Bayesian methods (2)

Bayesian linear regression
Gaussian process regression

Juho Lee
Grad school of AI
Samsung AI-expert course

Before we start

Open your terminal and type

```
git clone https://github.com/juho-lee/samsung_AI_expert/
```

You can find this new slide here.

Preliminary

Gaussian (Normal) distribution

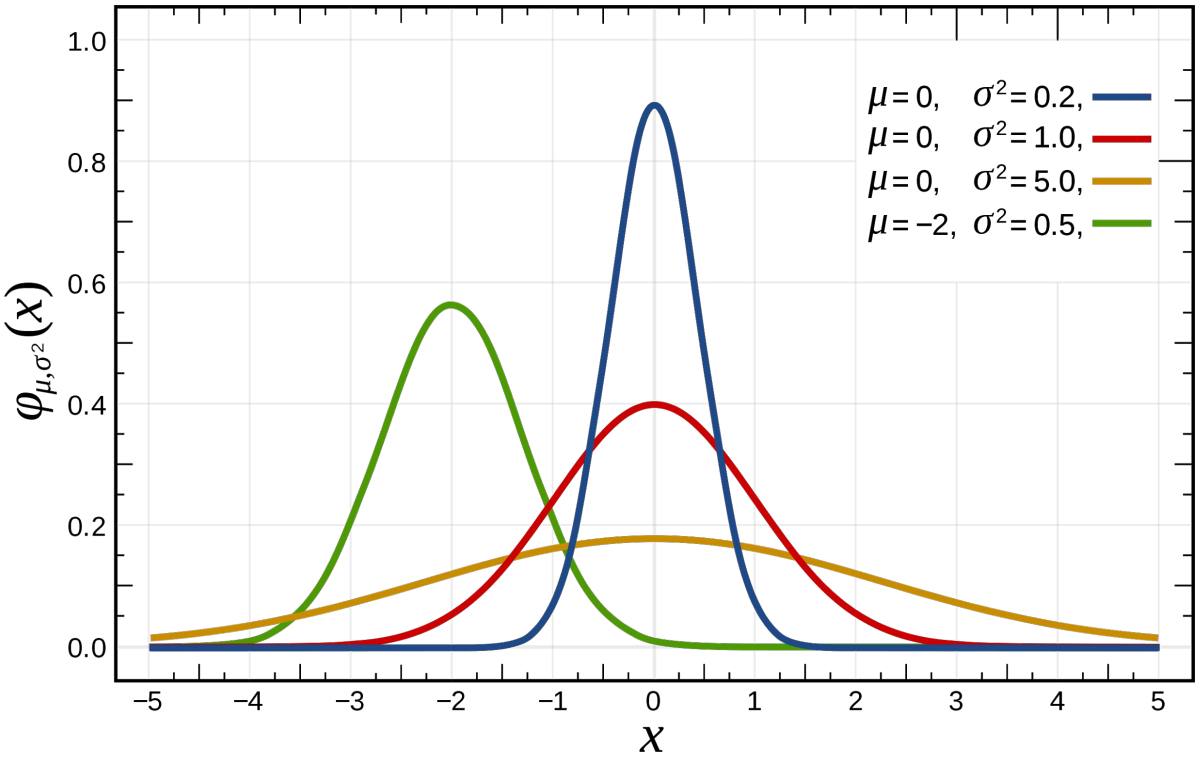
Univariate Gaussian distribution

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right)$$

$$\mathbb{E}[x] = \mu$$

$$\mathrm{Var}(x) = \mathbb{E}[(x-\mu)^2] = \sigma^2.$$

Univariate Gaussian distribution



https://upload.wikimedia.org/wikipedia/commons/7/74/Normal_Distribution_PDF.svg

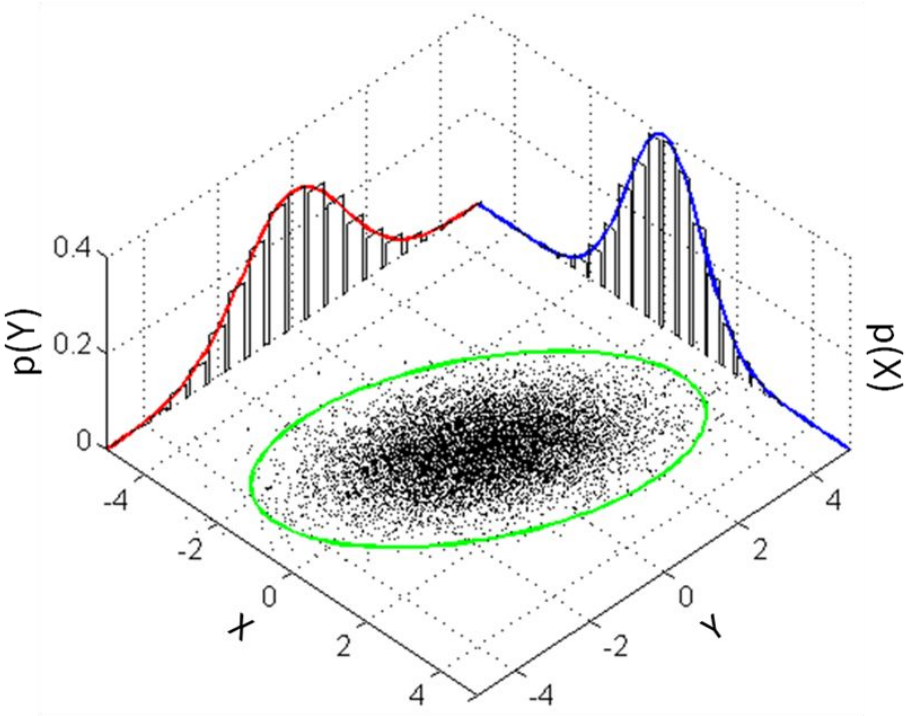
Multivariate Gaussian distribution

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \right)$$

$$\mathbf{x} = [x_1, \ldots, x_d]^\top \qquad \boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^\top$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_{11} & \ldots & \Sigma_{1d} \\ \vdots & \ddots & \vdots \\ \Sigma_{d1} & \ldots & \Sigma_{dd} \end{bmatrix}$$

Multivariate Gaussian distribution



https://commons.wikimedia.org/wiki/File:MultivariateNormal.png

Multivariate Gaussian distribution

Mean and covariance:

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$$

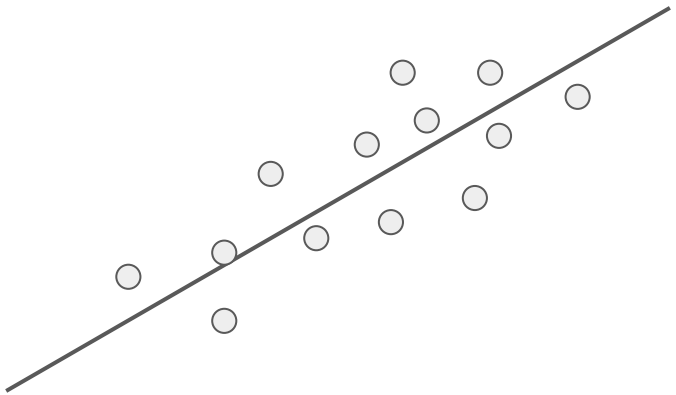
$$\mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top] = \boldsymbol{\Sigma}$$

Bayes' Theorem

$$\underbrace{\boxed{P(Y|X)}}_{\text{Posterior}} = \frac{\overbrace{\boxed{P(X|Y)}}^{\text{Likelihood}}\overbrace{\boxed{P(Y)}}^{\text{Prior}}}{P(X)}$$

Bayesian linear regression

Linear regression - setting



$$\mathbf{x}_i = [1, x_{i,1}, x_{i,2}, \ldots, x_{i,d}]^\top \in \mathbb{R}^{d+1}$$

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times (d+1)}$$

$$\mathbf{y} = [y_1, y_2, \ldots, y_n] \in \mathbb{R}^n$$

$$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}, \quad \mathbf{w} \in \mathbb{R}^{d+1}.$$

Linear regression - Least square method

Find the parameter by minimizing the squared error.

$$\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 = \min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$$

Exercise: the least square solution is given as

$$\mathbf{w}^\star = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Ridge regression

The inverse can be problematic.

$$\mathbf{w}^\star = \boxed{(\mathbf{X}^\top \mathbf{X})^{-1}} \mathbf{X}^\top \mathbf{y}$$

Ridge regression: add a regularization term.

$$\begin{aligned} \mathbf{w}^\star &= \min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda \|\mathbf{w}\|^2 \\ &= \boxed{(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1}} \mathbf{X}^\top \mathbf{y} \end{aligned}$$

Linear regression as a maximum likelihood

Linear regression as a probabilistic model

$$p(y|\mathbf{x};\mathbf{w}) = \mathcal{N}(y;\mathbf{x}^\top\mathbf{w},\beta^{-1}).$$

Maximum likelihood (ML) objective:

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{X};\mathbf{w}) &= \sum_{i=1}^{n}\log p(y_i|\mathbf{x}_i;\mathbf{w}) \\ &= \boxed{-\frac{\beta}{2}\sum_{i=1}^{n}(y_i-\mathbf{x}_i^\top\mathbf{w})^2} + \text{const.}\end{aligned}$$

Ridge regression as a maximum a posteriori

Prior distribution on the parameter

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; 0, \gamma \mathbf{I}).$$

Maximum a posteriori (MAP) objective:

$$\log p(\mathbf{y}|\mathbf{X}; \mathbf{w}) + \log p(\mathbf{w}) = \sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i; \mathbf{w}) + p(\mathbf{w})$$

$$= -\frac{\beta}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \mathbf{w})^2 - \frac{\gamma}{2} \|\mathbf{w}\|^2 + \text{const}$$

ML vs MAP vs Bayesian

ML: treats $\mathbf{w}$ as a fixed parameter, and computes a point-estimate $\mathbf{w}^\star$.

MAP: treats $\mathbf{w}$ as a random variable, but computes a point-estimate $\mathbf{w}^\star$.

Bayesian: treats $\mathbf{w}$ as a random variable, and computes the posterior distribution,

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})}$$

ML vs MAP vs Bayesian

For a novel observation $\mathbf{x}_*$,

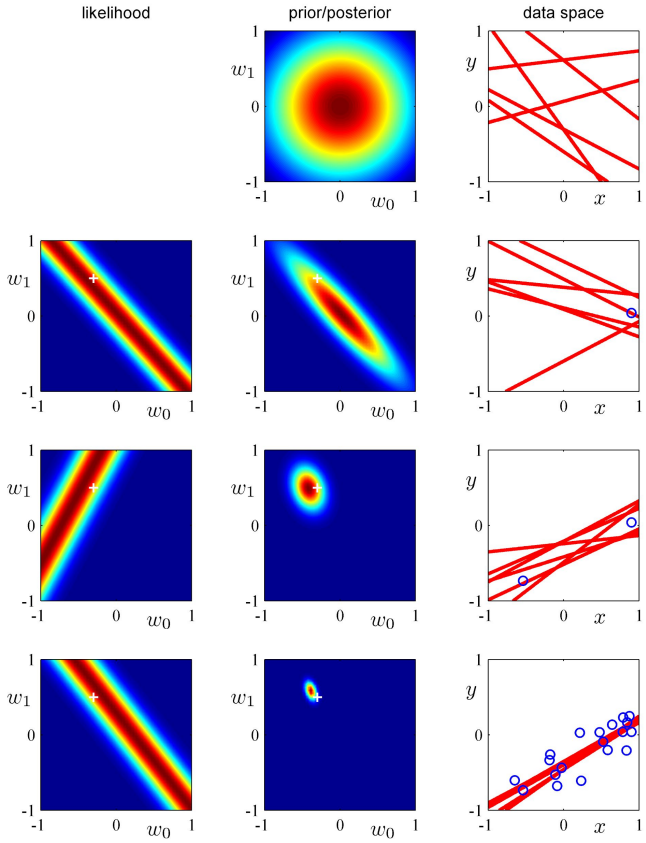
ML, MAP: computes a single prediction $f(\mathbf{x}_*) = \mathbf{x}_*^\top \mathbf{w}^\star$.

Bayesian: computes a predictive distribution,

$$p(y_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(y_* | \mathbf{x}_*, \mathbf{w}) p(\mathbf{w} | \mathbf{X}, \mathbf{y}) \mathrm{d}\mathbf{w}$$

Why Bayesian?

- Reduces overfitting
- Better generalization performance (not always)
- Models predictive uncertainty



Bishop, C., Pattern Recognition and Machine Learning, Chapter 3.

Bayesian linear regression - computing posteriors

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{n} \mathcal{N}(y_i|\mathbf{x}_i^\top \mathbf{w}, \beta^{-1}).$$

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{m}, \mathbf{S}).$$

Exercise: what is a posterior distribution?

Bayesian linear regression - computing posteriors

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = -\frac{\beta}{2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top \mathbf{w})^2 + \text{const.}$$

$$\log p(\mathbf{w}) = -\frac{1}{2}(\mathbf{w} - \mathbf{m})^\top \mathbf{S}^{-1}(\mathbf{w} - \mathbf{m}) + \text{const.}$$

$$\begin{aligned}
&\log p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = \\
&-\frac{1}{2}\mathbf{w}^\top(\beta\mathbf{X}^\top\mathbf{X} + \mathbf{S}^{-1})\mathbf{w} \\
&+ (\beta\mathbf{X}^\top\mathbf{y} + \mathbf{S}^{-1}\mathbf{m})^\top\mathbf{w} + \text{const.}
\end{aligned}$$

Bayesian linear regression - computing posteriors

$$\mathbf{S}_n = (\mathbf{S}^{-1} + \beta \mathbf{X}^\top \mathbf{X})^{-1},$$

$$\mathbf{m}_n = \mathbf{S}_n (\mathbf{S}^{-1} \mathbf{m} + \beta \mathbf{X}^\top \mathbf{y}).$$

$$p(\mathbf{w} | \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mathbf{w} | \mathbf{m}_n, \mathbf{S}_n).$$

A closer look at the posterior

Posterior variance = prior variance + inverse of empirical variance

$$\mathbf{S}_n = (\mathbf{S}^{-1} + \beta \mathbf{X}^\top \mathbf{X})^{-1},$$

Posterior mean = ML solution + prior mean

$$\begin{aligned}\mathbf{m}_n &= \mathbf{S}_n(\mathbf{S}^{-1}\mathbf{m} + \beta \mathbf{X}^\top \mathbf{y}). \\ &= \mathbf{S}_n \mathbf{S}^{-1}\mathbf{m} + \beta \mathbf{S}_n \mathbf{X}^\top \mathbf{y} \\ &= \mathbf{S}_n \mathbf{S}^{-1}\mathbf{m} + \beta(\mathbf{S}^{-1} + \beta \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}\end{aligned}$$

Posterior mean = Ridge regression solution if $\mathbf{m} = 0,\ \mathbf{S} = \gamma \mathbf{I}.$

Useful identities

Conditional Gaussian distributions

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1})$$

$$\Downarrow$$

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\top})$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}(\mathbf{A}^{\top}\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}), \boldsymbol{\Sigma}),$$

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^{\top}\mathbf{L}\mathbf{A})^{-1}$$

Predictions in Bayesian linear regression

Predictive distribution is also a Gaussian.

$$
\begin{aligned}
p(y_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \int p(y_* | \mathbf{w}, \mathbf{x}_*) p(\mathbf{w} | \mathbf{X}, \mathbf{y}) \mathrm{d}\mathbf{w} \\
&= \mathcal{N}(y_* | \mathbf{x}^\top \mathbf{m}_N, \beta^{-1} + \mathbf{x}^\top \mathbf{S}_N \mathbf{x}).
\end{aligned}
$$

Tuning the hyperparameters

Maximize the log marginal likelihood (a.k.a. log evidence) w.r.t. hyperparameters

$$\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\log\frac{|\mathbf{S}_N|}{|\mathbf{S}|} + \frac{N}{2}\log\frac{\beta}{2\pi}$$
$$+ \frac{1}{2}\mathbf{m}_N^\top\mathbf{S}_N^{-1}\mathbf{m}_N - \frac{1}{2}\mathbf{m}^\top\mathbf{S}^{-1}\mathbf{m} - \frac{\beta}{2}\mathbf{y}^\top\mathbf{y}$$

Simple prior

$$\mathbf{m} = \mathbf{0}, \quad \mathbf{S} = \alpha^{-1}\mathbf{I}$$

$$\mathbf{S}_n = (\alpha\mathbf{I} + \beta\mathbf{X}^\top\mathbf{X})^{-1}$$

$$\mathbf{m}_n = \beta\mathbf{S}_n\mathbf{X}^\top\mathbf{y}$$

$$\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\log|\mathbf{S}_n| + \frac{d+1}{2}\log\alpha$$

$$+ \frac{n}{2}\log\frac{\beta}{2\pi} + \frac{1}{2}\mathbf{m}_n^\top\mathbf{S}_n^{-1}\mathbf{m}_n - \frac{\beta}{2}\mathbf{y}^\top\mathbf{y}$$

Coding practice 1
Bayesian linear regression

Handling `numpy` arrays

- Shape of the array: `X.shape`
- Matrix multiplication: `np.dot(X, Y)`
- Matrix transpose: `X.T`
- Elementwise matrix multiplication: `X * Y`
- Matrix inverse: `np.linalg.inv(X)`
- Power (number/vector/matrices…): `X**p`
- Average, sum: `X.sum(), X.mean()`
- Matrix determinant: `np.linalg.det(X)`
- Checkout numpy cheat-sheet

Implementing linear regression

- Open `blr/lr.py`
- Fill in the missing part.
- Answer in `blr/lrr_complete.py`.

Implementing ridge regression

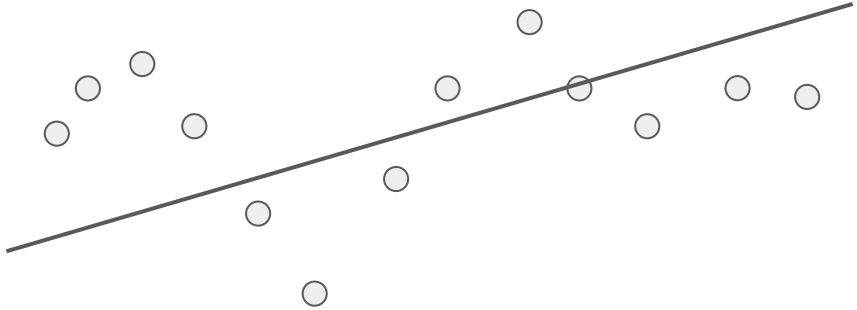
- Open `blr/rlr.py`
- Fill in the missing part.
- Answer in `blr/rlr_complete.py`.

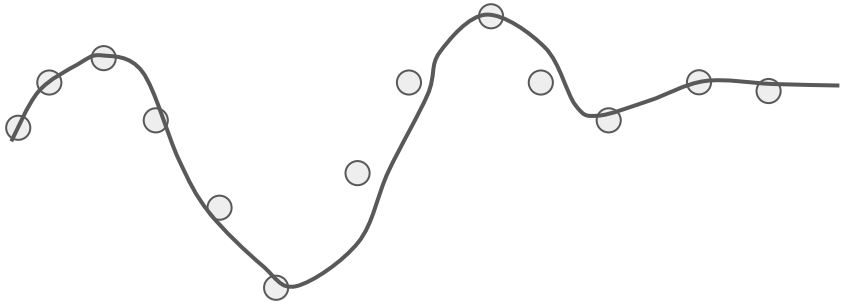
Implementing Bayesian linear regression

- Open `blr/blr.py`
- Fill in the missing part.
- Answer in `blr/blr_complete.py`.

Gaussian process regression

Beyond linear regression

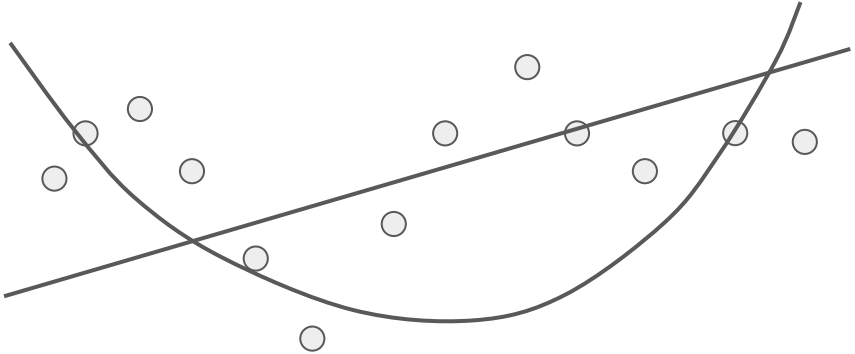




Parametric regression

Finite number of parameters

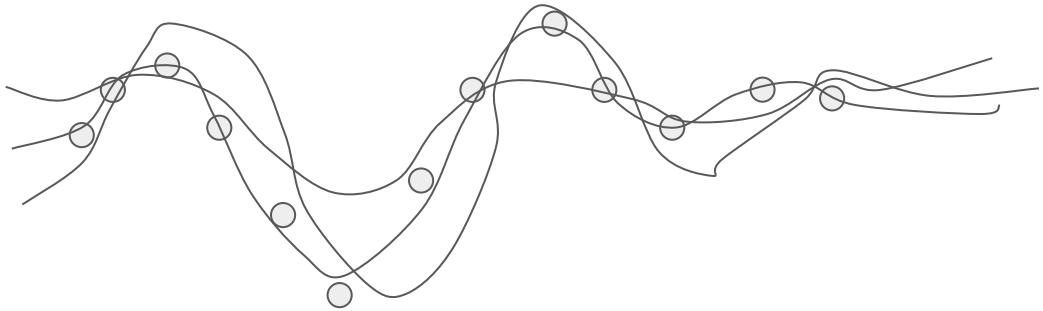
Assumes a parametric form



$$f(x) = w_0 + w_1 x + w_2 x^2 + \ldots w_k x^k.$$

Nonparametric regression

Potentially **infinite number** of parameters - the number of parameters increases as the number of training data (each training data point is a parameter)



Stochastic process and random function.

A function can be interpreted as an infinite-dimensional vector.

$$f = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \end{bmatrix}$$

Stochastic process: a collection of random variables indexed by some covariate.

Stochastic process as a prior distribution for a function (random function)

What defines a random variable?

A random variable is described with its distribution function (or density function)

$$p(x) = \mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(x-\mu)^2}{2\sigma^2} \right).$$

What is a distribution function of a stochastic process? Can we even define it for infinite collection of random variables?

Finite-dimensional distribution

A stochastic process can be defined with a finite-dimensional distribution:

For a stochastic process $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, \}$, a finite-dimensional distribution is the distribution function for finite n :

$$p(f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))$$

Gaussian process

Gaussian process is a stochastic process whose finite-dimensional distribution is Gaussian.

$$(f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots,) \sim \mathrm{GP}(\mu(\cdot), K(\cdot, \cdot))$$

$\mu(\cdot)$ mean function, usually set to zero.

$K(\cdot, \cdot)$ covariance function or kernel

Gaussian process

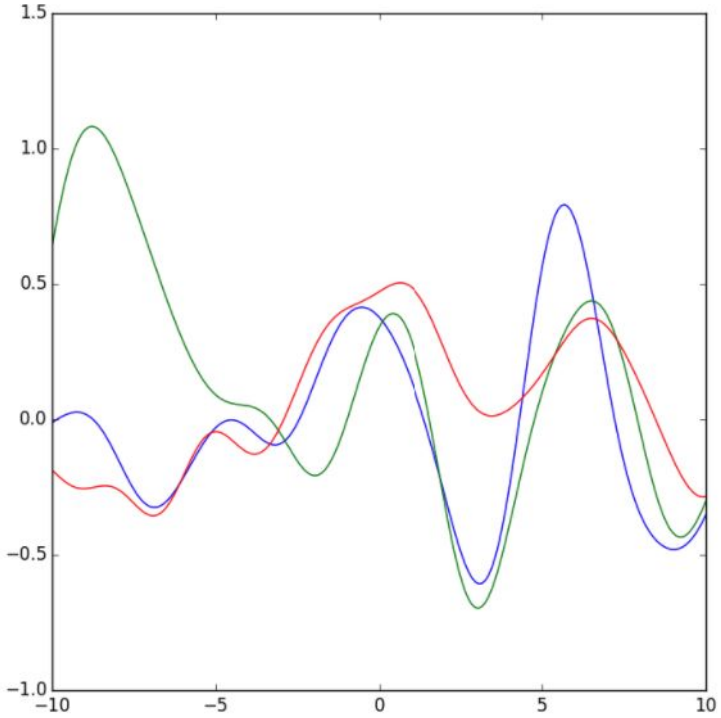
Finite dimensional distributions are Gaussian:

$$p(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)) = \mathcal{N}(f(\mathbf{X})|\mathbf{0}, K(\mathbf{X}, \mathbf{X})),$$

$$\mathbf{f}(\mathbf{X}) = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$$

$$K(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \ldots & K(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_n, \mathbf{x}_1) & \ldots & K(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$

Gaussian process as a random function prior



- Don’t assume any parametric form of the function
- Only assume that the function values changes **smoothly**
- How smooth? - the relationship between function values are described by Gaussian distribution defined with the kernel function

Gaussian process regression

Instead of assuming a parametric regression function with specific parameters, place a random function prior with additive noise!

$$\mathbf{y} = \mathbf{f}(\mathbf{X}) + \boldsymbol{\varepsilon},$$

$$(f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots,) \sim \mathrm{GP}(0, K(\cdot, \cdot))$$

$$\mathbf{f}(\mathbf{X}) = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$$

$$\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \beta^{-1}).$$

Gaussian process regression

Prior distribution: by the finite-dimensional distribution,

$$p(\mathbf{f}(\mathbf{X})|\mathbf{X}) = \mathcal{N}(\mathbf{f}(\mathbf{X})|\mathbf{0}, K(\mathbf{X}, \mathbf{X})).$$

Likelihood: Gaussian as well (additive noise is Gaussian)

$$p(\mathbf{y}|\mathbf{f}(\mathbf{X})) = \prod_{i=1}^{n} \mathcal{N}(y_i|f(\mathbf{x}_i), \beta^{-1}).$$

Prediction for a novel input $\mathbf{x}_*$: compute the posterior

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

Bayesian linear regression vs Gaussian process regression

Bayesian linear regression

- Assume a linear function
- Place a prior on the parameter $\mathbf{w}$
- Prediction using the posterior of the parameter.

Gaussian process regression

- No assumption in function form
- Place a GP prior on function values directly.
- Prediction using the posterior of function values.

Useful identities

Matrix inversion lemma

$$(\mathbf{A} + \mathbf{U}\mathbf{V}^\top)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1}$$

Jointly Gaussian distributions

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \middle| \begin{bmatrix} \mu_\mathbf{x} \\ \mu_\mathbf{y} \end{bmatrix}, \begin{bmatrix} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xy}} \\ \Sigma_{\mathbf{yx}} & \Sigma_{\mathbf{yy}} \end{bmatrix} \right)$$

$$\Downarrow$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\mu_\mathbf{x} + \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}(\mathbf{y} - \mu_\mathbf{y}),$$
$$\Sigma_{\mathbf{xx}} - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}\Sigma_{\mathbf{yx}})$$

Useful identities

Conditional Gaussian distributions

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1})$$

$$\Downarrow$$

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\top})$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}(\mathbf{A}^{\top}\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}), \boldsymbol{\Sigma}),$$

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^{\top}\mathbf{L}\mathbf{A})^{-1}$$

Computing predictive distribution

Decompose the posterior

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(y_*|f(\mathbf{x}_*))p(f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y})\mathrm{d}f(\mathbf{x}_*)$$

$$\int \mathcal{N}(y_*|f(\mathbf{x}_*), \beta^{-1})\boxed{p(f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y})}\mathrm{d}f(\mathbf{x}_*)$$

$$p(f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int \boxed{p(f(\mathbf{x}_*)|f(\mathbf{X}), \mathbf{x}_*, \mathbf{X})}\boxed{p(f(\mathbf{X})|\mathbf{X}, \mathbf{Y})}\mathrm{d}f(\mathbf{X})$$

Computing predictive distribution

$$\boxed{p(f(\mathbf{x}_*)|f(\mathbf{X}), \mathbf{x}_*, \mathbf{X},)} = ?$$

Use the Gaussian conditional distribution identity

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

Computing predictive distribution

$$\boxed{p(f(\mathbf{X})|\mathbf{X},\mathbf{y})} \propto p(\mathbf{y}|f(\mathbf{X}))p(f(\mathbf{X})|\mathbf{X}))$$
$$= \mathcal{N}(\mathbf{y}|f(\mathbf{X}),\beta^{-1}\mathbf{I})\mathcal{N}(f(\mathbf{X})|\mathbf{0},K(\mathbf{X},\mathbf{X}))$$

By Bayes' rule,

$$\propto \mathcal{N}(f(\mathbf{X})|(K^{-1}(\mathbf{X},\mathbf{X})+\beta\mathbf{I})^{-1}\beta\mathbf{y},\,(K^{-1}(\mathbf{X},\mathbf{X})+\beta\mathbf{I})^{-1})$$

Computing predictive distribution

Using the conditional Gaussian distribution identity,

$$p(f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})$$

$$= \mathcal{N}(f(\mathbf{x}_*)|K(\mathbf{x}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I})^{-1}\mathbf{y},$$

$$K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I})^{-1}K(\mathbf{X}, \mathbf{x}_*))$$

Computing predictive distribution

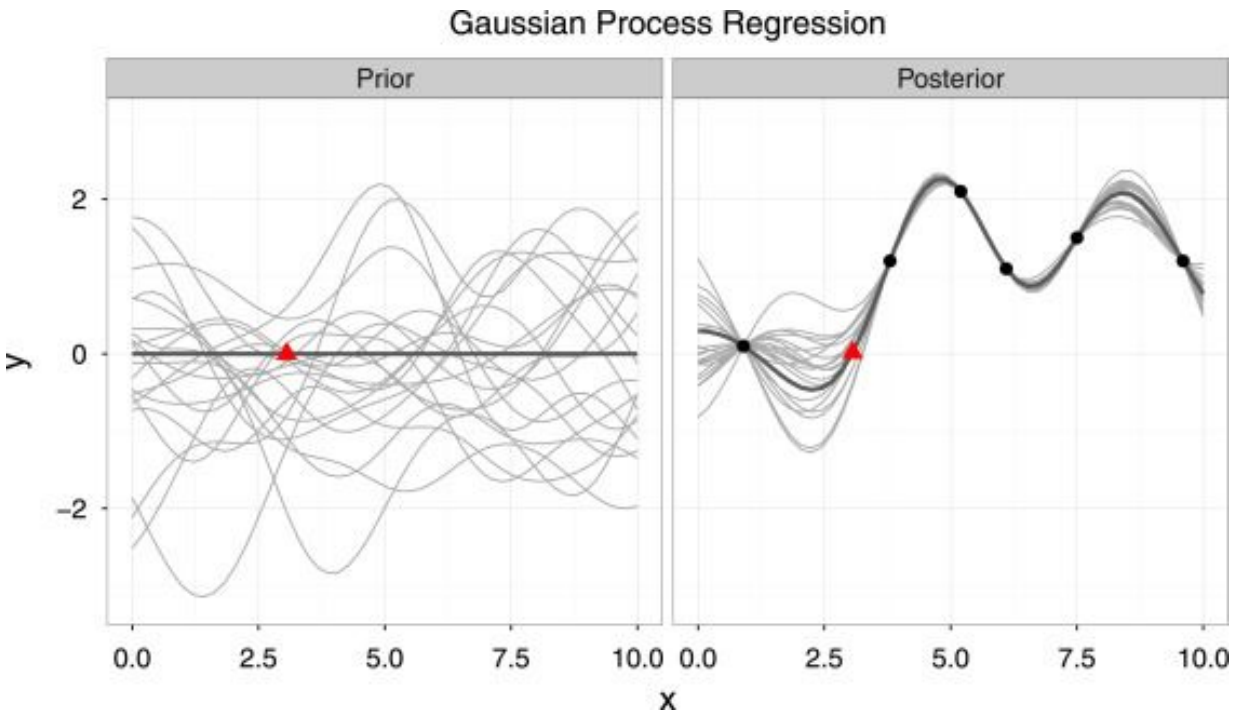
Using the conditional Gaussian distribution identity once again,

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

$$\mathcal{N}(y_*|K(\mathbf{x}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I})^{-1}\mathbf{y},$$

$$K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I})^{-1}K(\mathbf{X}, \mathbf{x}_*) + \beta^{-1}$$

Prior vs Predictive distribution



Typically used kernels

RBF kernel, Squared exponential kernel, Gaussian Kernel, ...

$$K(\mathbf{x}, \mathbf{y}) = c^2 \exp\left(- \frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2} \right)$$

Matern kernel

$$K(\mathbf{x}, \mathbf{y}) = c^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} d/\sigma)^\nu K_\nu(\sqrt{2\nu} d/\sigma), \quad (d = \|\mathbf{x} - \mathbf{y}\|)$$

Periodic kernel

$$K(\mathbf{x}, \mathbf{y}) = c^2 \exp\left(- \frac{2 \sin^2(\pi \|\mathbf{x} - \mathbf{y}\| / p)}{\sigma^2} \right)$$

Training GPR model

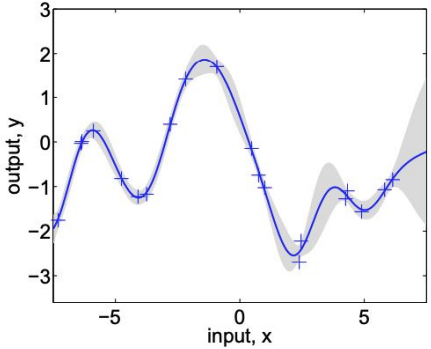
What parameters should be tuned?

- kernel hyperparameters α (for RBF Kernel $\alpha = (c, \sigma)$)
- observation noise β

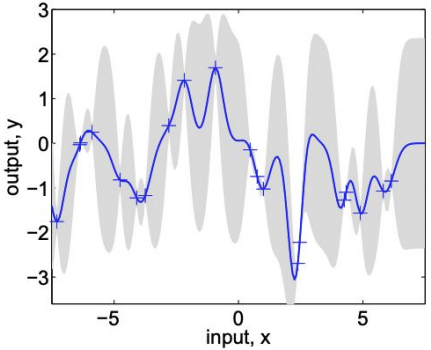
How? Cross validation? : maximizing marginal likelihood $p(\mathbf{y}|\mathbf{X})$ by gradient descent.

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I}).$$

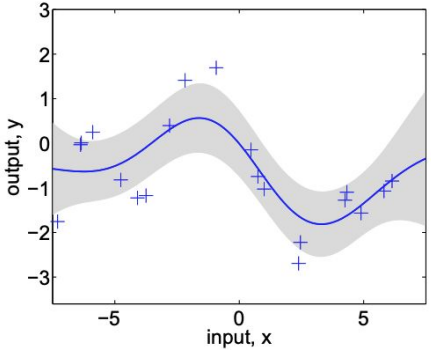
Importance of kernel hyperparameters



(a), $\ell = 1$



(b), $\ell = 0.3$



(c), $\ell = 3$

Gaussian process regression - practical issue

Matrix inversion can be costly: $O(n^3)$

$$\mathcal{N}(y_* | K(\mathbf{x}_*, \mathbf{X}) \boxed{(K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I})^{-1}} \mathbf{y},$$
$$K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{X}) \boxed{(K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I})^{-1}} K(\mathbf{X}, \mathbf{x}_*) + \beta^{-1}$$

Approximation: Nystrom approximation, sparse Gaussian process (stochastic gradient descent)

Coding practice 2
Gaussian process regression

Implementing Gaussian process regression

- Fill in the missing part in `gpr/gpr.py`.
- Answer in `gpr/gpr_complete.py`.
- Run `gpr/gpr_sklearn.py` to compare.