Chapter 4: Bayesian Machine Learning

Maschinelles Lernen 1 -
Grundverfahren WS21/22

Prof. Gerhard Neumann

KIT, Institut für Anthrophomatik und Robotik

Learning Outcomes

What will we learn today?

- Understand the “Bayesian formulation” of machine learning
- What are the 2 basics steps needed for Bayesian learning
- What are the advantages of being “Bayesian”?
- For which representations can Bayesian learning be done in closed form?
- How to compute the posterior and predictive distribution for Bayesian Linear Regression?
- How to compute the posterior and predictive distribution for Gaussian Processes?

Today‘s Agenda!

Bayesian Learning:

- Posterior and Predictive Distribution
- Bayesian estimation for Gaussians
- Maximum A-posteriori (MAP) Estimates

Bayesian Regression Algorithms:

- Bayesian Linear Regression
- Gaussian Processes
- General Models and Approximations

Basics:

Gaussian Identities:

- Completing the Square
- Gaussian Bayes Rules
- Gaussian Propagation

Bayesian Learning

So far:

- We mainly considered single models, i.e., a point estimate $\boldsymbol{\theta}^*$ for the parameter vector

However, …

1. As the data is noisy, the estimated optimal parameter vector $\boldsymbol{\theta}^*$ is also uncertain
   - I.e. parameters are just random variables
   - We so far do not really know how wrong / uncertain our estimate $\boldsymbol{\theta}^*$ is
2. We have also seen that multiple models (ensembles, see trees + forests) usually work better!

Motivation of Bayesian Learning:

- Estimate uncertainty in $\boldsymbol{\theta}^*$
- Find a more robust predictor by averaging over (infinitely) many predictors
- … where each predictor is weighted by the probability of being “right”
- Use this estimate to quantify uncertainty of the prediction

1-step: Compute Posterior

Compute the probability of “being right” for a parameter $\boldsymbol{\theta}$ using Bayes theorem:

$$\underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} = \frac{\overbrace{p(\mathcal{D}|\boldsymbol{\theta})}^{\text{data likelihood}} \quad \overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\mathcal{D})}_{\text{evidence}}}$$

- Prior: Can encode our subjective belief
- Posterior: Probability of parameter vector given the data
- Likelihood: Specified by our parametric model $\mathcal{D}$
- Evidence: Normalization, can be used for model comparison (later)

2-step: Compute predictive distribution

Predicting of a new data-point $\boldsymbol{x}^*$:

$$\underbrace{p(\boldsymbol{x}^*|\mathcal{D})}_{\text{marginal likelihood}} = \int \underbrace{p(\boldsymbol{x}^*|\boldsymbol{\theta})}_{\text{likelihood}} \underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} d\boldsymbol{\theta}$$

- Parameter vector $\boldsymbol{\theta}$ is integrated out
- Likelihood $p(\boldsymbol{x}^*|\mathcal{D})$ is now purely determined by the data $\mathcal{D}$
- $p(\boldsymbol{x}^*|\mathcal{D})$ is often called marginal likelihood as $\boldsymbol{\theta}$ is marginalized out

Intuition: If you assign each parameter estimator a “probability of being right”, the average of these parameter estimators will be better than the single one

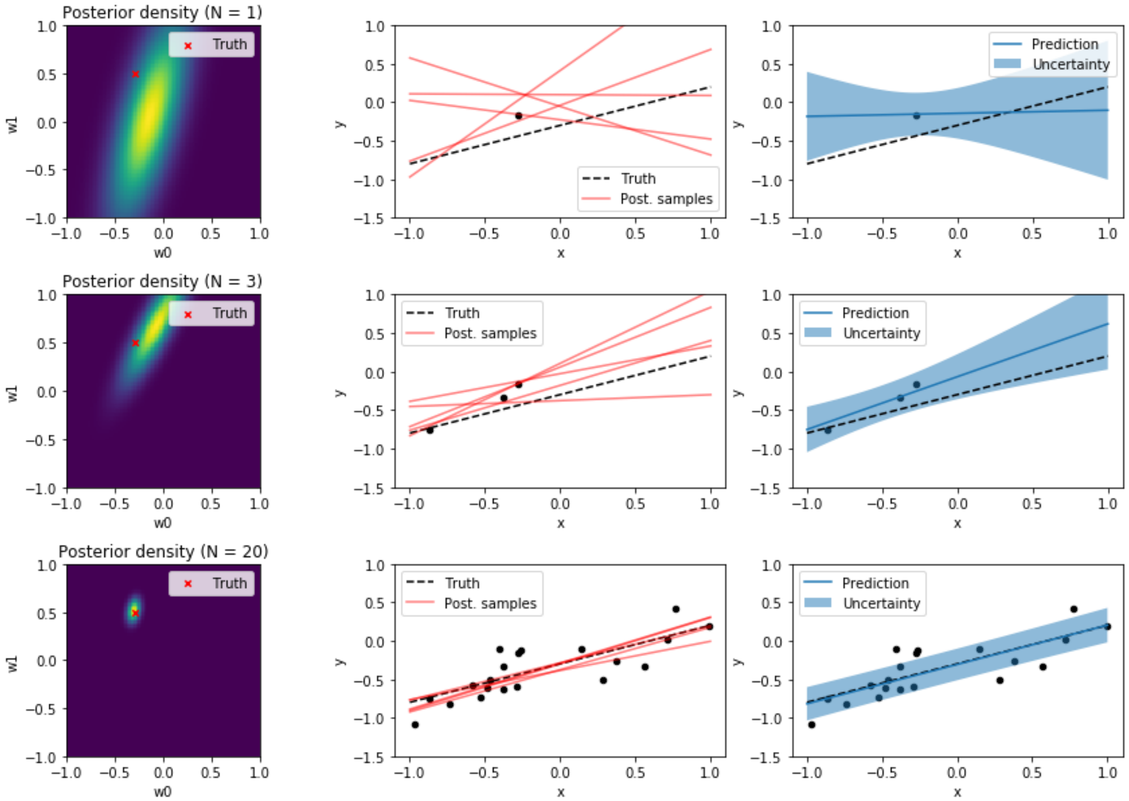
- Weighted ensemble method (with potentially infinite amount of models)
- … often, samples from $p(\boldsymbol{\theta}|\mathcal{D})$ are used to approximate the integral (finite number of models in this case)

Example:

Bayesian Linear Regression
(math comes later…)

Observation:

- The posterior becomes more narrow with more data



$p(\boldsymbol{\theta}|\mathcal{D})$

Samples from posterior

Posterior integrated out

Picture taken from http://krasserm.github.io/2019/02/23/bayesian-linear-regression/

Priors

Prior $p(\boldsymbol{\theta})$ should capture our belief and domain knowledge as well as possible

What is our domain knowledge for a general ML algorithm?

- For most ML algorithms, we know that the weights $\boldsymbol{\theta}$ should be small
- This knowledge can be expressed with a Gaussian prior, e.g.

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \lambda^{-1}\boldsymbol{I})$$

  - Most common for weight vectors (linear regression, neural nets…)
  - λ is the precision of the prior
- However, many other priors are possible

Example: Gaussian Distribution

Likelihood (sample): $p(x|\boldsymbol{\theta} = \{\mu, \sigma\}) = \mathcal{N}(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$

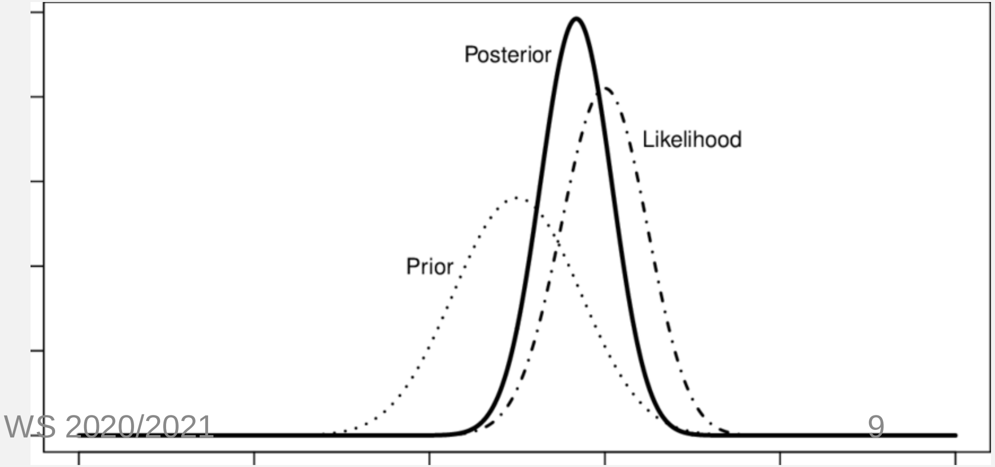
Likelihood (dataset): $p(\boldsymbol{X}|\mu, \sigma) = \prod_i p(x_i|\mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{\sum_i (x_i-\mu)^2}{2\sigma^2}\right\}$

Prior: $p(\mu) = \mathcal{N}(\mu|\mu_0, \sigma_0) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left\{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right\}$

Compute posterior $p(\mu|\boldsymbol{X})$ for μ assuming σ is known:

$$p(\mu|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\mu)p(\mu)}{p(\boldsymbol{X})}$$

$$\propto p(\boldsymbol{X}|\mu)p(\mu)$$



Basics: Completing the square

Posterior: $p(\mu|\boldsymbol{X}) \propto p(\boldsymbol{X}|\mu)p(\mu) \propto \exp\left\{-\frac{\sum_i (x_i-\mu)^2}{2\sigma^2} - \frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right\}$

Completing the square: Bring exponent in canonical squared form, i.e.

$$\exp\Big(-\underbrace{\frac{1}{2}a\mu^2}_{\text{squared term}} + \underbrace{b\mu}_{\text{linear term}} + \text{const}\Big)$$

Then we know that $p(\mu|\boldsymbol{X}) = \mathcal{N}(\mu|\mu_N, \sigma_N^2)$ with:

- Mean: $\mu_N = a^{-1}b$
- Variance: $\sigma_N^2 = a^{-1}$

Basics: Completing the square

Posterior: $$p(\mu|\boldsymbol{X}) \propto p(\boldsymbol{X}|\mu)p(\mu) \propto \exp\left\{-\frac{\sum_i (x_i-\mu)^2}{2\sigma^2} - \frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right\}$$

$$= \exp\left\{-\frac{1}{2}\underbrace{\left(\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}\right)}_{a}\mu^2 + \underbrace{\left(\frac{\sum_i x_i}{\sigma^2}+\frac{\mu_0}{\sigma_0^2}\right)}_{b}\mu + \text{const}\right\}$$

Completing the square: $p(\mu|\boldsymbol{X}) = \mathcal{N}(\mu|\mu_N, \sigma_N^2)$

- Mean: $$\mu_N = a^{-1}b = \frac{\sigma_0^2}{N\sigma_0^2+\sigma^2}\sum_i x_i + \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0$$
  $$= \frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\mu_{\text{ML}} + \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0$$
- Variance: $$\sigma_N^2 = a^{-1} = \frac{\sigma^2\sigma_0^2}{N\sigma_0^2+\sigma^2}$$

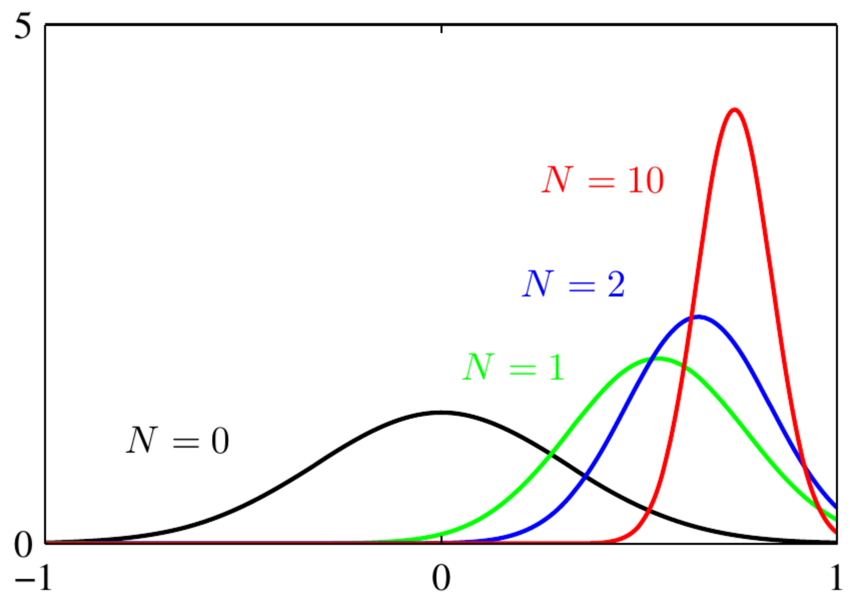
Example: Posterior Distribution

The posterior is Gaussian with:

- Mean: $$\mu_N = \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{\mathrm{ML}} + \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0$$
- Variance: $$\sigma_N^2 = \frac{\sigma^2\sigma_0^2}{N\sigma_0^2 + \sigma^2}$$

Observations:

- Variance decreases with more training samples
- Will eventually reach 0
- Posterior mean interpolates between prior mean and sample average



Example: Computing the predictive distribution

The predictive distribution is given by:

$$\underbrace{p(x^*|\boldsymbol{X})}_{\text{marginal likelihood}} = \int \underbrace{p(x^*|\mu)}_{\text{likelihood}} \underbrace{p(\mu|\boldsymbol{X})}_{\text{posterior}} d\mu$$

$$= \int \mathcal{N}(x^*|\mu,\sigma)\mathcal{N}(\mu|\mu_N,\sigma_N)d\mu \ \dots \ \text{Gaussian propagation (proof not shown)}$$

$$= \mathcal{N}(x^*|\mu_{x^*},\sigma^2_{x^*})$$

The predictive distribution is Gaussian with:

- Mean: $\mu_{x^*} = \mu_N$
- Variance: $\sigma^2_{x^*} = \sigma^2_N + \sigma^2$

Observations:

- The predictive mean is the same as the posterior mean
- However, the predictive variance also considers the uncertainty of the mean

Exercise: Multivariate Gaussian Distribution

Likelihood (sample):

$$p(\boldsymbol{x}|\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left\{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right\}$$

Likelihood (dataset):

$$p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_i p(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{N/2}} \exp\left\{-\frac{1}{2}\sum_i (\boldsymbol{x}_i-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})\right\}$$

Prior: $p(\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$

Exercise: What is the posterior distribution $p(\boldsymbol{\mu}|\boldsymbol{X})$?

Conjugate priors

If the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ is in the same probability distribution family as the prior probability distribution $p(\boldsymbol{\theta})$, the prior and posterior are then called conjugate distributions, and the prior is called a conjugate prior for the likelihood function.

- In our example, the prior and posterior are Gaussian
- I.e. the Gaussian distribution is conjugate to itself

Other conjugate prior distributions:

- Gamma distribution is conjugate for the variance of a scalar Gaussian
- Wishart distribution is conjugate for the covariance of a multivariate Gaussian

… won’t be covered in the lecture but good to know it exists.

Summary: Bayesian Learning

1. Compute posterior: $$\underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} = \frac{\overbrace{p(\mathcal{D}|\boldsymbol{\theta})}^{\text{data likelihood}} \; \overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\mathcal{D})}_{\text{evidence}}}$$

1. Integrate out posterior: $$\underbrace{p(\boldsymbol{x}^*|\mathcal{D})}_{\text{marginal likelihood}} = \int \underbrace{p(\boldsymbol{x}^*|\boldsymbol{\theta})}_{\text{likelihood}} \; \underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} \, d\boldsymbol{\theta}$$

Properties:

- For very large datasets, the posterior will be a point estimate $\lim_{n\rightarrow\infty} p(\boldsymbol{\theta}|\mathcal{D}_n) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$
  - I.e., Bayesian Learning will be equivalent to maximum likelihood
- **However, large advantage for smaller datasets!**
  1. We know where our model is uncertain
  2. More robust estimate due to averaging

Summary: Bayesian Learning

1. Compute posterior:

$$\underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} = \frac{\overbrace{p(\mathcal{D}|\boldsymbol{\theta})}^{\text{data likelihood}} \quad \overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\mathcal{D})}_{\text{evidence}}}$$

1. Integrate out posterior:

$$\underbrace{p(\boldsymbol{x}^*|\mathcal{D})}_{\text{marginal likelihood}} = \int \underbrace{p(\boldsymbol{x}^*|\boldsymbol{\theta})}_{\text{likelihood}} \; \underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} \, d\boldsymbol{\theta}$$

In most cases, both operations can not be performed analytically

- **Exception:** Bayesian Linear Regression + Gaussian Processes (coming soon)
- Very high-dimensional integrals, hard to compute
- **Simplification:** Maximum A-posteriori (MAP) Solution
- **Various Approximations:** Laplace Approximation, Variational Inference, Sampling, etc… (not covered)

Maximum a-posteriori solution

Simplification of Bayesian Learning:

1. Find the parameter vector $\boldsymbol{\theta}_{\mathrm{MAP}}$ that maximizes the posterior

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{D}) = \arg\max_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

   - Uncertainty in $\boldsymbol{\theta}$ is ignored
   - Optimization is done in log-domain

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})$$

2. Use $\boldsymbol{\theta}_{\mathrm{MAP}}$ for prediction

$$p(\boldsymbol{x}^*|\mathcal{D}) \approx p(\boldsymbol{x}^*|\boldsymbol{\theta}_{\mathrm{MAP}})$$

Maximum a-posteriori solution

MAP solution: $$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} \left(\log p(\mathcal{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \right)$$

- Prior has similar role than a regularization loss

Example: Regression

- Gaussian likelihood $p(\mathcal{D}|\boldsymbol{\theta}) = p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta}) = \prod_i \mathcal{N}(y_i|f_{\boldsymbol{\theta}}(\boldsymbol{x}_i),\sigma^2)$
- Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{0},\lambda^{-1}\boldsymbol{I})$
- Corresponding objective: $$\arg\max_{\boldsymbol{\theta}} \underbrace{\sum_i -\frac{1}{2\sigma^2}\left(y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right)^2}_{\text{Sum of squared errors}} - \underbrace{\frac{\lambda}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}}_{-\lambda/2\|\boldsymbol{\theta}\|^2} + \underbrace{c(\sigma^2,\lambda)}_{\text{only interested in } \boldsymbol{\theta}}$$
  - Gaussian prior corresponds to a L2 regularization loss!
  - Gaussian likelihood corresponds to squared loss!

Example: MAP for Linear Regression

Remember: In linear regression $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$

Objective:

$$\boldsymbol{w}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{w}} \underbrace{\sum_i -\frac{1}{2\sigma^2}\left(y_i - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i)\right)^2}_{\text{Sum of squared errors}} \underbrace{- \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}}_{-\lambda/2\|\boldsymbol{\theta}\|} + \underbrace{c(\sigma^2, \lambda)}_{\text{only interested in } \boldsymbol{w}}$$

$$= \arg\min_{\boldsymbol{w}} \sum_i \left(y_i - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i)\right)^2 + \lambda\sigma^2\boldsymbol{w}^T\boldsymbol{w} + c(\sigma^2, \lambda)$$

- The MAP objective for Linear Regression is equivalent to Ridge Regression!
  - with $\lambda_{\mathrm{ridge}} = \lambda\sigma^2$

Example: MAP for Linear Regression

Objective:

$$\boldsymbol{w}_{\mathrm{MAP}} = \arg\min_{\boldsymbol{w}} \sum_i \left(y_i - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i)\right)^2 + \lambda\sigma^2 \boldsymbol{w}^T \boldsymbol{w}$$

Result:

$$\boldsymbol{w}_{\mathrm{MAP}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\sigma^2\boldsymbol{I})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}$$

- We have now 2 parameters:
  - λ … sets the importance of the prior
  - σ^2 … uncertainty of the training data

Can be set by optimizing log likelihood + log prior via cross validation

Q: Why do we need cross validation here?

Predictive Model:

$$p(y^*|\boldsymbol{x}^*, \mathcal{D}) \approx p(y^*|\boldsymbol{x}^*, \boldsymbol{w}_{\mathrm{MAP}}) = \mathcal{N}(y^*|\boldsymbol{w}_{\mathrm{MAP}}^T\boldsymbol{\phi}(\boldsymbol{x}^*), \sigma^2)$$

- Uncertainty solely depends on estimated noise level σ^2
  - I.e. noise is input independent

Intermediate Wrap-up Bayesian Learning

- Treat parameter vector as random variable and estimate posterior
  - Estimate probability of “being right” for $\boldsymbol{\theta}$
- Posterior distribution is integrated out for prediction
  - All possible parameter vectors are used for the prediction
  - Weighted by probability of “being right”
- Posterior quantifies our uncertainty in the model
  - Can also be used to quantify uncertainty in the prediction

We will now look at 2 examples for Bayesian Learning:

- Bayesian Linear Regression
- Gaussian Processes

Today‘s Agenda!

Bayesian Learning:

- Posterior and Predictive Distribution
- Bayesian estimation for Gaussians
- Maximum A-posteriori (MAP) Estimates

Bayesian Regression Algorithms:

- Bayesian Linear Regression
- Gaussian Processes
- General Models and Approximations

Bayesian Model Selection

Basics:

Gaussian Identities:

- Completing the Square
- Gaussian Bayes Rules
- Gaussian Propagation

Bayesian Linear Regression

For Bayesian Linear Regression, the posterior and the prediction can be computed in closed form:

- For all other cases, we need approximations
- While linear models are limited, it is still insightful to look at the properties of this case

Model:

Linear model

Noise variance

- Likelihood (single sample): $p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(y|\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}), \sigma^2)$

Feature Matrix

- Likelihood (full dataset): $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) = \prod_i \mathcal{N}(y_i|\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i), \sigma^2) = \underbrace{\mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{w}, \sigma^2\boldsymbol{I})}_{\text{Multivariate distribution}}$
  - Write product of independent Gaussians as multivariate Gaussian

- Gaussian prior: $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\boldsymbol{I})$

Parameter precision

Bayesian Linear Regression

2 Steps:

1. **Compute posterior**

$$p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{y}|\boldsymbol{X})} = \frac{\overbrace{p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})}^{\text{Likelihood}}\overbrace{p(\boldsymbol{w})}^{\text{Prior}}}{\underbrace{\int p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}}_{\text{Evidence/Normalizer}}}$$

2. **Compute predictive distribution:** Integrate posterior out

$$p(y^*|\boldsymbol{x}^*,\boldsymbol{X},\boldsymbol{y}) = \int \overbrace{p(y^*|\boldsymbol{w},\boldsymbol{x}^*)}^{\text{Parameter-specific prediction}}\underbrace{p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y})}_{\text{Posterior}}d\boldsymbol{w}$$

We have to look at some basics first…

Basics: Gaussian Identities

- Eq (1): Joint from Marginal and Conditional:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})\mathcal{N}(\boldsymbol{y}|\boldsymbol{F}\boldsymbol{x}, \boldsymbol{\Sigma}_{\boldsymbol{y}}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{x}} \\ \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{x}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{x}} & \boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T \\ \boldsymbol{F}\boldsymbol{\Sigma}_{\boldsymbol{x}} & \boldsymbol{\Sigma}_{\boldsymbol{y}} + \boldsymbol{F}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T \end{bmatrix}\right)$$

- Eq (2): Marginal and Conditional Gaussian from Joint:

$$\mathcal{N}\left(\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{x}} \\ \boldsymbol{\mu}_{\boldsymbol{y}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{x}} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{\Sigma}_{\boldsymbol{y}} \end{bmatrix}\right) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})\mathcal{N}\left(\boldsymbol{y}|\boldsymbol{\mu}_{\boldsymbol{y}} + \boldsymbol{C}^T\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}), \boldsymbol{\Sigma}_{\boldsymbol{y}} - \boldsymbol{C}^T\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\boldsymbol{C}\right).$$

- Can also be derived by “completing the square”…

Basics: Gaussian Bayes rule

Bayes rule for Gaussian distribution:

- Marginal: $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$
- Conditional: $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{F}\boldsymbol{x}, \sigma_{\boldsymbol{y}}^2\boldsymbol{I})$

- **Gaussian Bayes Rule 1:** The posterior $p(\boldsymbol{x}|\boldsymbol{y})$ is Gaussian with
  - Mean: $\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T(\sigma_{\boldsymbol{y}}^2\boldsymbol{I} + \boldsymbol{F}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T)^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{x}})$
  - Covariance: $\boldsymbol{\Sigma}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\Sigma}_{\boldsymbol{x}} - \boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T(\sigma_{\boldsymbol{y}}^2\boldsymbol{I} + \boldsymbol{F}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T)^{-1}\boldsymbol{F}\boldsymbol{\Sigma}_{\boldsymbol{x}}$
- **Derivation:**
  - Use Eq (1) to form joint $p(\boldsymbol{x}, \boldsymbol{y})$ from marginal $p(\boldsymbol{x})$ and conditional $p(\boldsymbol{y}|\boldsymbol{x})$
  - Use Eq (2) to form posterior $p(\boldsymbol{x}|\boldsymbol{y})$ from joint $p(\boldsymbol{x}, \boldsymbol{y})$

Basics: Gaussian Bayes rule

Bayes rule for Gaussian distribution:

- Marginal: $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$
- Conditional: $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{F}\boldsymbol{x}, \sigma_{\boldsymbol{y}}^2 \boldsymbol{I})$

- **Gaussian Bayes Rule 2:** The posterior $p(\boldsymbol{x}|\boldsymbol{y})$ is Gaussian with
  - Mean: $\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\mu}_{\boldsymbol{x}} + (\sigma_{\boldsymbol{y}}^2 \boldsymbol{I} + \boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{F}^T \boldsymbol{F})^{-1} \boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{F}^T (\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{x}})$
  - Covariance: $\boldsymbol{\Sigma}_{\boldsymbol{x}|\boldsymbol{y}} = (\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} + \sigma_{\boldsymbol{y}}^{-2} \boldsymbol{F}^T \boldsymbol{F})^{-1} = \sigma_{\boldsymbol{y}}^2 (\sigma_{\boldsymbol{y}}^2 \boldsymbol{I} + \boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{F}^T \boldsymbol{F})^{-1} \boldsymbol{\Sigma}_{\boldsymbol{x}}$
- **Derivation: Use following identities for Bayes rule 1 from the matrix cookbook…**
  - Use identity $\boldsymbol{A}(\boldsymbol{I} + \boldsymbol{B}\boldsymbol{A})^{-1} = (\boldsymbol{I} + \boldsymbol{A}\boldsymbol{B})^{-1}\boldsymbol{A}$ for the mean (Searl identity)
  - Use identity $(\boldsymbol{A} + \boldsymbol{C}\boldsymbol{B}\boldsymbol{C}^T)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{C}(\boldsymbol{B}^{-1} + \boldsymbol{C}^T\boldsymbol{A}^{-1}\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{A}^{-1}$ for the covariance (Woodbury identity)

Basics: Gaussian Bayes rule

- **Gaussian Bayes Rule 1**
  - Mean: $\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T(\sigma_{\boldsymbol{y}}^2\boldsymbol{I} + \boldsymbol{F}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T)^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{x}})$
  - Covariance: $\boldsymbol{\Sigma}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\Sigma}_{\boldsymbol{x}} - \boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T(\sigma_{\boldsymbol{y}}^2\boldsymbol{I} + \boldsymbol{F}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T)^{-1}\boldsymbol{F}\boldsymbol{\Sigma}_{\boldsymbol{x}}$
- **Gaussian Bayes Rule 2:**
  - Mean: $\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\mu}_{\boldsymbol{x}} + (\sigma_{\boldsymbol{y}}^2\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} + \boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{x}})$
  - Covariance: $\boldsymbol{\Sigma}_{\boldsymbol{x}|\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^2(\sigma_{\boldsymbol{y}}^2\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} + \boldsymbol{F}^T\boldsymbol{F})^{-1}$

Observations: Both rules are mathematically equivalent…

- However, numerically it can be a huge difference
- Bayes rule 1: Invert a matrix with dimension *dim(y) x dim(y)*
- Bayes rule 2: Invert a matrix with dimension *dim(x) x dim(x)*

Use Rule 1 if dim(y) < dim(x),
otherwise Rule 2

Basics: Gaussian propagation

Given marginal $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ and conditional $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{F}\boldsymbol{x}, \sigma_{\boldsymbol{y}}^2\boldsymbol{I})$ we want to obtain the marginal for y

- The marginal distribution $p(\boldsymbol{y}) = \int p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})d\boldsymbol{x}$ is Gaussian with
  - Mean: $\boldsymbol{\mu}_{\boldsymbol{y}} = \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{x}}$
  - Variance: $\boldsymbol{\Sigma}_{\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^2\boldsymbol{I} + \boldsymbol{F}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{F}^T$
- Derivation:
  - Use Eq (1) to obtain joint distribution
  - Marginal p(***y***) can be directly read from the joint
- Variance in ***y*** increases due to uncertainty in ***x***

Ok… now we are ready to derive Bayesian Linear Regression!

Computing the Posterior

- Likelihood (conditional): $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{w}, \sigma^2\boldsymbol{I})$
- Prior (marginal): $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\boldsymbol{I})$
- Dimensions: $\boldsymbol{w} \in \mathbb{R}^d$ and $\boldsymbol{Y} \in \mathbb{R}^{N\times 1}$, typically $d \ll N$

As the dimensionality of the marginal variable (parameter vector) is smaller than dimensionality of cond. variable (number of datapoints), we have to use Gaussian Bayes rule 2!

$$\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\mu}_{\boldsymbol{x}} + (\sigma_{\boldsymbol{y}}^2\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} + \boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\mu}_{\boldsymbol{x}})$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}|\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^2(\sigma_{\boldsymbol{y}}^2\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} + \boldsymbol{F}^T\boldsymbol{F})^{-1}$$

- with $\boldsymbol{\mu}_{\boldsymbol{x}} = \boldsymbol{0}$, $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \lambda^{-1}\boldsymbol{I}$, $\boldsymbol{F} = \boldsymbol{\Phi}$ and $\sigma_{\boldsymbol{y}}^2 = \sigma_{\boldsymbol{y}}^2$

Computing the Posterior

Posterior $p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}})$ **:**

- Posterior mean: $\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \sigma_{\boldsymbol{y}}^2\lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}$
- Posterior covariance: $\boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^2(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \sigma_{\boldsymbol{y}}^2\lambda\boldsymbol{I})^{-1}$

Observations:

- The posterior mean is equivalent to the MAP estimate
- … results from the linearity of the likelihood (not the case for non-linear models)

So whats the advantage?

- We also get an uncertainty estimate for the parameter vector!

Example: Samples from the posterior

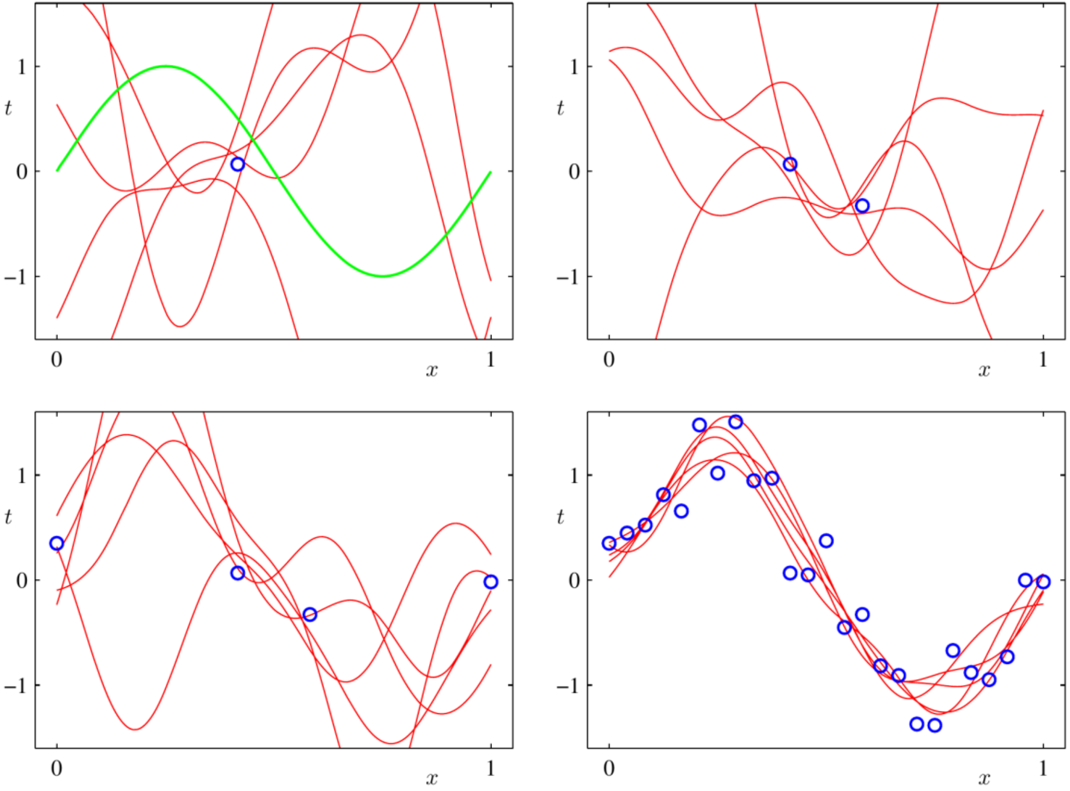
- We can create samples

$$\boldsymbol{w}_i \sim p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y})$$

- Each $\boldsymbol{w}_i$ represents a function

$$f_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{\phi}(\boldsymbol{x})$$

- Basis functions are given by RBF basis functions



Predictive Distribution

The predictive distribution is given by:

$$p(y^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{y}) = \int p(y^*|\boldsymbol{w}, \boldsymbol{x}^*) p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}) d\boldsymbol{w}$$

$$= \int \mathcal{N}(y_*|\boldsymbol{\phi}_*^T \boldsymbol{w}, \sigma_{\boldsymbol{y}}^2) \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}}) d\boldsymbol{w}$$

- Using Gaussian propagation, we can evaluate the predictive distribution. It is Gaussian with
  - Mean: $\mu(\boldsymbol{x}^*) = \boldsymbol{\phi}(\boldsymbol{x}^*)^T (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \sigma_{\boldsymbol{y}}^2 \boldsymbol{I})^{-1} \boldsymbol{\Phi}^T \boldsymbol{y}$
  - Variance: $\sigma^2(\boldsymbol{x}^*) = \sigma_{\boldsymbol{y}}^2 \left(1 + \boldsymbol{\phi}(\boldsymbol{x}^*)^T (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \sigma_{\boldsymbol{y}}^2 \boldsymbol{I})^{-1} \boldsymbol{\phi}(\boldsymbol{x}^*)\right)$
- Nothing new for the mean (same as Ridge Regression / MAP solution)
- **However**: The variance is now input dependent!

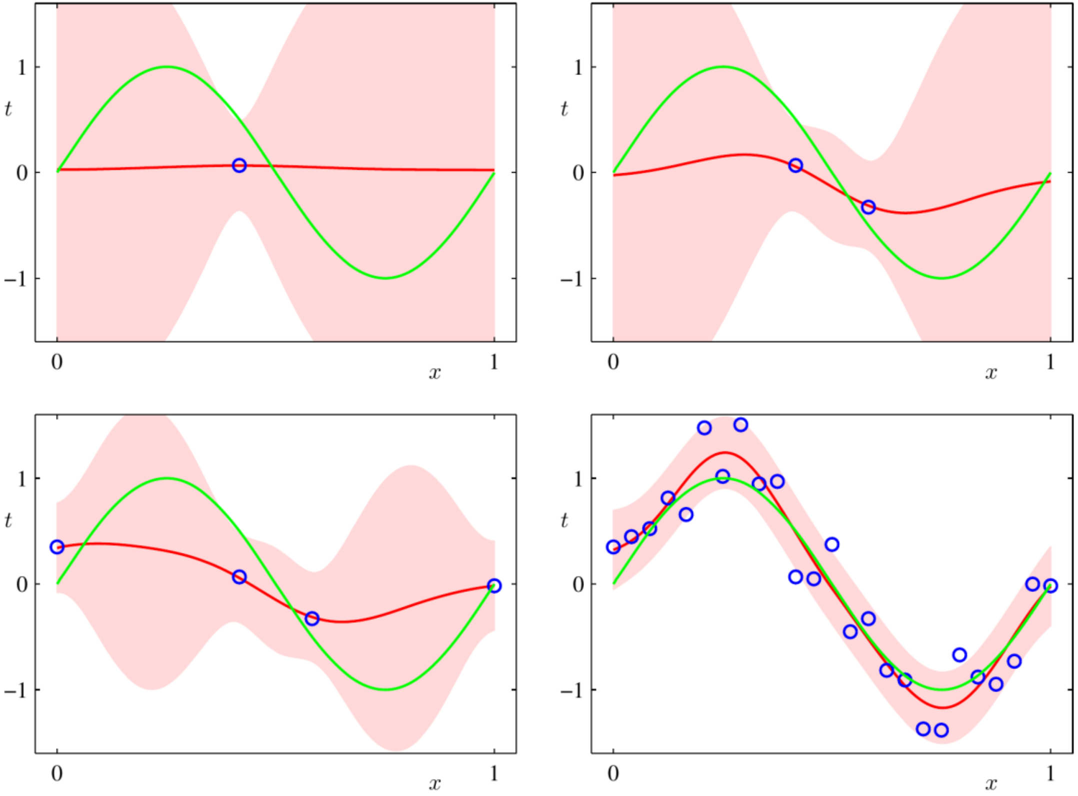
Example: Predictive Distribution

Visualize predictive distribution with

$$\mu(\boldsymbol{x}^*) \pm 2\sigma(\boldsymbol{x}^*)$$

- Model uncertainty is reduced if training data contains information about weight for specific feature

- In the limit $N \rightarrow \infty$ model uncertainty vanishes and only noise variance $\sigma_{\boldsymbol{y}}^2$ remains



Today‘s Agenda!

Bayesian Learning:

- Posterior and Predictive Distribution
- Bayesian estimation for Gaussians
- Maximum A-posteriori (MAP) Estimates

Bayesian Regression Algorithms:

- Bayesian Linear Regression
- Gaussian Processes
- General Models and Approximations

Basics:

Gaussian Identities:

- Completing the Square
- Gaussian Bayes Rules
- Gaussian Propagation

Gaussian Processes

A Gaussian Process (GP) $f(\boldsymbol{x}) \sim \mathcal{GP}(\underbrace{m(\boldsymbol{x})}_{\text{mean function}}, \underbrace{k(\boldsymbol{x}, \boldsymbol{x}')}_{\text{covariance function}})$

is a probability distribution over functions *f(**x**)*, such that any finite set of function values $t_i = f(\boldsymbol{x}_i)$ evaluated at inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ is jointly Gaussian distributed

- Mean function evaluates our prior belief about the function

  $$\mathbb{E}[f(\boldsymbol{x})] = m(\boldsymbol{x})$$

  – For simplicity, we will use $m(\boldsymbol{x}) = 0$

- Covariance function evaluates how similar/correlated two function evaluations at inputs $\boldsymbol{x}, \boldsymbol{x}'$ are

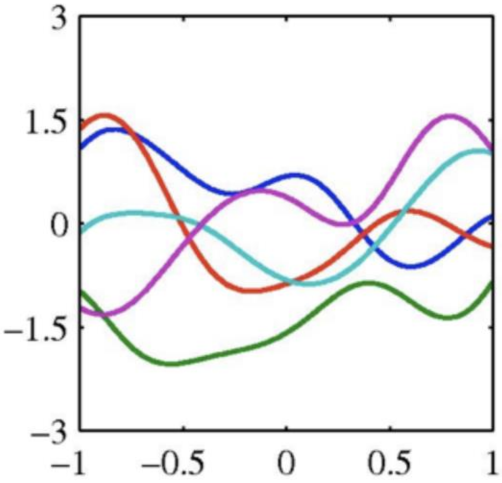
  $$\mathbb{E}[f(\boldsymbol{x})f(\boldsymbol{x}')] = k(\boldsymbol{x}, \boldsymbol{x}')$$

  – Covariance function needs to be a positive definite function (similar to a kernel function)

Different covariance functions

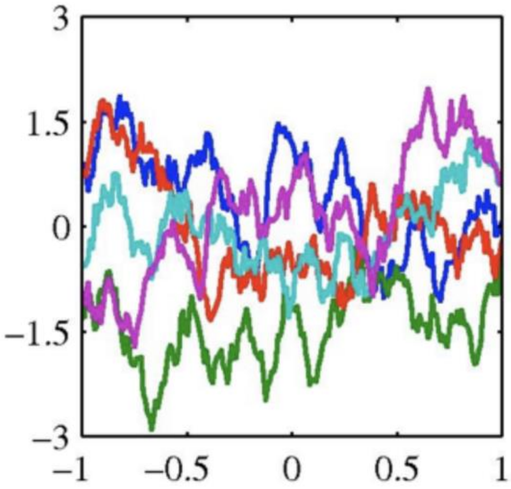
Samples from a GP prior with different covariance functions

- The covariance encodes our prior belief in the smoothness of the function



$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$$

Gaussian Kernel



$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\theta\left\|\mathbf{x}_i - \mathbf{x}_j\right\|\right)$$

Ornstein-Uhlenbeck Process (Brownian Motion)

Gaussian Processes

I.e. a Gaussian process over N function evaluations $\boldsymbol{t} = [t_1, \dots, t_N]^T$ is completely specified by the 2nd order statistics, i.e., mean and covariance, i.e.

$$p(\boldsymbol{t}|\boldsymbol{X}) = \mathcal{N}\left(\boldsymbol{t}|\boldsymbol{0}, \boldsymbol{K}\right) \quad \text{with} \quad \boldsymbol{K} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}$$

In reality, we can only measure noisy function values, i.e. $y_i = f(\boldsymbol{x}_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_y^2)$. We get the following Gaussian distribution over $\boldsymbol{y}$

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X}) &= \int p(\boldsymbol{y}|\boldsymbol{t}) p(\boldsymbol{t}|\boldsymbol{X}) d\boldsymbol{t} \\ &= \int \mathcal{N}(\boldsymbol{y}|\boldsymbol{t}, \sigma_y^2 \boldsymbol{I}) \mathcal{N}(\boldsymbol{t}|\boldsymbol{0}, \boldsymbol{K}) d\boldsymbol{t} \dots \text{ Gaussian propagation} \\ &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{K} + \sigma_y^2 \boldsymbol{I}) \end{aligned}$$

Predictive distribution

We know that the function values $\boldsymbol{y}$ for the training set $\boldsymbol{X}$ and for a new data point $\boldsymbol{x}^*$ are jointly Gaussian distributed. Hence, also the conditional $p(y^*|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x}^*)$ is also Gaussian distributed.

- For a new data-point y^* we can obtain the joint distribution over function values

$$p\left(\left[\begin{array}{c}\boldsymbol{y}\\ y^*\end{array}\right] \middle| \left[\begin{array}{c}\boldsymbol{X}\\ \boldsymbol{x}^*\end{array}\right]\right) = \mathcal{N}\left(\left[\begin{array}{c}\boldsymbol{y}\\ y^*\end{array}\right] \middle| \left[\begin{array}{cc}\boldsymbol{K}+\sigma_y^2\boldsymbol{I} & \boldsymbol{k}_{\boldsymbol{x}^*}\\ \boldsymbol{k}_{\boldsymbol{x}^*}^T & k^*+\sigma_y^2\end{array}\right]\right)$$

$$\boldsymbol{K} \ldots \text{ kernel matrix, } \boldsymbol{k}_{\boldsymbol{x}^*} = [k(\boldsymbol{x}_1, \boldsymbol{x}^*), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x}^*)]^T \ldots \text{ kernel vector , } k^* = k(\boldsymbol{x}^*, \boldsymbol{x}^*)$$

- We can condition on $\boldsymbol{y}$ to obtain $p(y^*|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x}^*)$ using the Gaussian identities (Eq. 2). The predictive distribution is Gaussian with
  - Mean: $\mu(\boldsymbol{x}^*) = \boldsymbol{k}_{\boldsymbol{x}^*}^T(\boldsymbol{K}+\sigma_y^2\boldsymbol{I})^{-1}\boldsymbol{y}$
  - Variance: $\sigma(\boldsymbol{x}^*) = k^* + \sigma_y - \boldsymbol{k}(\boldsymbol{x}^*)^T(\boldsymbol{K}+\sigma_y^2\boldsymbol{I})^{-1}\boldsymbol{k}(\boldsymbol{x}^*)$

Predictive distribution

Predictive GP distribution:

- Mean:

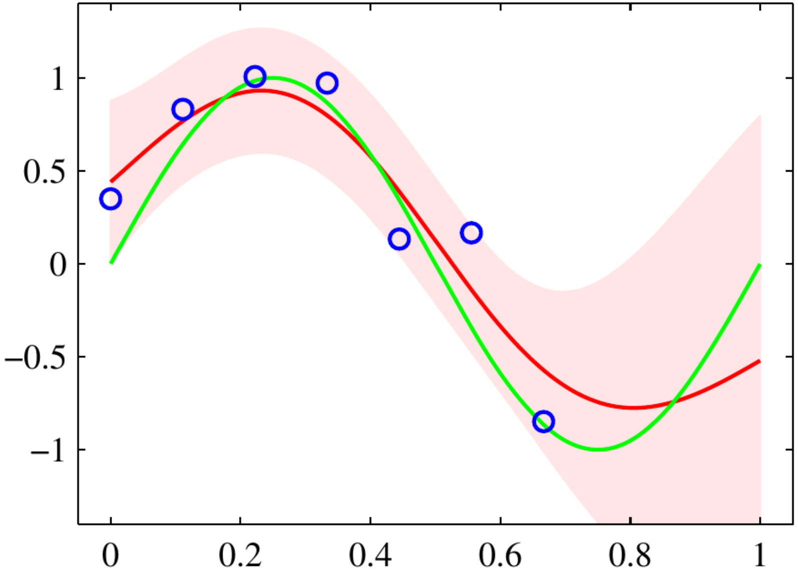
$$\mu(\boldsymbol{x}^*) = \boldsymbol{k}_{\boldsymbol{x}^*}^T(\boldsymbol{K} + \sigma_y^2\boldsymbol{I})^{-1}\boldsymbol{y}$$

- Variance:

$$\sigma^2(\boldsymbol{x}^*) = k^* + \sigma_y^2 - \boldsymbol{k}_{\boldsymbol{x}^*}^T(\boldsymbol{K} + \sigma_y^2\boldsymbol{I})^{-1}\boldsymbol{k}_{\boldsymbol{x}^*}$$

Observations:

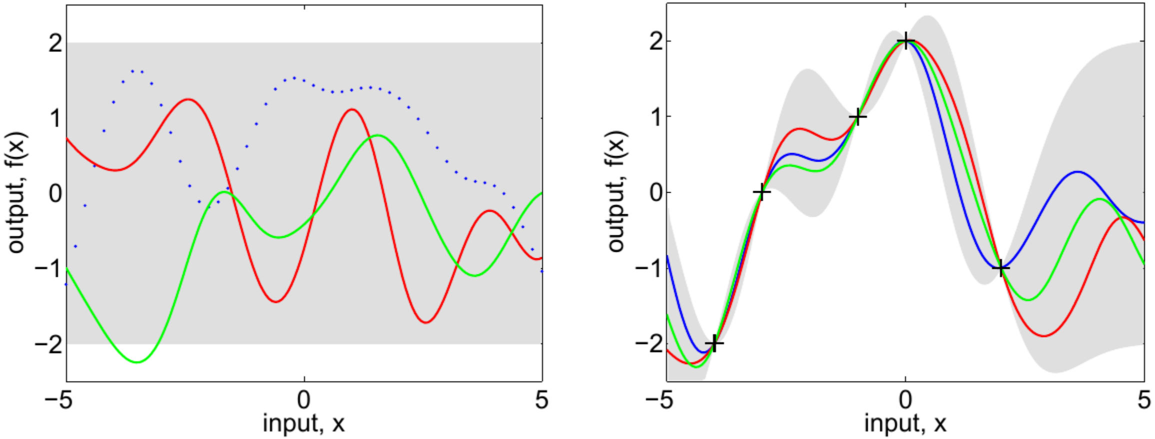
- The mean corresponds to the Kernel Ridge Regression solution
- Yet, we also get an input dependent variance estimate
- Variance is reduced if kernel activations are high



Example of Sinusoidal Data Set (green: true function; blue: noisy data; red: GPR predictive mean; shaded: ±2σ)

Illustration of Posterior

Samples of the prior and the posterior (after conditioning on y)



(a), prior

(b), posterior

Weight space view

So why are GPs an instance of Bayesian Learning?

- We can also derive GPs from the Bayesian Linear Regression view
- Kernelized version of Bayesian Linear Regression (with infinite dimensional feature spaces)

So back to Bayesian linear regression….

- Likelihood (conditional): $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{w}, \sigma_y^2 \boldsymbol{I})$
- Prior (marginal): $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \lambda^{-1}\boldsymbol{I})$
- Dimensions: $\boldsymbol{w} \in \mathbb{R}^d$ and $\boldsymbol{Y} \in \mathbb{R}^{N \times 1}$, $\text{high/infinite dimensional features } d \gg N$

As the dimensionality of the marginal variable (parameter vector) is now larger than dimensionality of cond. Variable (number of samples), we have to use Gaussian Bayes rule 1

Recap: Gaussian Bayes rule

- **Gaussian Bayes Rule 1**
  - Mean: $\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{F}^T (\sigma_{\boldsymbol{y}}^2 \boldsymbol{I} + \boldsymbol{F} \boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{F}^T)^{-1} (\boldsymbol{y} - \boldsymbol{F} \boldsymbol{\mu}_{\boldsymbol{x}})$
  - Covariance: $\boldsymbol{\Sigma}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\Sigma}_{\boldsymbol{x}} - \boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{F}^T (\sigma_{\boldsymbol{y}}^2 \boldsymbol{I} + \boldsymbol{F} \boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{F}^T)^{-1} \boldsymbol{F} \boldsymbol{\Sigma}_{\boldsymbol{x}}$
- **Gaussian Bayes Rule 2:**
  - Mean: $\boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{\mu}_{\boldsymbol{x}} + (\sigma_{\boldsymbol{y}}^2 \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} + \boldsymbol{F}^T \boldsymbol{F})^{-1} \boldsymbol{F}^T (\boldsymbol{y} - \boldsymbol{F} \boldsymbol{\mu}_{\boldsymbol{x}})$
  - Covariance: $\boldsymbol{\Sigma}_{\boldsymbol{x}|\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^2 (\sigma_{\boldsymbol{y}}^2 \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} + \boldsymbol{F}^T \boldsymbol{F})^{-1}$

Observations: Both rules are mathematically equivalent…

- However, numerically it can be a huge difference
- Bayes rule 1: Invert a matrix with dimension *dim(y) x dim(y)*
- Bayes rule 2: Invert a matrix with dimension *dim(x) x dim(x)*

Use Rule 1 if dim(y) < dim(x),
otherwise Rule 2

Recap: A few kernel identities

A kernel is an inner product of a feature space: $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x}')$

Let $\boldsymbol{\Phi}_X = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)^T \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}_N)^T \end{bmatrix} \in \mathbb{R}^{N \times d}$ then the following identities hold:

- **Kernel matrix:** $\boldsymbol{K} = \boldsymbol{\Phi}_X \boldsymbol{\Phi}_X^T$
  - Check: $[\boldsymbol{K}]_{ij} = \boldsymbol{\phi}(\boldsymbol{x}_i)^T \boldsymbol{\phi}(\boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- **Kernel vector:** $\boldsymbol{k}(\boldsymbol{x}^*) = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}^*) \\ \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}^*) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)^T \boldsymbol{\phi}(x^*) \\ \vdots \\ \phi(\boldsymbol{x}_N)^T \boldsymbol{\phi}(x^*) \end{bmatrix} = \boldsymbol{\Phi}_X \boldsymbol{\phi}(\boldsymbol{x}^*)$

Recap: A few kernel identities

A kernel is an inner product of a feature space: $k(\boldsymbol{x}, \boldsymbol{x}') = \lambda^{-1} \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle = \lambda^{-1} \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x}')$

Let $\boldsymbol{\Phi}_X = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)^T \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}_N)^T \end{bmatrix} \in \mathbb{R}^{N \times d}$ then the following identities hold:

In the Bayesian case, we will subsume the prior precision λ into the kernel

- **Kernel matrix:** $\boldsymbol{K} = \lambda^{-1} \boldsymbol{\Phi}_X \boldsymbol{\Phi}_X^T$
  - Check: $[\boldsymbol{K}]_{ij} = \lambda^{-1} \boldsymbol{\phi}(\boldsymbol{x}_i)^T \boldsymbol{\phi}(\boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- **Kernel vector:** $\boldsymbol{k}(\boldsymbol{x}^*) = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}^*) \\ \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}^*) \end{bmatrix} = \lambda^{-1} \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)^T \boldsymbol{\phi}(x^*) \\ \vdots \\ \phi(\boldsymbol{x}_N)^T \boldsymbol{\phi}(x^*) \end{bmatrix} = \lambda^{-1} \boldsymbol{\Phi}_X \boldsymbol{\phi}(\boldsymbol{x}^*)$

Computing the Posterior

Using the Gaussian Bayes Rule 1 results in the following posterior:

$$\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}} = \lambda^{-1}\boldsymbol{\Phi}^T(\sigma_{\boldsymbol{y}}^2\boldsymbol{I} + \underbrace{\lambda^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^T}_{\boldsymbol{K}})^{-1}\boldsymbol{y}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}} = \lambda^{-1}\boldsymbol{I} - \lambda^{-2}\boldsymbol{\Phi}^T\underbrace{(\sigma_{\boldsymbol{y}}^2\boldsymbol{I} + \boldsymbol{K})}_{N\times N \text{ matrix}}^{-1}\boldsymbol{\Phi}$$

- We used the Kernel trick to evaluate the inverse matrix
  - The prior precision λ has been subsumed in the kernel
- Both quantities are still potentially infinite dimensional and can not be evaluated!

Predictive distribution

Still, we can use the posterior to evaluate the predictive distribution (again using the Kernel trick)

$$
\begin{aligned}
p(y^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{y}) &= \int p(y^*|\boldsymbol{w}, \boldsymbol{x}^*) p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}) d\boldsymbol{w} \\
&= \int \mathcal{N}(y_*|\boldsymbol{\phi}_*^T \boldsymbol{w}, \sigma_{\boldsymbol{y}}^2) \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}}) d\boldsymbol{w}
\end{aligned}
$$

The predictive distribution is again Gaussian with

- Mean:

$$
\begin{aligned}
\mu(\boldsymbol{x}^*) &= \lambda^{-1} \boldsymbol{\phi}(\boldsymbol{x}^*)^T \boldsymbol{\Phi}^T (\sigma_{\boldsymbol{y}}^2 \boldsymbol{I} + \boldsymbol{K})^{-1} \boldsymbol{y} \\
&= \boldsymbol{k}(\boldsymbol{x}^*)^T (\sigma_{\boldsymbol{y}}^2 \boldsymbol{I} + \boldsymbol{K})^{-1} \boldsymbol{y}
\end{aligned}
$$

- Variance:

$$
\begin{aligned}
\sigma(\boldsymbol{x}^*) &= \sigma_y^2 + \lambda^{-1} \boldsymbol{\phi}(\boldsymbol{x}^*)^T \boldsymbol{\phi}(\boldsymbol{x}^*) - \lambda^{-2} \boldsymbol{\phi}(\boldsymbol{x}^*)^T \boldsymbol{\Phi}^T (\sigma_{\boldsymbol{y}}^2 \boldsymbol{I} + \boldsymbol{K})^{-1} \boldsymbol{\Phi} \boldsymbol{\phi}(\boldsymbol{x}^*) \\
&= \sigma_y^2 + k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}(\boldsymbol{x}^*)^T (\sigma_{\boldsymbol{y}}^2 \boldsymbol{I} + \boldsymbol{K})^{-1} \boldsymbol{k}(\boldsymbol{x}^*)
\end{aligned}
$$

… which is the same result as obtained with Gaussian conditioning

Wrap-up: GP derivations

Function View:

- A Gaussian process is a distribution over functions, where every set of N function evaluations is jointly Gaussian distributed
- Predictions can hence be performed by conditioning

Weight Space View:

- A Gaussian process is a Bayesian Kernel Regression approach
- Underlying feature space is potentially infinite dimensional
- Weight vector (which is not representable) is integrated out using the Kernel trick

While GP for Regression is computationally very expensive ($O(N^3)$), it is one of the most principled approaches to statistical learning for regression

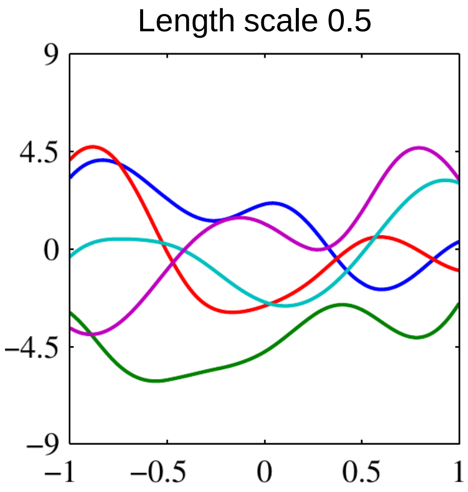
Kernels and Hyperparameters

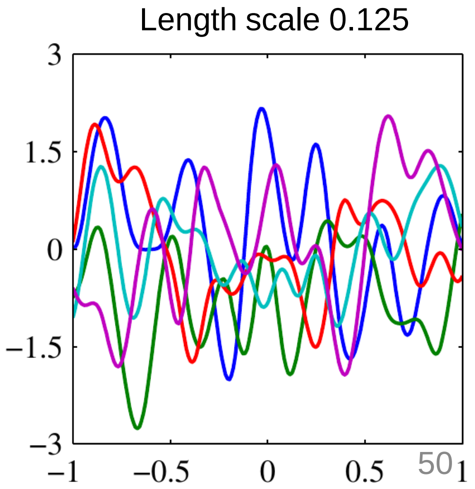
- The parameters of the kernel (e.g. length-scale of Gaussian kernel) are called “hyper-parameters” β .
- The prior precision of the weights as well as the observation noise are for simplicity also subsumed in the kernel hyper-parameters

The most common kernel is the Gaussian / RBF / squared-exponential Kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \lambda^{-1} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{2l^2}\right) + \delta_{ij}\sigma_y^2$$

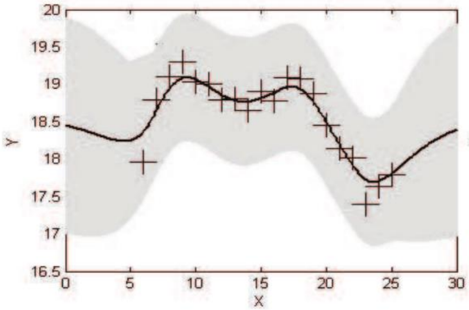
- $\lambda \dots$ prior precision of the weight vector
- $\sigma_y^2 \dots$ noise variance
  - (only applied if i = j, in this notation $\boldsymbol{K} + \sigma_y^2 \boldsymbol{I}$ is replaced by $\boldsymbol{C}$
- $l \dots$ length scale



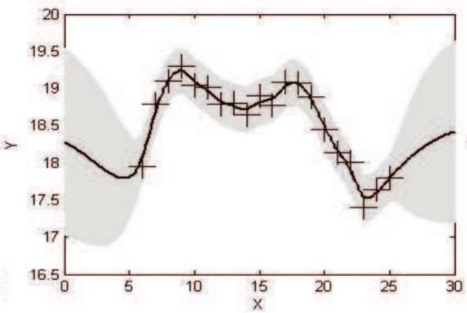


Influence of the Hyper-Parameters

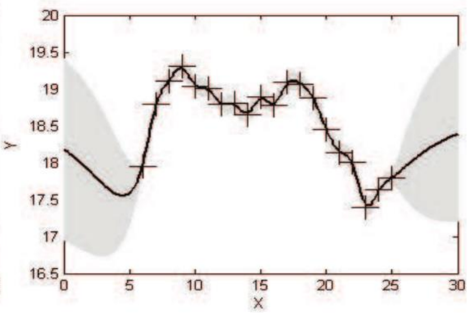
Different noise levels



Too big / Underfitting

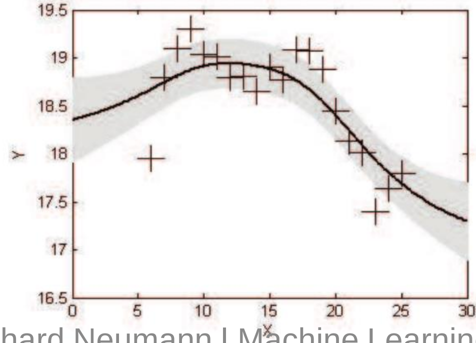


About right

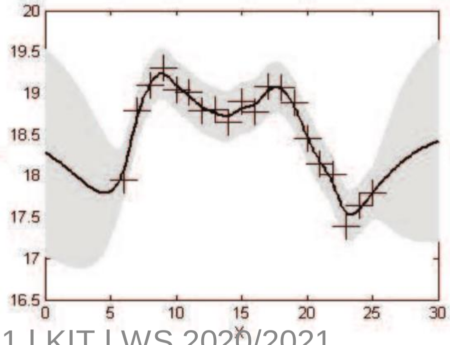


Too small / Overfitting

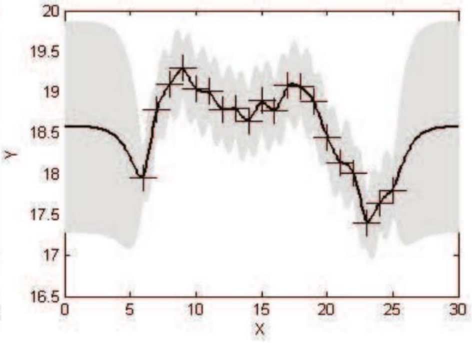
Different length scales



Too big / Underfitting



About right



Too small / Overfitting

Kernels and Hyperparameters

Squared-exponential Kernel can be extended with a length-scale per dimension

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \lambda^{-1} \exp\left(- \sum_{k=1}^{d} \frac{(x_{i,k} - x_{j,k})^2}{2 l_k^2} \right) + \delta_{ij} \sigma_y^2$$

- $\lambda \ldots$ prior precision of the weight vector
- $\sigma^2 \ldots$ noise variance (only applied if i = j)
- $l_k \ldots$ length scale for dimension k

Also called Automatic Relevance Determination (ARD) kernel:

- Optimizing the length-scale determines the relevance of each dimension
- Large length-scale -> dimension is less important

Optimization of the Hyperparameters

- In GPs, the parameters $\boldsymbol{w}$ can be integrated out in closed form
- Yet, no closed form solution exists for the hyper-parameters

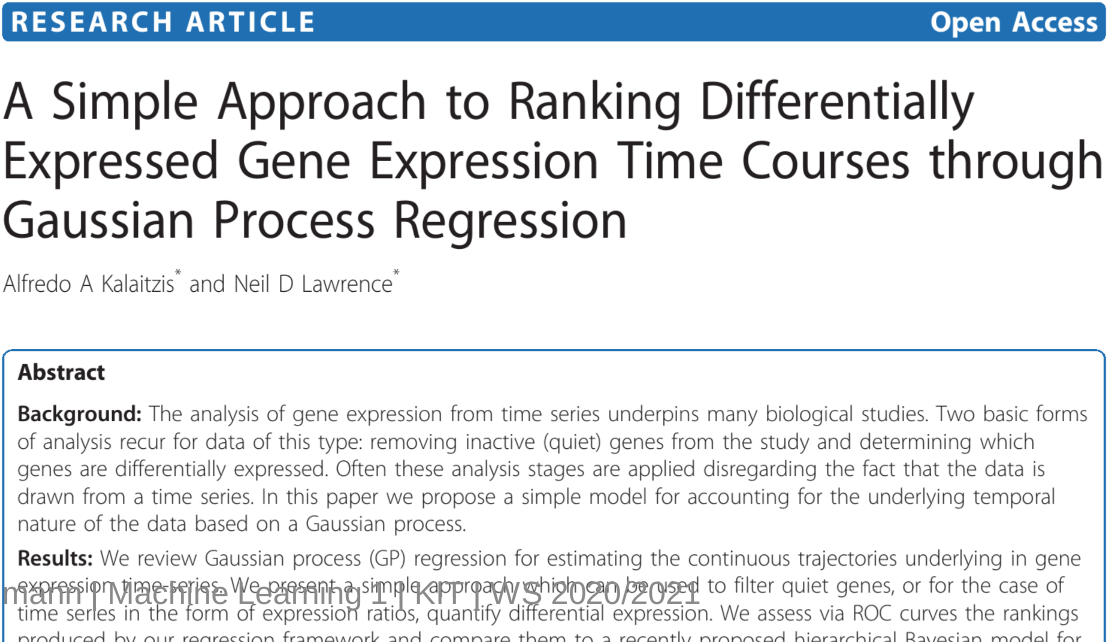
Objective: Log-likelihood of the training data

$$\begin{aligned}\boldsymbol{\beta}^* &= \arg\max_{\boldsymbol{\beta}} \log \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\beta}}) \\ &= \arg\max_{\boldsymbol{\beta}} -\frac{1}{2}\log|\boldsymbol{C}_{\boldsymbol{\beta}}| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{C}_{\boldsymbol{\beta}}^{-1}\boldsymbol{y} - \frac{N}{2}\log(2\pi)\end{aligned}$$

- Need to be optimized via gradient descent (only batch)
- Non-convex, multiple optima
- Only a small number of hyper-parameters
- Very flexible representation, beware of overfitting (would be better to do that on validation set)!

Example: Gene Expression

- Given gene expression levels in the form of a time series
- Want to detect if a gene is expressed or not, fit a GP to each gene [Kalaitzis and Lawrence, 2011]



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*

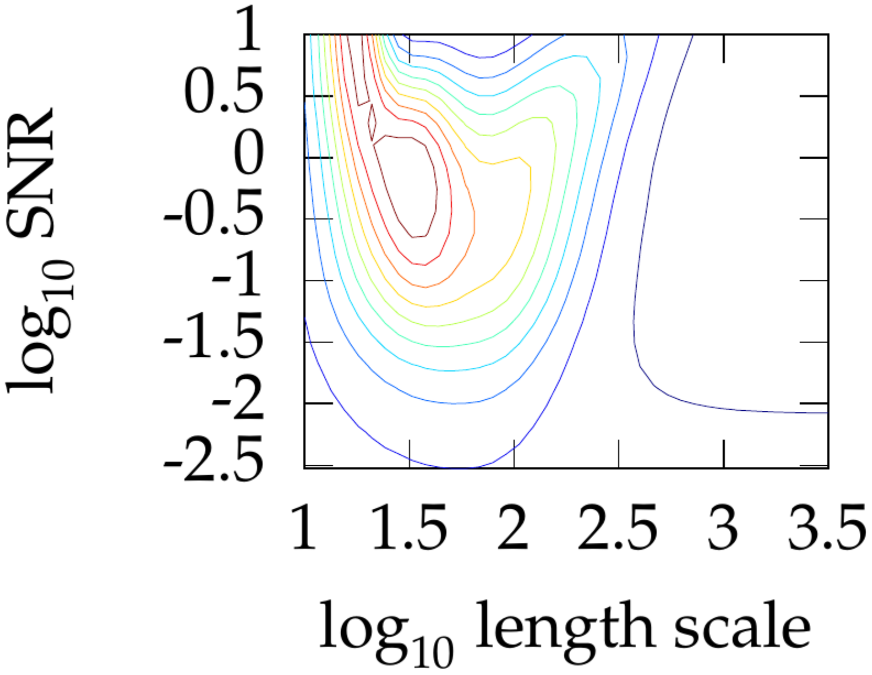
Abstract

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

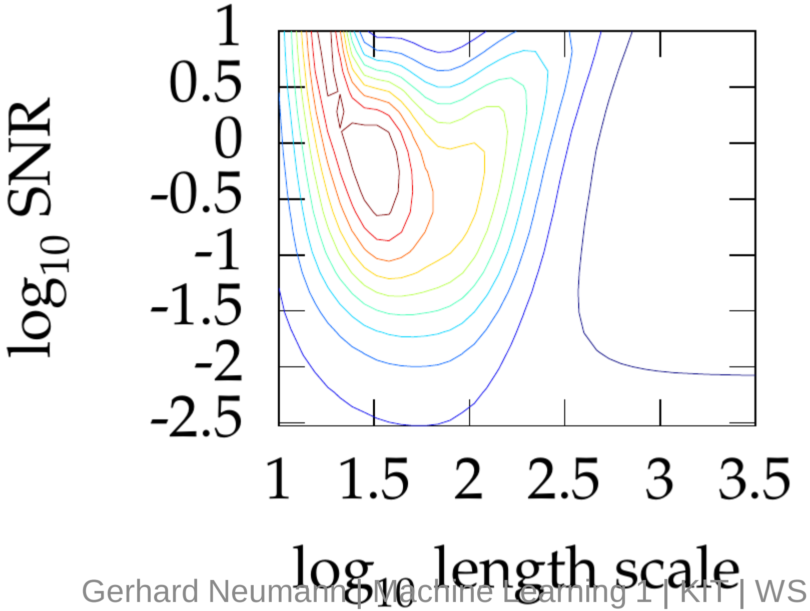
Example: GP Log-Likelihood

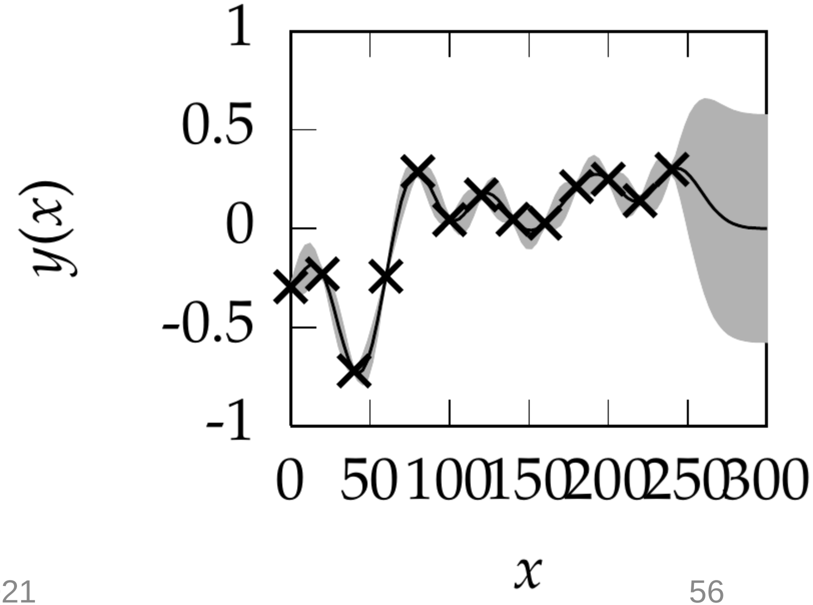
- Contour plot of the log-likelihood
- We can see multiple optima in the plot
- SNR = signal to noise ratio (ratio between lambda and sigma)



Example: Multiple Optima

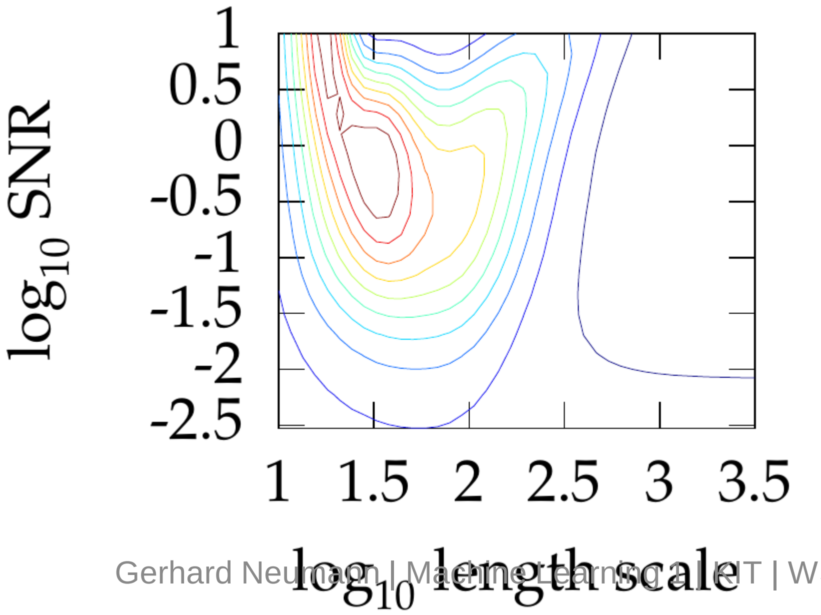
- Optimum 1: length scale of 1.2221 and log10 SNR of 1.9654
- Log-likelihood is -0.22317.

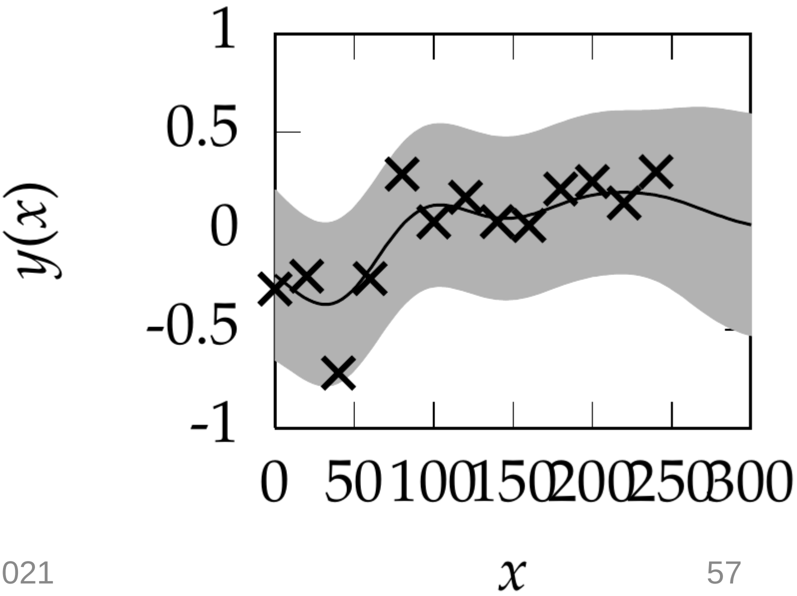




Example: Multiple Optima

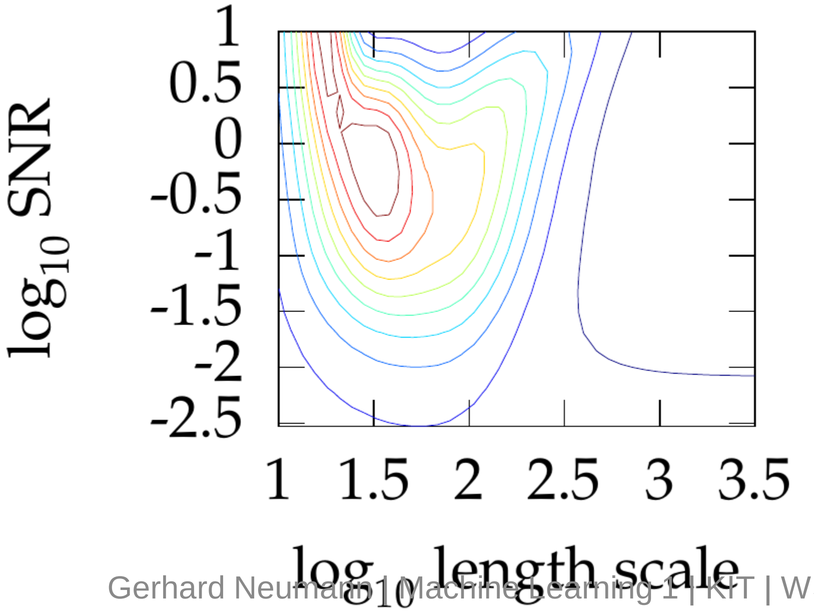
- Optimum 2: length scale of 1.5162 and log10 SNR of 0.21306
- Log-likelihood is -0.23604.

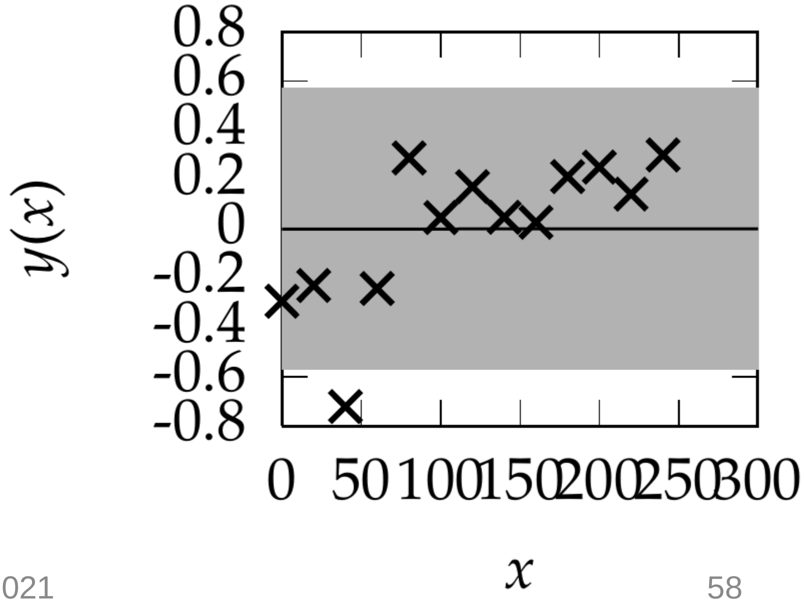




Example: Multiple Optima

- Optimum 3: length scale of 2.9886 and log10 SNR of -4.506
- Log-likelihood is -2.1056.





GPs: Summary

- GPs are a non-parametric Bayesian approach to regression with possibly infinite feature spaces
- Can estimate predictive uncertainty by integrating out model uncertainty
- Resulting prediction equations are straightforward and obtained in closed-form because of the Gaussian properties
- Hyperparameter optimization more complex, non-convex and expensive
- While GP for Regression is computationally very expensive, it is one of the most principled approaches to statistical learning for regression
- For small data-sets, they typically also outperform Neural Nets by a large margin

Today‘s Agenda!

Bayesian Learning:

- Posterior and Predictive Distribution
- Bayesian estimation for Gaussians
- Maximum A-posteriori (MAP) Estimates

Bayesian Regression Algorithms:

- Bayesian Linear Regression
- Gaussian Processes
- General Models and Approximations

Bayesian Model Selection

Basics:

Gaussian Identities:

- Completing the Square
- Gaussian Bayes Rules
- Gaussian Propagation

Bayesian Learning - Pros

Regularization can be obtained automatically

- No need for splitting into training and test sets
- Model comparison
- Automatic relevance detection (which inputs are important)

Uncertainty estimates can be obtained

- Active learning (determine where to sample next, Bayesian Optimization)
- Black-box learning approaches

Bayesian methods are a superset of many learning methods

- Theoretically among the most powerful method

Bayesian Learning - Cons

- **Requires to choose prior distributions, mostly based on analytic convenience rather than real knowledge about the problem**
- **Computationally intractable**
  - Posterior probabilities involve the computation of an integral

$$\underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} = \frac{\overbrace{p(\mathcal{D}|\boldsymbol{\theta})}^{\text{data likelihood}} \quad \overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{\int p(\boldsymbol{\theta}|\mathcal{D})p(\boldsymbol{\theta})d\boldsymbol{\theta}}_{\text{evidence}}}$$

  - In non-Bayesian statistics we estimate parameters with maximum likelihood estimation, for which you can still use gradient descent, if there is no analytical solutions

Example

What if the Gaussian assumptions do not hold?

Example: Bayesian Logistic Regression

- Prior: $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \lambda^{-1}\boldsymbol{I})$
- Bernoulli Likelihood: $p(y|\boldsymbol{w}, \boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x})^y(1 - \sigma(\boldsymbol{w}^T\boldsymbol{x}))^{(1-y)}$
- Posterior: $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}) = \dfrac{p(\boldsymbol{w})\prod_{i=1}^N p(y_i|\boldsymbol{w}, \boldsymbol{x}_i)}{p(\boldsymbol{X}, \boldsymbol{y})} = \dfrac{p(\boldsymbol{w})\prod_{i=1}^N p(y_i|\boldsymbol{w}, \boldsymbol{x}_i)}{\int p(\boldsymbol{w})\prod_{i=1}^N p(y_i|\boldsymbol{w}, \boldsymbol{x}_i)d\boldsymbol{w}}$

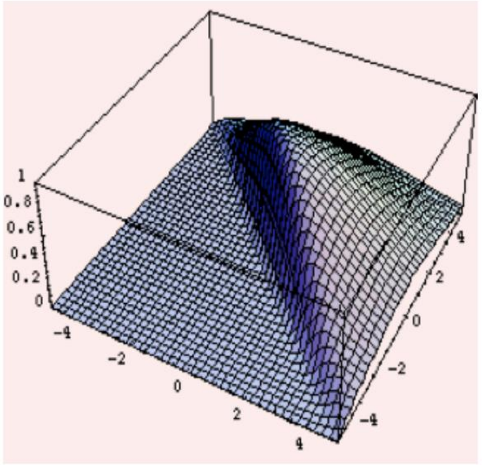
- Posterior is NOT Gaussian any more
- Intractable!

Example: Bayesian Logistic Regression

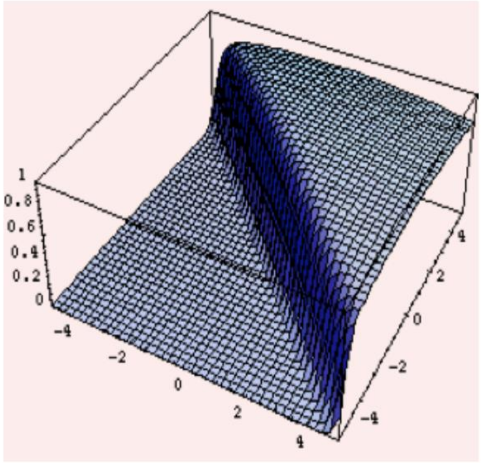
- Consider the dataset with N = 4:

$$\boldsymbol{X} = \begin{bmatrix} 5 & 5 \\ -5 & -5 \\ 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- Use only 2 data-points



High Prior Precision

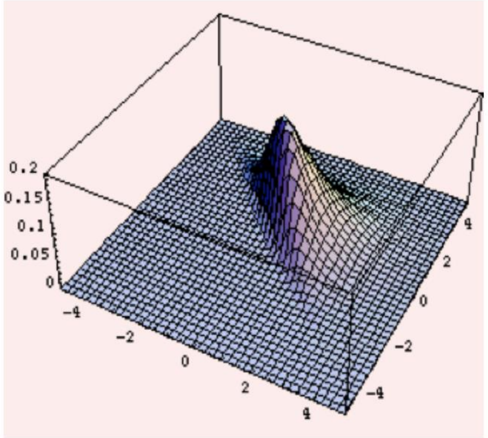


Low Prior Precision

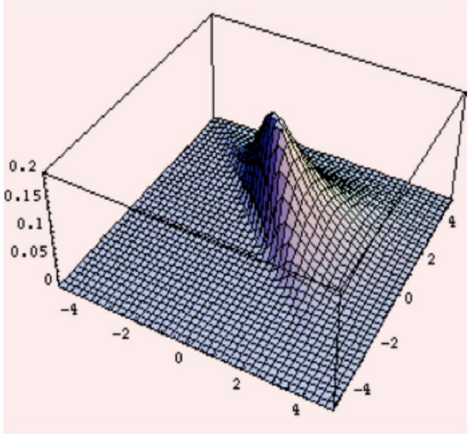
- Posterior is intractable, but can still be approximated by a grid in 2-D
- Non-Gaussian
- Low precision -> more acceptable solutions

Example: Bayesian Logistic Regression

- Use all data-points



High Prior Precision



Low Prior Precision

- Influence of prior diminishes with more samples!

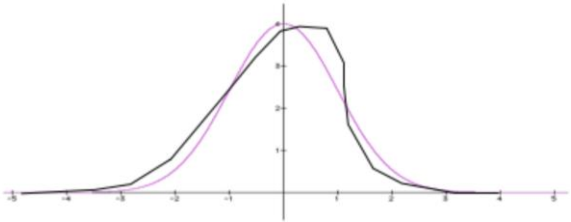
What can we do if posterior is intractable?

Sampling:

- We can still sample from it using Monte-Carlo Markov Chain (MCMC) sampling techniques (not covered)
- Use samples to approximate integrals in prediction step
- Computationally very expensive

Approximate intractable distribution:

- Laplace approximation



- Variational Inference (not covered)

Laplace Approximation

- **Assume the generic probability distribution**

$$p(z) = \frac{1}{Z}\tilde{p}(z) \text{ with } Z = \int \tilde{p}(z)dz$$

Intractable Integral

- **Goal:** Approximate p(z) with a Gaussian distribution, centred the mode z_0

$$\left.\frac{d\tilde{p}(z)}{dz}\right|_{z=z_0} = 0 \text{ ... as } z_0 \text{ is a (local) maximum/mode}$$

  - z_0 can be found by maximizing the likelihood (e.g., gradient descent)

- Hence, the 2nd order Taylor expansion in log domain is given by

$$\log \tilde{p}(z) \approx \log \tilde{p}(z_0) + \frac{1}{2}\left.\frac{d^2 \log \tilde{p}(z)}{d^2 z}\right|_{z=z_0}(z - z_0)^2$$

Laplace Approximation

- **Taking the exp, we get**

$$\tilde{p}(z) \approx \exp\left(-\frac{A(z-z_0)^2}{2}\right), \quad \text{with } A = -\frac{d^2 \log \tilde{p}(z)}{d^2 z}$$

- **Hence, the approximate normalized distribution is Gaussian**

$$p(z) \approx \left(\frac{A}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{A(z-z_0)^2}{2}\right) = \mathcal{N}(z_0, A^{-1})$$

  - Mean is given by the mode
  - Variance is given by 2nd order derivative
  - We do not need to evaluate intractable normalization constant Z

Laplace approximation

- **Multivariate case:**

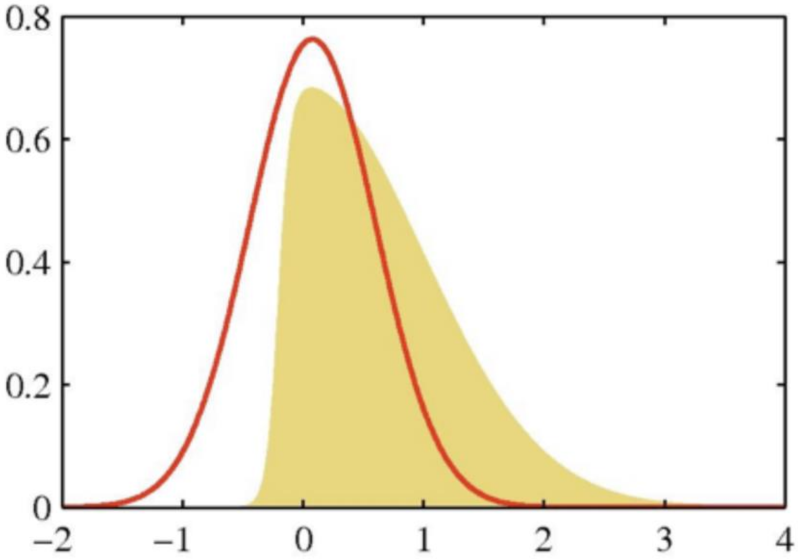
$$\log \tilde{p}(\boldsymbol{z}) \approx \log \tilde{p}(\boldsymbol{z}_0) - \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{z}_0)$$

  - where $\boldsymbol{A} = -\nabla_{\boldsymbol{z}}^2 \log \tilde{p}(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{z}_0}$ is the negative Hessian matrix

- **Consequently:**
  - Unnormalized density: $\tilde{p}(\boldsymbol{z}) \approx \tilde{p}(\boldsymbol{z}_0) \exp\left(-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{z}_0)\right)$
  - Normalized density: $p(\boldsymbol{z}) \approx \left|\frac{\boldsymbol{A}}{2\pi}\right|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{z}_0)\right) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{z}_0, \boldsymbol{A}^{-1})$

Laplace Approximation

- Illustration of the approximation for logistic regression



Today‘s Agenda!

Bayesian Learning:

- Posterior and Predictive Distribution
- Bayesian estimation for Gaussians
- Maximum A-posteriori (MAP) Estimates

Bayesian Regression Algorithms:

- Bayesian Linear Regression
- Gaussian Processes
- General Models and Approximations

Basics:

Gaussian Identities:

- Completing the Square
- Gaussian Bayes Rules
- Gaussian Propagation

Take-away messages

- **Bayesian learning consists of 2 steps:**
  - Compute posterior over parameters / models
  - Average over all parameters / models weighted by posterior
- **Both steps are in general intractable**
  - Can only be done for linear feature / kernelized regression models in closed form
  - For all other cases, we need to rely on approximations
- **However, theoretically one of the most powerful learning methods**
  - Robust against overfitting (averages over unspecified behaviour in between datapoints)
  - Does not require test set
  - Quantifies model uncertainty
  - Can be used for model selection
  - **Hot research topic:** Bayesian Neural Network

Self-test questions

- What are the 2 basic steps behind Bayesian Learning?
- Why is Bayesian Learning more robust against overfitting?
- What happens with the posterior if we add more data to the training set?
- What is completing the square and how does it work?
- For which 2 cases can Bayesian Learning be solved in closed form?
- Which approximations can we use if no closed form is available?
- How can we derive Bayesian Linear regression
- What is the advantage of Bayesian Linear regression to Ridge regression? What is the conceptual difference?
- What is the major advantage of GPs over Kernel Ridge Regression?
- Why are GPs a Bayesian approach?
- What principle allowed deriving GPs from a Bayesian regression point of view?