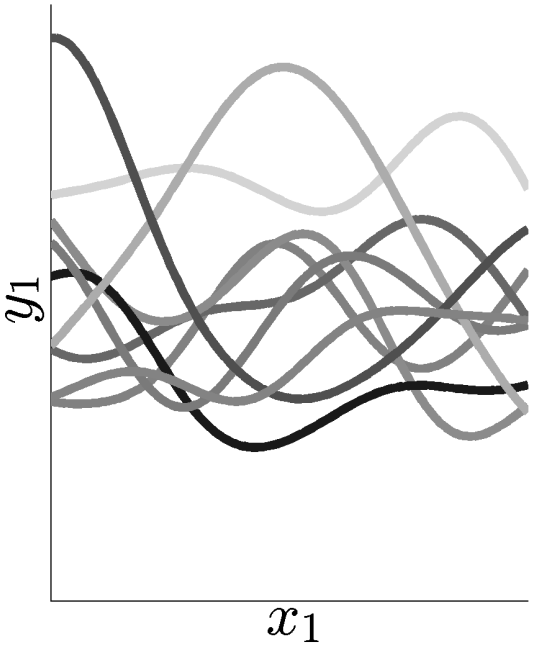


k(xi,xj) as squared exponential covariance

$$\Theta_1^{\text{GP}} = 1, \quad \Theta_2^{\text{GP}} = 0.1, \quad \Theta_3^{\text{GP}} = 0$$

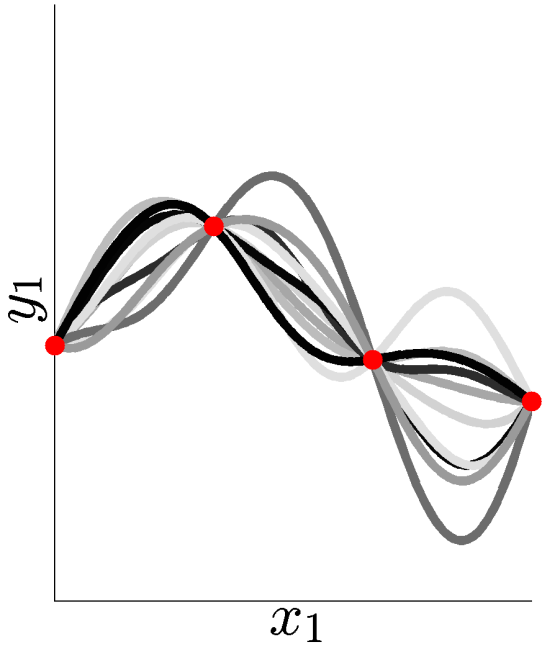
$\boldsymbol{y}^* \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}^*), \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)\big)$

Samples from prior



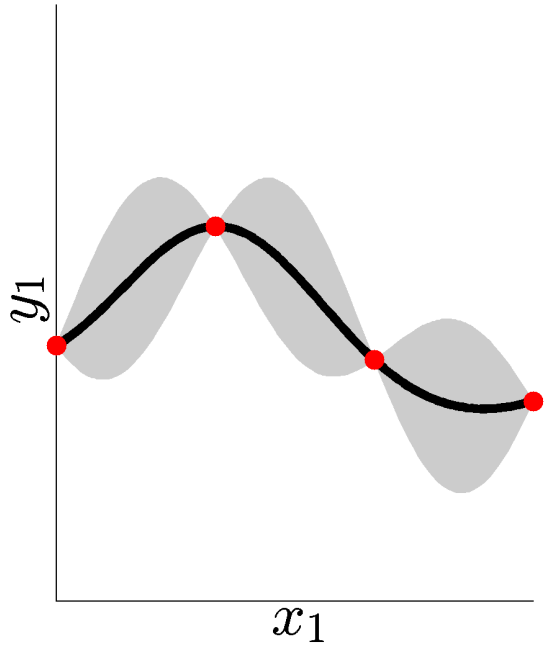
$\boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$

Samples from posterior



$\mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$

Trajectory distribution



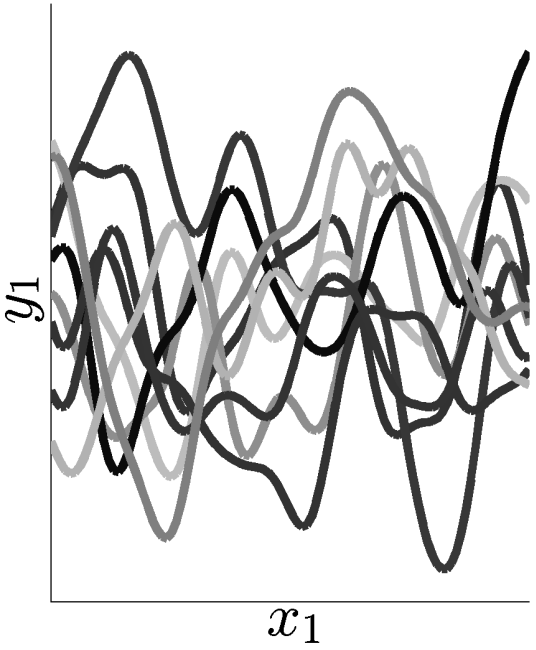
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\text{GP}} \exp\left(-\frac{1}{\Theta_2^{\text{GP}}}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)\right) + \Theta_3^{\text{GP}}\delta_{i,j}$$

k(xi,xj) as squared exponential covariance

$$\Theta_1^{\text{GP}} = 1, \quad \Theta_2^{\text{GP}} = 0.01, \quad \Theta_3^{\text{GP}} = 0$$

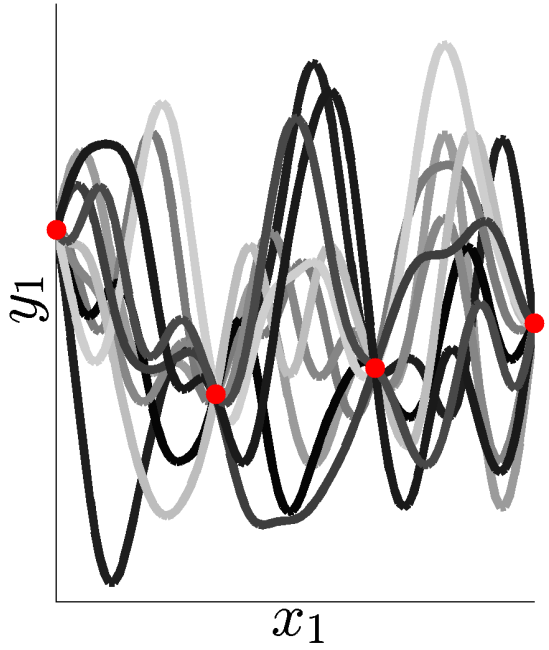
$$\boldsymbol{y}^* \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}^*), \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)\big)$$

Samples from prior



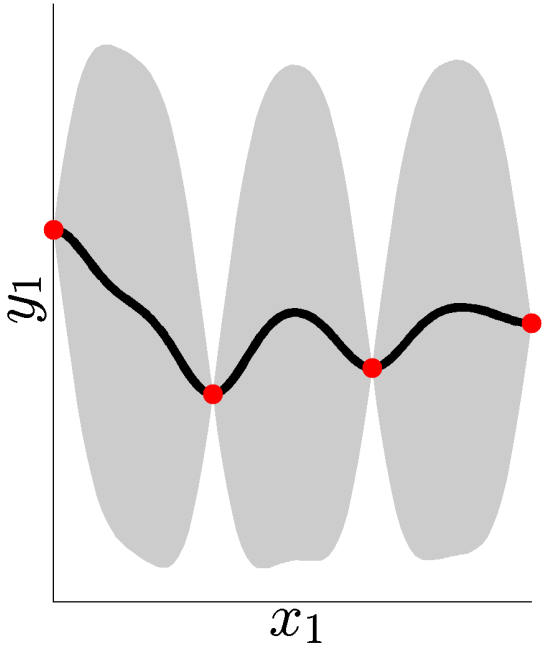
$$\boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$$

Samples from posterior



$$\mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$$

Trajectory distribution



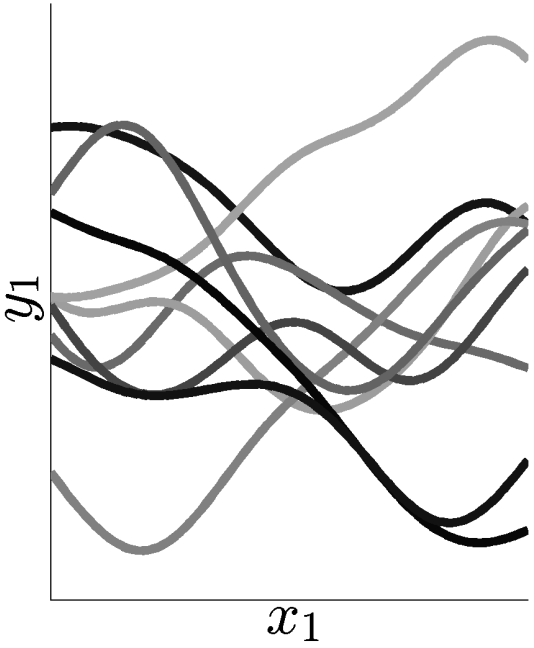
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\text{GP}} \exp\left(-\frac{1}{\Theta_2^{\text{GP}}}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)\right) + \Theta_3^{\text{GP}}\delta_{i,j}$$

k(xi,xj) as squared exponential covariance

$$\Theta_1^{\text{GP}} = 1, \quad \Theta_2^{\text{GP}} = 0.1, \quad \Theta_3^{\text{GP}} = 0.01$$

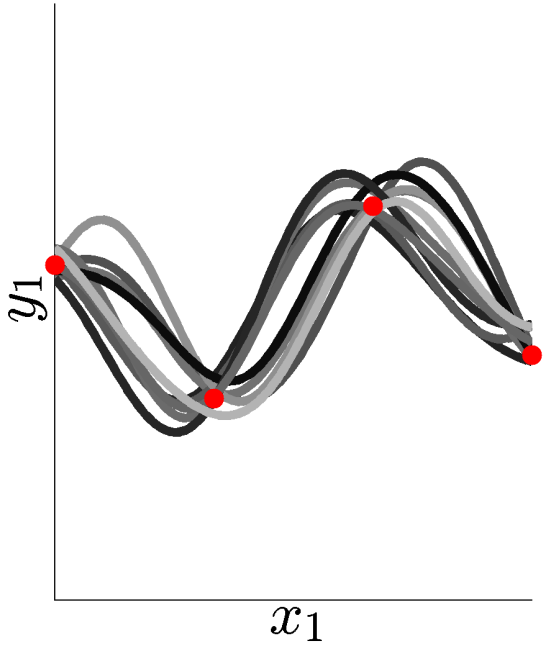
$$\boldsymbol{y}^* \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}^*), \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)\big)$$

Samples from prior



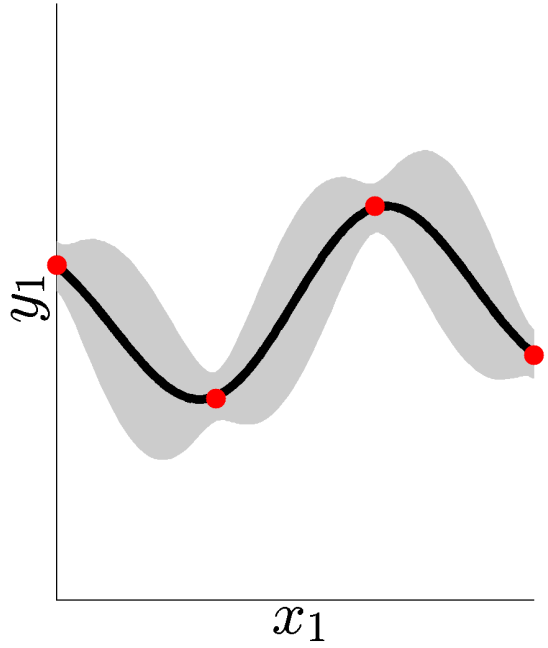
$$\boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$$

Samples from posterior



$$\mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$$

Trajectory distribution



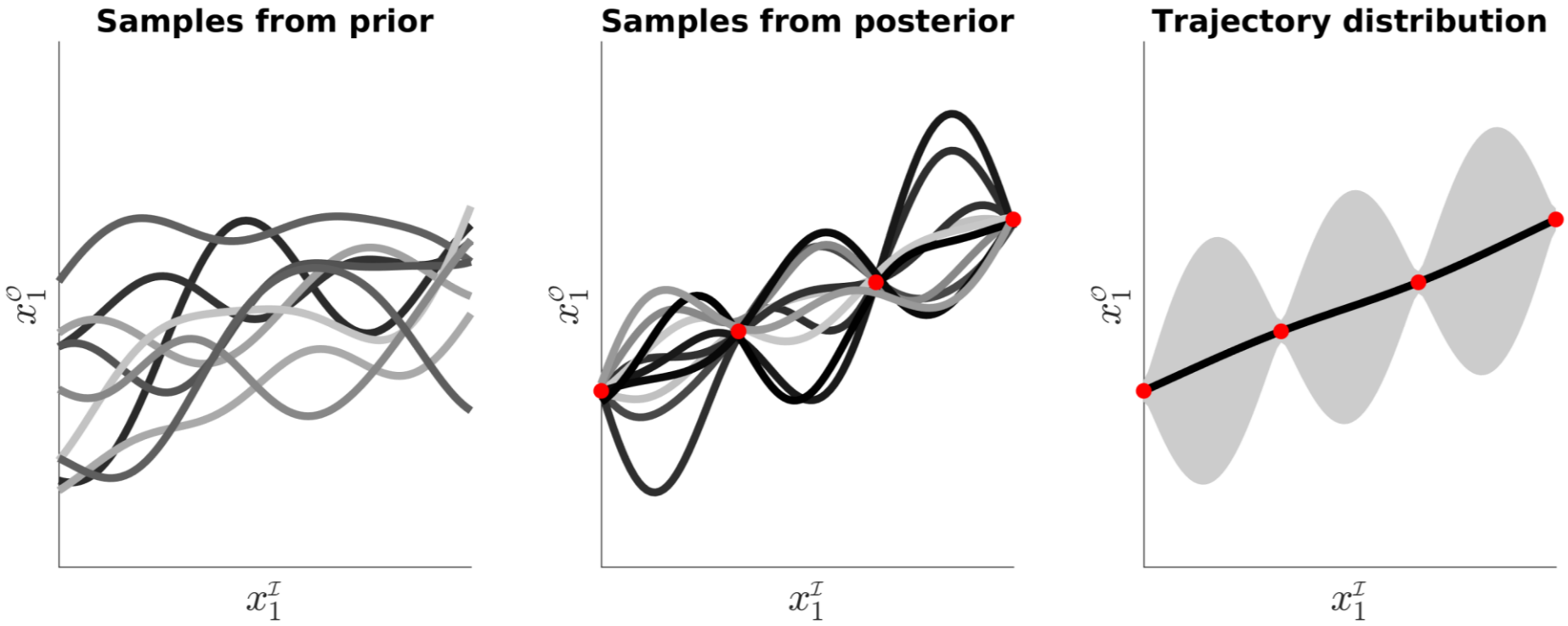
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\text{GP}} \exp\left(-\frac{1}{\Theta_2^{\text{GP}}}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)\right) + \Theta_3^{\text{GP}}\delta_{i,j}$$

k(xi,xj) as squared exponential covariance

$\Theta_1^{\mathrm{GP}} = 1, \quad \Theta_2^{\mathrm{GP}} = 0.1, \quad \Theta_3^{\mathrm{GP}} = 0.01$

$\boldsymbol{\mu(x) = \alpha x}$

$\boldsymbol{y}^* \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}^*), \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)\big)$ $\quad \boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ $\quad \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$



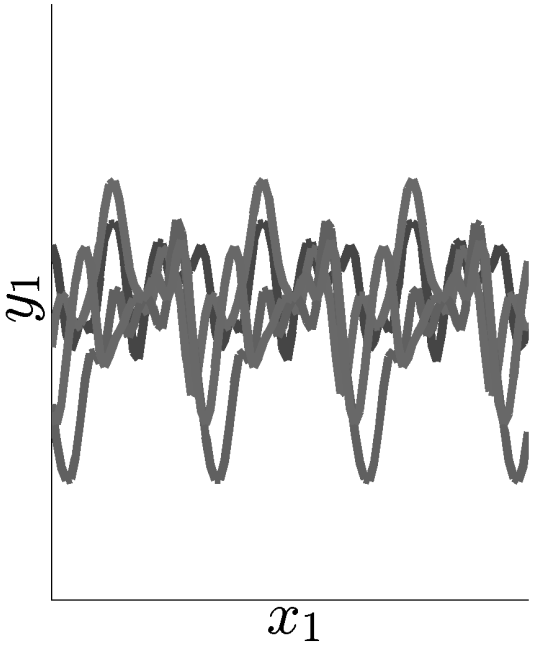
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\mathrm{GP}} \exp\left(-\frac{1}{\Theta_2^{\mathrm{GP}}}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top(\boldsymbol{x}_i - \boldsymbol{x}_j)\right) + \Theta_3^{\mathrm{GP}}\delta_{i,j}$$

k(xi,xj) as periodic covariance function

$$\Theta_1^{\mathrm{GP}} = 0.1, \quad \Theta_2^{\mathrm{GP}} = 0.1, \quad \Theta_3^{\mathrm{GP}} = 0, \quad \Theta_4^{\mathrm{GP}} = 10$$

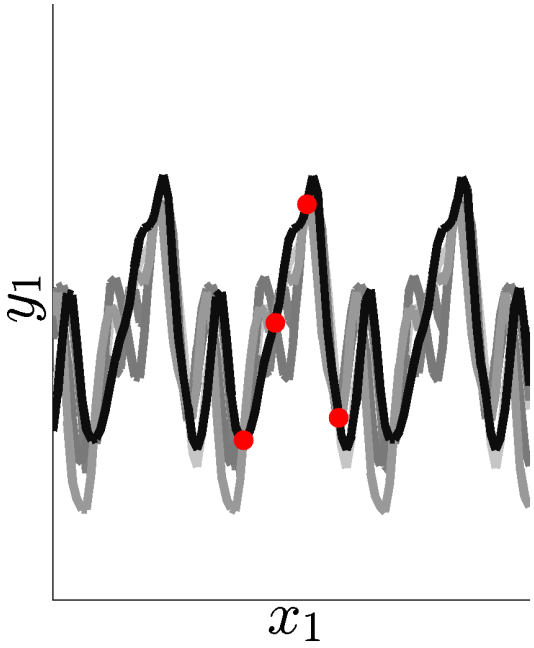
$\boldsymbol{y}^* \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}^*), \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)\big)$

Samples from prior



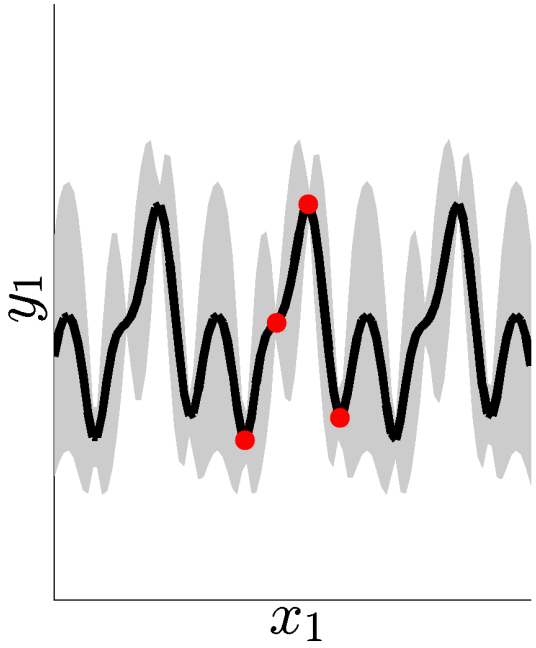
$\boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$

Samples from posterior



$\mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$

Trajectory distribution



$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\mathrm{GP}} \exp\left(-\frac{1}{\Theta_2^{\mathrm{GP}}} \sin^2\left(\Theta_4^{\mathrm{GP}} |\boldsymbol{x}_i - \boldsymbol{x}_j|\right)\right) + \Theta_3^{\mathrm{GP}} \delta_{i,j}$$

k(xi,xj) as Matern covariance function

Another popular covariance kernel function is the Matérn function

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{d}{\rho} \right)^{\nu} K_\nu \left(\sqrt{2\nu} \frac{d}{\rho} \right)$$

$$\text{with } d = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$$

where Γ is the gamma function, K_ν is the modified Bessel function of the second kind, and ρ and ν are non-negative parameters of the covariance.

A Gaussian process with Matérn covariance has sample paths that are $\lfloor \nu - 1 \rfloor$ times differentiable.

As $\nu \to \infty$, the Matérn covariance converges to the squared exponential covariance function.

k(xi,xj) as Matern covariance function

Simplification for ν half integer

When $\nu = p + 1/2,\ p \in \mathbb{N}^+$, the Matérn covariance can be written as a product of an exponential and a polynomial of order p:

$$C_{p+1/2}(d) = \sigma^2 \exp\left(-\frac{\sqrt{2\nu}d}{\rho}\right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\nu}d}{\rho}\right)^{p-i}$$

For $\nu = 1/2$ $(p = 0)$: $C_{1/2}(d) = \sigma^2 \exp\left(-\frac{d}{\rho}\right)$

For $\nu = 3/2$ $(p = 1)$: $C_{3/2}(d) = \sigma^2 \left(1 + \frac{\sqrt{3}d}{\rho}\right) \exp\left(-\frac{\sqrt{3}d}{\rho}\right)$

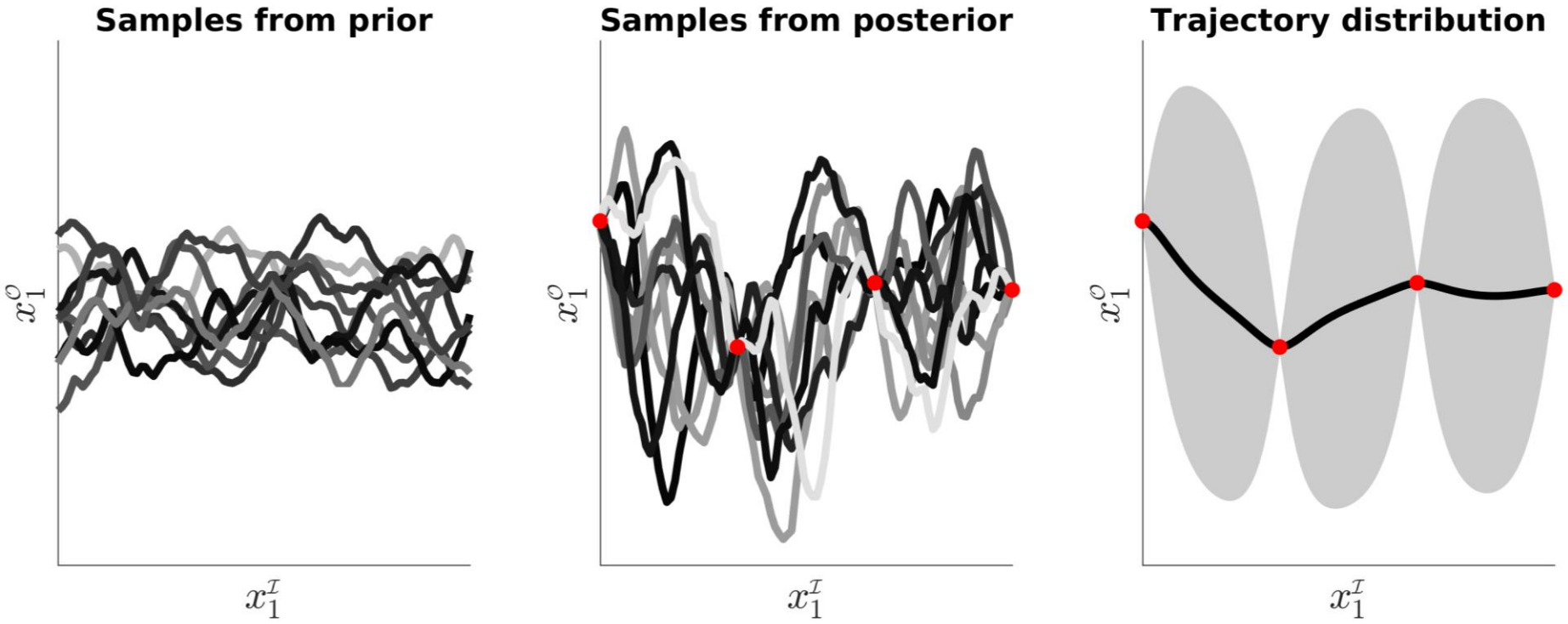
For $\nu = 5/2$ $(p = 2)$: $C_{5/2}(d) = \sigma^2 \left(1 + \frac{\sqrt{5}d}{\rho} + \frac{5d^2}{3\rho^2}\right) \exp\left(-\frac{\sqrt{5}d}{\rho}\right)$

As $\nu \to \infty$, the Matérn covariance converges to the squared exponential covariance function.

k(xi,xj) as Matern covariance function

$$\Theta_1^{\mathrm{GP}} = 0.1, \quad \Theta_2^{\mathrm{GP}} = 0.1, \quad \Theta_3^{\mathrm{GP}} = 0.0001$$

$$\boldsymbol{y}^* \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}^*), \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)\big) \qquad \boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \qquad \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$$

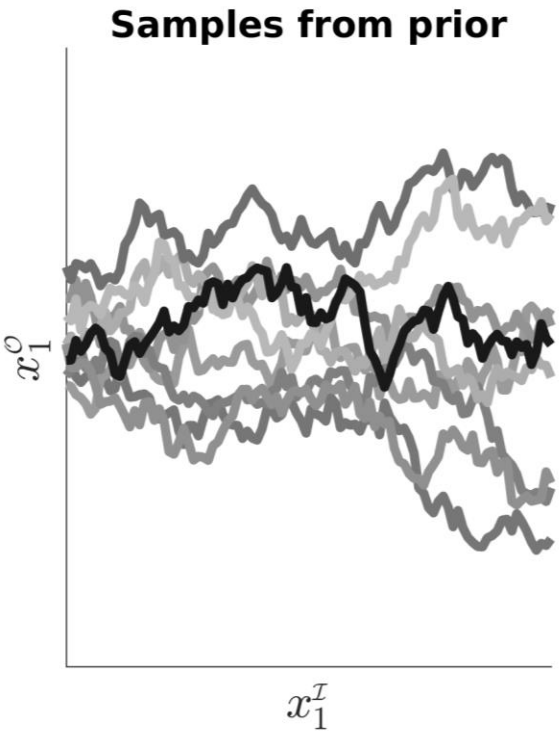


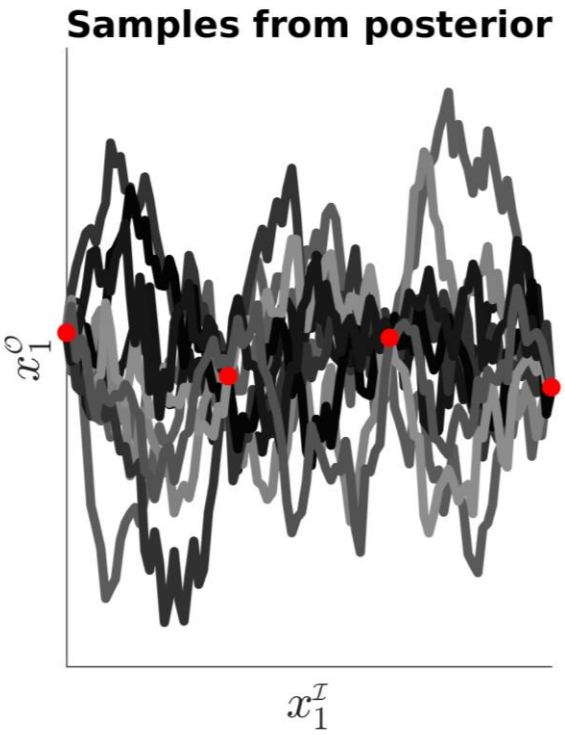
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\mathrm{GP}} \left(1 + \frac{\sqrt{3}\,\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\Theta_2^{\mathrm{GP}}}\right) \exp\left(-\frac{\sqrt{3}\,\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\Theta_2^{\mathrm{GP}}}\right)$$

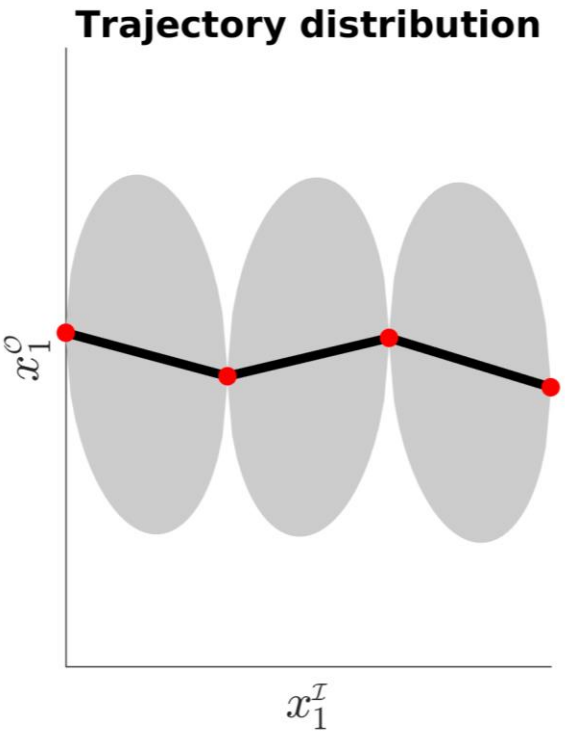
k(xi,xj) as Brownian motion covariance function

The **Wiener process** is a simple continuous-time stochastic process often put in connection to the Brownian motion.

$$\boldsymbol{y}^* \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}^*), \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)\big) \qquad \boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \qquad \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$$





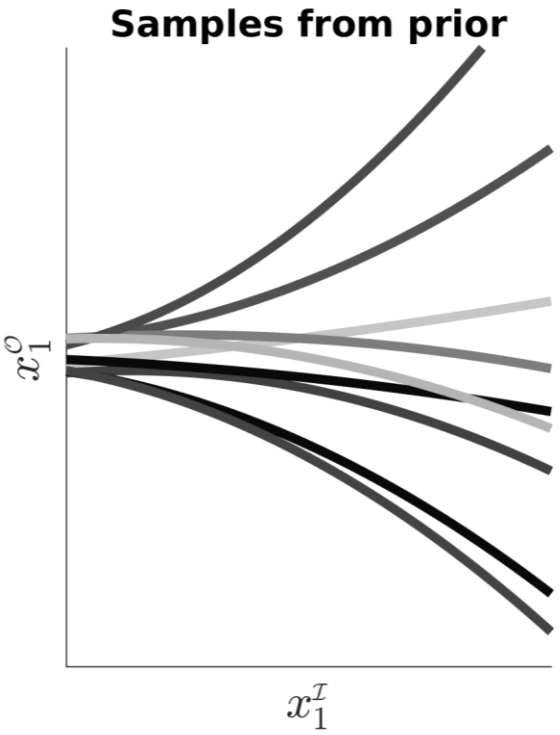


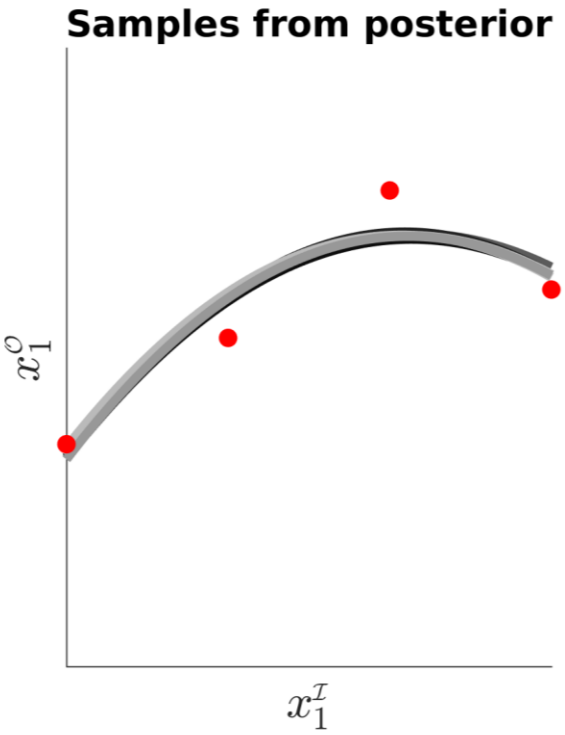
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \min(\boldsymbol{x}_i, \boldsymbol{x}_j) + \Theta_1^{\mathrm{GP}} \qquad \Theta_1^{\mathrm{GP}} = 0.1$$

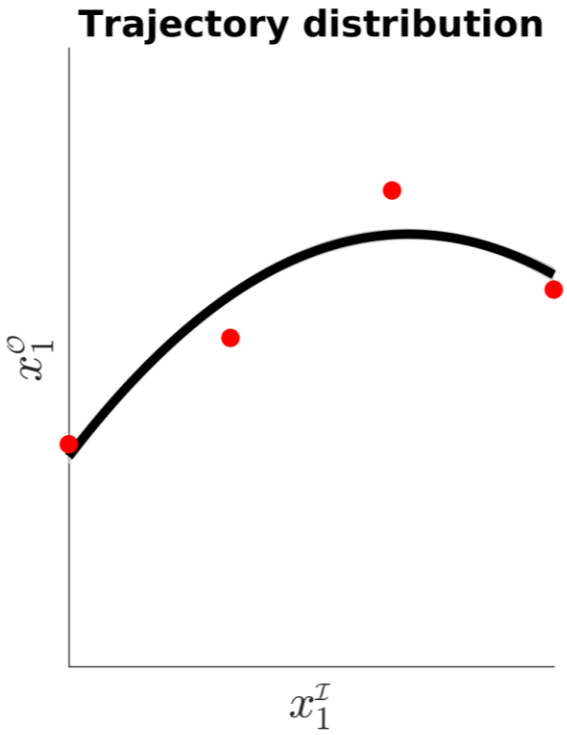
k(xi,xj) as quadratic covariance function

Bayesian linear regression is equivalent to a GP with covariance function $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$.

$$\boldsymbol{y}^* \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}^*), \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)\big) \qquad \boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \qquad \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$$





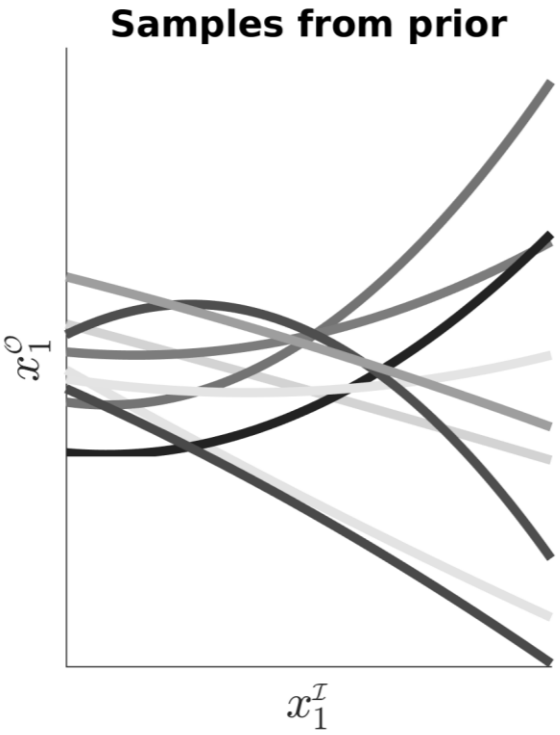


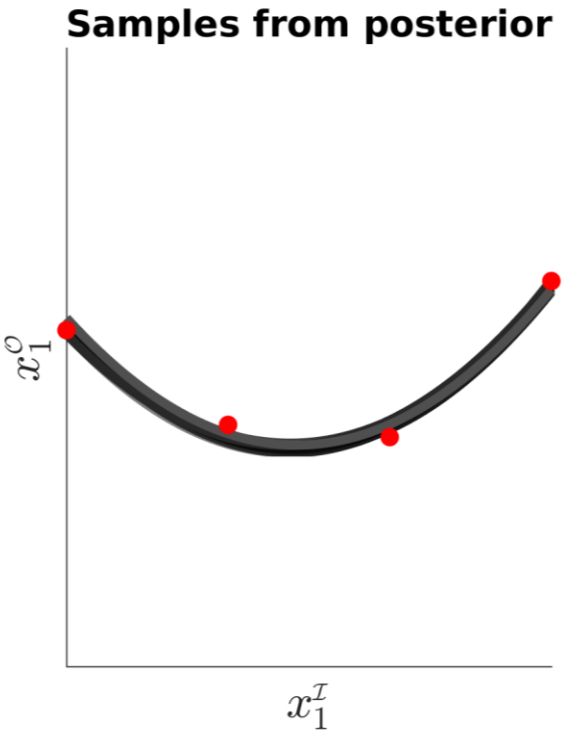
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j + \Theta_1^{\mathrm{GP}})^2 \qquad \Theta_1^{\mathrm{GP}} = 0.1$$

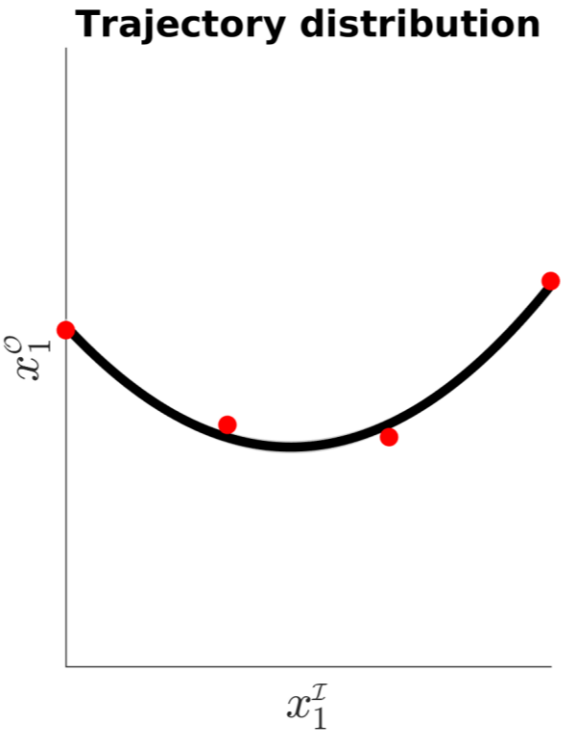
k(xi,xj) as polynomial covariance function

$\Theta_1^{\mathrm{GP}} = 0.1$

$\boldsymbol{y}^* \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}^*), \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)\big)$ $\quad$ $\boldsymbol{y}^* | \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ $\quad$ $\mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$







$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j)^2 + \boldsymbol{x}_i^\top \boldsymbol{x}_j + \Theta_1^{\mathrm{GP}}$$

$k(x_i,x_j)$ as probabilistic model covariance

- Another powerful approach to the construction of kernels is to exploit probabilistic models.
- Given a generative model $P(\mathbf{x})$, a valid kernel can be defined as $k(\mathbf{x}_i, \mathbf{x}_j) = P(\mathbf{x}_i)\, P(\mathbf{x}_j)$, which can be interpreted as an inner product in the one-dimensional feature space defined by the mapping $P(\mathbf{x})$.
- Namely, two inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ will be similar if they both have high probabilities to belong to the model.
- This approach allows the **application of generative models in a discriminative setting**, thus combining the performance of both generative and discriminative models.
- This can bring additional properties to the underlying process such as the capability of handling missing data or partial sequences of various lengths (e.g., with HMM).

k(xi,xj) as weighted sum of kernel functions

- In a more general perspective, it is important to note that a covariance function can be defined as a **linear combination of other covariance functions**, which can be exploited to incorporate **different insights about the dataset**.
- Such an approach can be exploited as an alternative to optimizing kernel parameters (also known as multiple kernel learning). The idea is to define the kernel as a **weighted sum of basis kernels**, and then to **optimize the weights instead of the kernel parameters**.

Dictionary of basis kernel functions

$$\begin{cases} k_1(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j)^2 + \boldsymbol{x}_i^\top \boldsymbol{x}_j \\ k_2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \min(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ k_3(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) \right) \end{cases}$$

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta_1^{\mathrm{GP}} k_1(\boldsymbol{x}_i, \boldsymbol{x}_j) + \Theta_2^{\mathrm{GP}} k_2(\boldsymbol{x}_i, \boldsymbol{x}_j) + \Theta_3^{\mathrm{GP}} k_3(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Extensions of Gaussian processes

- **Cokriging:** Extending GPR to multiple target variables y.
- **Sparse GP:** A known bottleneck in Gaussian process prediction is that the computational complexity of prediction is $O(N^3)$
  → not feasible for large data sets!
  Sparse Gaussian processes circumvent this issue by building a representative set for the given process $y = f(\mathbf{x})$.
- **Wishart process:** The Wishart distribution defines a probability density function over positive definite matrices. The generalised Wishart process (GWP) is a collection of positive semi-definite random matrices indexed by any arbitrary dependent variable. It can for example be used to model time varying covariance matrices $\Sigma(t)$. A draw from a Wishart process is then a collection of matrices indexed by time, similarly to a draw from a GP being a collection of function values indexed by time. Similarly to GP, GWP can capture diverse covariance structures and it can easily handle missing data.

Gaussian process latent variable models (GPLVM)

- GPLVM is a probabilistic dimensionality reduction method that uses GPs to find a lower dimensional non-linear embedding of high dimensional data.
- It is an extension of PPCA, where the model is defined probabilistically, the latent variables are marginalized and the parameters are obtained by maximizing the likelihood.
- As for kernel PCA, GPLVM uses a kernel function to form a non linear mapping (in the form of a GP). However, in GPLVM, the mapping is from the embedded (latent) space to the data space, whereas in kernel PCA, the mapping is in the opposite direction.
- It was originally proposed for visualization of high dimensional data but has been extended to construct various forms of shared manifold models between two observation spaces.

Main references

GPR

C.K.I. Williams and C.E. Rasmussen. Gaussian processes for regression.
In Advances in Neural Information Processing Systems (NIPS), pages 514–520, 1996

C.E. Rasmussen and C.K.I. Williams. Gaussian processes for machine learning. MIT Press, Cambridge, MA, USA, 2006

S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. Philosophical Trans. of the Royal Society A, 371(1984):1–25, 2012

GPLVM

N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of machine learning research, 6:1783-1816, 2005

GWP

A.G. Wilson and Z. Ghahramani. Generalised Wishart processes. Uncertainty in Artificial Intelligence, 2011