Gaussian processes

- **Definition:** A Gaussian process (GP) is a collection of random/unknown variables, any finite number of which have a joint Gaussian pdf.

- **Formally:** A Gaussian process is a generalization of the Gaussian probability distribution. Probability distr. are for scalars/vectors (finite-dim.), Gaussian process are for functions (infinite-dim.)

- **In practice:** Think of functions as "long vectors" $[f(x_1), f(x_2), \ldots]$ and GPs as prob. distr. for such vectors.
  We don't ask questions about $f(x)$ everywhere (would be infinite-dim.), but only at a finite set of points $x_1, x_2, \ldots$.

- GPs are used for both regression and classification

- We'll focus on regression

- [Go through slides from dScience seminar to get the intuitive picture. Slides 27–41.]

- Our basic tool is the multivariate Gaussian pdf
- Refresher on basic definitions and concepts:

General concepts (not restricted to Gaussian pdfs)

- Variance: $Var(x) = E[(x-\mu_x)^2]$, where $\mu_x = E[x]$
- Covariance: $cov(x,y) = E[(x-\mu_x)(y-\mu_y)]$

  [Note: always for a pair of variables]

  - $cov(x,x) = Var(x)$
  - $cov(x,y) = cov(y,x)$
- $\sigma_x = \sqrt{Var(x)}$
- Correlation coeff: $\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y}$

  - $-1 \leq \rho \leq 1$
  - $\rho_{x,y} = \rho_{y,x}$
- Covariance matrix: Matrix with covariances for all pairs of unknown/random variables

$$2D: \quad \Sigma = \begin{bmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{bmatrix} = \begin{bmatrix} \sigma_x^2 & \rho_{x,y}\sigma_x\sigma_y \\ \rho_{x,y}\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}$$

$$1D: \quad \Sigma = \begin{bmatrix} \sigma_x^2 \end{bmatrix}$$

- $\det \Sigma = |\Sigma|$ = "generalized variance" [1D: $\det \Sigma = \sigma_x^2 = Var(x)$]

Multivariate Gaussians:

1-dim: $$p(x) = \mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$= \frac{1}{\sqrt{(2\pi)^1 \sigma^2}} e^{-\frac{1}{2}(x-\mu)\frac{1}{\sigma^2}(x-\mu)}$$

N-dim: $$p(\bar{x}) = \mathcal{N}(\bar{x};\bar{\mu},\Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu})^T \Sigma^{-1} (\bar{x}-\bar{\mu})}$$

- Written out for 2D case:

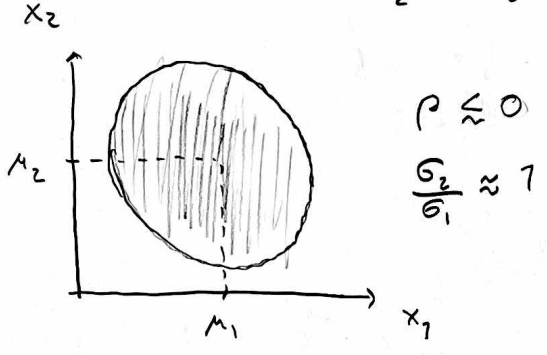
$$p(x_1,x_2) = \mathcal{N}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma\right)$$

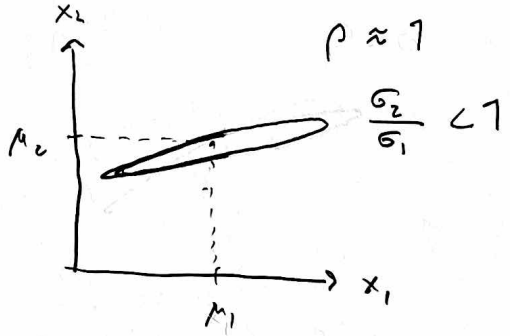
$$= \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right) + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right]}$$

- Notice that for $\rho \to 0$, $p(x_1,x_2) \to p(x_1)\,p(x_2) = \mathcal{N}(x_1;\mu_1,\sigma_1)\,\mathcal{N}(x_2;\mu_2,\sigma_2)$

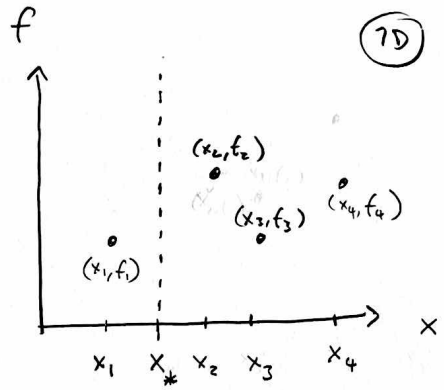
- As $\rho \to \pm 1$, the joint distr. $p(x_1,x_2)$ is "squeezed" towards falling on the line

$$x_2 = \text{sgn}(\rho)\,\frac{\sigma_2}{\sigma_1}(x_1-\mu_1) + \mu_2$$





GP regression



Notation

- $\bar{x}$: input point in m-dim space (feature space)
- $f(\bar{x})$: unknow, true function (target space)
- $\bar{x}_i$: input points where we know $f(\bar{x}_i)$
- $f_i \equiv f(\bar{x}_i)$: the known function values
- $D = \{\bar{x}_i, f_i\}_{i=1}^{n}$: training set with n points
- $X = \begin{bmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_n \end{bmatrix}$: n × m matrix with all input components in training set (n points in m-dim space)
- $\bar{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}$: n-vector with all known f values
- So can say $D = \{X, \bar{f}\}$
- $\bar{x}_*$: test point, i.e. point where we will estimate the unknown true function value $f_* \equiv f(\bar{x}_*)$
- Start by formulating <u>joint prior</u> for f values at <u>all</u> relevant $\bar{x}$ points :

$$p(\bar{f}, f_* | X, \bar{x}_*) \equiv p\left((n+1)\text{-dim. pdf}\right)$$

Notation : Here we assume that we always know the <u>input</u> points (x points), so we can also write

$$p(D, f_* | \bar{x}_*)$$

[In other words, $p(D) \equiv p(\bar{f}|X)$]

- Require our joint prior to be an $(n+1)$-dimensional Gaussian pdf
- To specify it, we need to specify

  1) An $(n+1)$-dim. mean vector $\bar{\mu} = \begin{bmatrix} \mu_{f_1} \\ \vdots \\ \mu_{f_n} \\ \mu_{f_*} \end{bmatrix}$

  2) An $(n+1) \times (n+1)$ covariance matrix for all the pairs of f variables

- We do this indirectly by rather choosing a mean function and a covariance function (kernel) that give means and covariances for f values at arbitrary input points
  - Mean function: $m(\bar{x}) = E[f(\bar{x})] \equiv \int_{-\infty}^{\infty} f \, p(f|\bar{x}) \, df$
  - Covar. function: $k(\bar{x}, \bar{x}') = E\left[(f(\bar{x}) - m(\bar{x}))(f(\bar{x}') - m(\bar{x}'))\right]$
- Keep in mind: $m(\bar{x})$ and $k(\bar{x}, \bar{x}')$ are functions of inputs in $\bar{x}$ space, but the function values represent mean and covars. in f space.

- Our joint prior is then

$$p(\bar{f}, f_* \mid X, \bar{x}_*) = p\left(\begin{bmatrix} \bar{f} \\ f_* \end{bmatrix} \middle| \begin{bmatrix} X \\ \bar{x}_* \end{bmatrix}\right)$$

$$= N\left(\begin{bmatrix} m(X) \\ m(\bar{x}_*) \end{bmatrix}, \begin{bmatrix} \Sigma & k(X, \bar{x}_*) \\ k(\bar{x}_*, X) & k(\bar{x}_*, \bar{x}_*) \end{bmatrix}\right)$$

$(n+1)$-dim mean vector

$(n+1) \times (n+1)$ covariance matrix

where we have used this notation:

- $m(X) \equiv \begin{bmatrix} m(\bar{x}_1) \\ \\ m(\bar{x}_n) \end{bmatrix}$

- $\Sigma \equiv k(X, X) \equiv \begin{bmatrix} k(\bar{x}_1, \bar{x}_1) & \dots & k(\bar{x}_1, \bar{x}_n) \\ & & \\ k(\bar{x}_n, \bar{x}_1) & \dots & k(\bar{x}_n, \bar{x}_n) \end{bmatrix}$

- $k(X, \bar{x}_*) \equiv \begin{bmatrix} k(\bar{x}_1, \bar{x}_*) \\ \\ k(\bar{x}_n, \bar{x}_*) \end{bmatrix}$

- $k(\bar{x}_*, X) \equiv k(X, \bar{x}_*)^T$

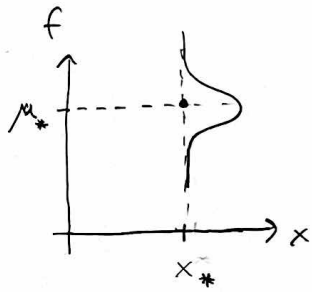
- The choice and optimization of the <u>covar. function</u> (and mean function) constitute the main challenge in GP regression. (We'll get back to that)

- Assume we have chosen a specific $m(\bar{x})$ and $k(\bar{x}, \bar{x}')$, i.e. we have a <u>fully specified prior</u> $p(\bar{f}, f_* \mid X, \bar{x}_*)$.

- Our goal: A predictive distr. for the unknown f_*

- Get it from "looking at" the known data $\bar{f}$, i.e. we derive from the prior the <u>conditional</u> <u>1-dim</u> pdf $p(f_* \mid \bar{f}, X, \bar{x}_*)$

- Gaussian prior $\longrightarrow$ conditional pdf also Gaussian



$$p(f_* \mid \bar{f}, X, \bar{x}_*) = \mathcal{N}(\mu_*, \sigma_*^2)$$

where

$$\mu_* = m(\bar{x}_*) + k(\bar{x}_*, X)\,\Sigma^{-1}\left(\bar{f} - m(X)\right)$$ ← point prediction for unknown f_*

$$\sigma_*^2 = k(\bar{x}_*, \bar{x}_*) - k(\bar{x}_*, X)\,\Sigma^{-1}\,k(X, \bar{x}_*)$$ ← prediction uncertainty

- <u>Intuition:</u> $\mu_* =$ [prior mean at $\bar{x}_*$] + [shift given by weighted sum of how much known f values ($\bar{f}$) where shifted from their prior means $m(X)$. The weights dep. on covariance of f_* and f_i.]

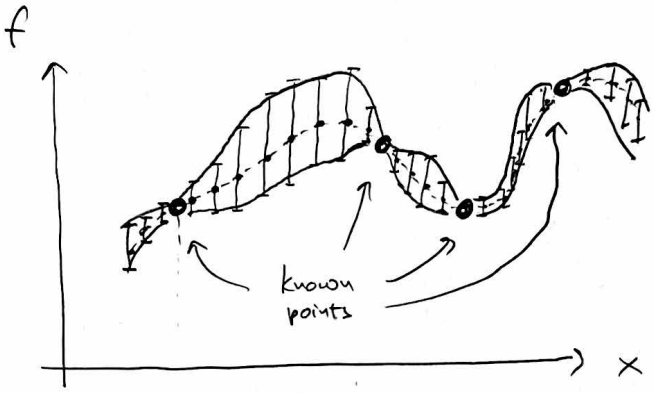
$$\sigma_*^2 = [\text{prior variance } k(\bar{x}_*, \bar{x}_*)] - [\text{reduction due to the additional information about } f(\bar{x}_*) \text{ provided by the training data. Only depends on covar. function.}]$$

Note:
we need the inverse $n \times n$ matrix Σ^{-1}.
Source of main comp. challenge for GP regression

Note: If $k(\bar{x}_*, \bar{x}_7) = 0$, i.e. $\mathrm{Cov}(f_*, f_7) = 0$, then our knowledge of f_7 tells us nothing about f_*, so it doesn't contribute to reducing σ_*

- Repeat prediction for many differen points $\bar{x}_*$

  → get the familiar GP-style plots



- **Keep in mind:** That our prediction for the unknown f_* is given as a pdf does not imply that $f(\bar{x})$ is an indeterministic function. It simply expresses our degree of belief about the true but unknown $f(\bar{x}_*)$.

- GPs are often used for problems with noisy data

$$Y(\bar{x}) = f(\bar{x}) + \varepsilon \qquad \text{where } \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

and often there is confusion about what we are predicting, i.e. whether our posterior is

$$p(y_* \mid \bar{y}, X, \bar{x}_*)$$ degree of belief about unknown, noisy data point

or

$$p(f_* \mid \bar{y}, X, \bar{x}_*)$$ degree of belief about the true function without noise

(More on this later)

[end lecture here]

Choosing and optimising covariance function (kernel)

- By choosing kernel we define our prior on function space.
- Encodes our expectations about the true function, but without specifying and assumed functional form for the true $f(\bar{x})$
- Kernels encode expectations about e.g.
  - smoothness
  - typical length scales
  - periodicity
  - trend (increase/decrease)
  - symmetry

  etc.
- This is the main modelling step in GP regression!

[Choice of mean function $m(\bar{x})$ usually much less important. Often set to $m(\bar{x}) = 0$ or $m(\bar{x}) = \text{constant}$. But keep in mind that predictions are "pulled" towards $m(\bar{x})$ in regions of $\bar{x}$ space far away from known points]

- Need a fully specified prior → fully specified kernel
- Common approach:

  1) Construct a kernel with some free params. (hyperparameters)

  2) Fit these hyperparameters in a max. likelihood fit using training data (the training step in GPR)

- Note that this is a bit "un-Bayesian".

  Proper Bayesian approach: Assign priors for the hyperparameters and marginalise over them

  $\bar{\theta}$: hyperparameters

$$\text{GP posterior} = p(f_* \mid \bar{f}, X, \bar{x}_*) = \int p(f_*, \bar{\theta} \mid \bar{f}, X, \bar{x}_*)\, d\bar{\theta}$$

$$= \int p(f_* \mid \bar{\theta}, \bar{f}, X, \bar{x}_*)\, p(\bar{\theta} \mid \bar{f}, \bar{X})\, d\bar{\theta}$$

$$\propto \int p(f_* \mid \bar{\theta}, \bar{f}, X, \bar{x}_*)\, p(\bar{f} \mid \bar{\theta}, X)\, p(\bar{\theta})\, d\bar{\theta}$$

- But this integration is usually very expensive

- So common approach is to instead use a point est. for $\bar{\theta}$, namely the $\bar{\theta}$ value that maximises the likelihood

$$L(\bar{\theta}) = p(\bar{f} \mid \bar{\theta}, X) = N(m(X), \Sigma(\bar{\theta}))$$

- Effectively like "peeking" at the data and setting a delta function prior $p(\bar{\theta}) = \delta(\bar{\theta} - \bar{\theta}_{ML})$. ("Empirical Bayes")

- Usually maximise the log-likelihood

$$\ln L(\bar{\theta}) = -\frac{1}{2}(\bar{f} - m(X))^T \Sigma^{-1}(\bar{\theta})(\bar{f} - m(X)) - \frac{1}{2}\ln|\Sigma(\bar{\theta})| - \frac{n}{2}\ln(2\pi)$$

- Note: For every $\bar{\theta}$ we try we need a new matrix inverse $\Sigma^{-1}(\bar{\theta})$ and determinant $|\Sigma(\bar{\theta})|$

  - Standard techniques for matrix inversion: $O(n^3)$
  - Can use Cholesky decomp. technique instead: $O(n^2)$

  Still very slow when n grows large [See algo. 2.1 in Rasmussen and Williams]

Some kernels

[Browse scikit-learn documentation to see kernel examples.]

1) The squared-exponential kernel

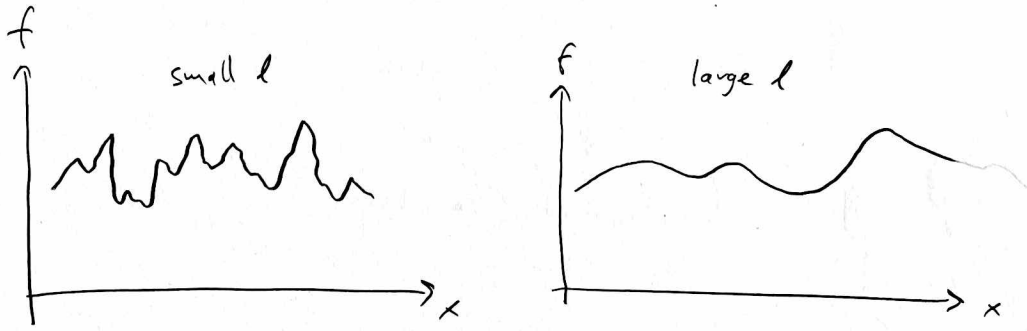
$$k(\bar{x}, \bar{x}') = \sigma_f^2 \, e^{-\frac{1}{2}\frac{(\bar{x}-\bar{x}')^2}{\ell^2}}$$

Hyperpars.:

$$\bar{\theta} = \left\{ \sigma_f^2, \ell \right\}$$

As the Euclidean dist. between the input points $\bar{x}, \bar{x}'$ increases, the covariance between $f(\bar{x})$ and $f(\bar{x}')$ decreases exponentially.

- Universal kernel (can approx any cont. function given enough data)
- Prefers very smooth functions
- σ_f^2 : Scale factor, sets average dist. away from the mean function
- ℓ : Length scale, sets how quickly correlation between f-values drop as $\bar{x}$-dist. increase. Sets the typical length scale for "wiggles" in the function



- Example of stationary kernel, i.e. only depends on distance between $\bar{x}$ and $\bar{x}'$, not the values of $\bar{x}$ and $\bar{x}'$ in an absolute sense. (The kernel acts the same across all of $\bar{x}$ space). A non-stationary kernel depends on the specific locations of $\bar{x}$ and $\bar{x}'$ in $\bar{x}$ space.

2) The Matérn kernel

$$k(\bar{x},\bar{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left[\sqrt{2\nu}\,\frac{|\bar{x}-\bar{x}'|}{\ell}\right]^{\nu} K_\nu\left[\sqrt{2\nu}\,\frac{|\bar{x}-\bar{x}'|}{\ell}\right]$$

- Hyperparams: $\bar{\theta} = \{\nu, \ell\}$
- $\Gamma(\nu)$: gamma function
- K_ν: modified Bessel function
- Common choices: $\nu = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}$
- ν sets smoothness, $\nu \to \infty$ corresponds to the squared exp. kernel

- Other common kernels
  - Linear kernel
  - Periodic kernel
  - Rational quadratic kernel (effectively infinite sum of squared-exp. kernels w/ different length scales)
  - Noise kernel

- Kernels can be summed and multiplied to construct new kernels

$k(\bar{x},\bar{x}') = k_1(\bar{x},\bar{x}')\,k_2(\bar{x},\bar{x}')$: "AND operator" (Both k_1 and k_2 must be large for k to be large)

$k(\bar{x},\bar{x}') = k_1(\bar{x},\bar{x}') + k_2(\bar{x},\bar{x}')$: "OR operator"

- Can use different kernel components for different input components, e.g. with $\bar{x} = [x_1, x_2]$

$$k(\bar{x}, \bar{x}') = e^{-\frac{1}{2}\frac{(x_1 - x_1')^2}{l_1^2}} \; e^{-\frac{1}{2}\frac{(x_2 - x_2')^2}{l_2^2}}$$

to allow different lengthscales in different directions in $\bar{x}$ space. (But end up with more hyperpars. to determine.)

Noisy data

- So far we've focused on the case of <u>noise-free data</u>, i.e. training data were values of the true $f(\bar{x})$ directly.

$$\Rightarrow \text{Got posterior } p(f_* \mid \bar{f}, X, \bar{x}_*)$$

- In this case $p(f_* \mid \bar{f}, X, \bar{x}_*) \to \delta(f_* - f_i)$ when $\bar{x}_* \to \bar{x}_i$, i.e. posterior collapses to delta function when $\bar{x}_*$ is a known point

- Reasonable in theory (given assumption of training points with no uncert.), but in practice numerically problematic

- Need to allow for uncertainty in data, either because this is actually the case, or just for numerical stability

- Assume data y_i are from

$$Y_i \equiv Y(\bar{x}_i) = f(\bar{x}_i) + \varepsilon_i$$

ε_i: noise, $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, σ_ε^2: noise level

[Note: We say "noise", but there doesn't have to be randomness involved. $\mathcal{N}(0, \sigma_\varepsilon^2)$ could just express our degree of certainty in the training values]

- Now we must distinguish between

1) $p(f_* | \bar{y}, X, \bar{x}_*)$: Degree of belief in underlying true function value f_* at $\bar{x}_*$

and

2) $p(y_* | \bar{y}, X, \bar{x}_*)$: Degree of belief in value for a data point y_* at $\bar{x}_*$

- If we only add the noise variance to the <u>training set</u> part of the covariance matrix, we get case 1:

$$\Sigma \rightarrow \Sigma + \sigma_\varepsilon^2 I$$

[Note: we can use different uncert. at different training points.]

$$\Rightarrow \quad p(f_* | \bar{y}, X, \bar{x}_*) = \mathcal{N}(\mu_*, \sigma_*^2)$$

where

$$\mu_* = m(\bar{x}_*) + k(\bar{x}_*, X)\left[\Sigma + \sigma_\varepsilon^2 I\right]^{-1}(\bar{y} - m(X))$$

$$\sigma_*^2 = k(\bar{x}_*, \bar{x}_*) - k(\bar{x}_*, X)\left[\Sigma + \sigma_\varepsilon^2 I\right]^{-1} k(X, \bar{x}_*)$$

- If we add noise variance to <u>both</u> Σ and $k(\bar{x}_*, \bar{x}_*)$ we get case 2:

$$\Sigma \rightarrow \Sigma + \sigma_\varepsilon^2 I$$

$$k(\bar{x}_*, \bar{x}_*) \rightarrow k(\bar{x}_*, \bar{x}_*) + \sigma_\varepsilon^2$$

$$\Rightarrow \quad p(y_* \mid \bar{y}, X, \bar{x}_*) = N(\mu_*, \sigma_*^2)$$

where

$$\mu_* = m(\bar{x}_*) + k(\bar{x}_*, X)\left[\Sigma + \sigma_\varepsilon^2 I\right]^{-1}(\bar{y} - m(X)) \qquad \left(\text{same as before}\right)$$

$$\sigma_*^2 = k(\bar{x}_*, \bar{x}_*) + \sigma_\varepsilon^2 - k(\bar{x}_*, X)\left[\Sigma + \sigma_\varepsilon^2 I\right]^{-1} k(X, \bar{x}_*)$$

↑ note!

- Common approach: Try to learn the typical noise level from data by treating σ_ε^2 as a hyperparameter, i.e. <u>add</u> a kernel term of the form $k(\bar{x}, \bar{x}') = \delta_{\bar{x}\bar{x}'}\,\sigma_\varepsilon^2$

- Pitfall: In hyperparameter space, there will then often be a huge region towards <u>large σ_ε^2</u> where the model can obtain OK fit to data simply by "assuming" that all variation in the data is due to noise.

[See documentation of scikit-learn for example.]