Introduction to Probabilistic Machine Learning

Ralf Herbrich, Rainer Schlosser

Bayesian Regression

Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Overview

1. Bayesian Linear Regression
2. Bayesian Linear Regression via Message Passing
   - Normal Distribution Revisited
   - Posterior and Predictive Distribution
3. Fast Bayesian Linear Regression

Overview

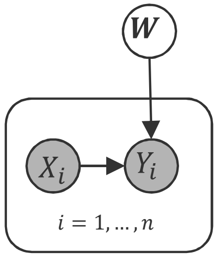
1. **Bayesian Linear Regression**
2. Bayesian Linear Regression via Message Passing
   - Normal Distribution Revisited
   - Posterior and Predictive Distribution
3. Fast Bayesian Linear Regression

Bayesian Inference of Linear Basis Function Models

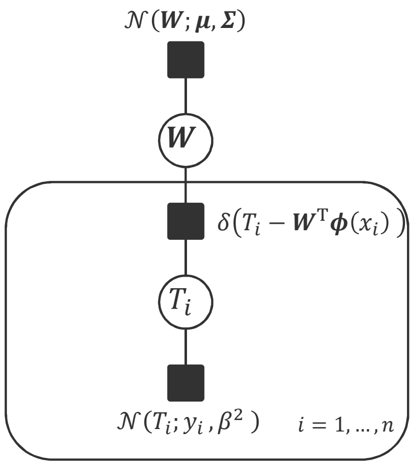
- **Given**:
  1. **Training Data**: $D \in (\mathcal{X} \times \mathbb{R})^n$ of n (labelled) examples (x_i, y_i)
  2. **Linear Basis Functions**: Basis function mapping $\boldsymbol{\phi}: \mathcal{X} \to \mathbb{R}^M$ and linear function model
     $$f(x; \boldsymbol{w}) := \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(x)$$
     weight vector feature vector
  3. **Likelihood of functions**:
     $$p(D|f) = p(D|\boldsymbol{w}) = \prod_{i=1}^{n} \mathcal{N}\left(y_i; \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(x_i), \beta^2\right)$$
  4. **Prior belief over functions**:
     $$p(f) = p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$
- **Bayesian Inference**: Posterior belief over functions

$$p(f|D) = p(\boldsymbol{w}|D) = \frac{\prod_{i=1}^{n} \mathcal{N}\left(y_i; \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(x_i), \beta^2\right) \cdot \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\int_{\mathbb{R}^M} \prod_{i=1}^{n} \mathcal{N}\left(y_i; \widetilde{\boldsymbol{w}}^{\mathrm{T}} \boldsymbol{\phi}(x_i), \beta^2\right) \cdot \mathcal{N}(\widetilde{\boldsymbol{w}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \, \mathrm{d}\widetilde{\boldsymbol{w}}}$$

Bayesian Network



Factor Graph



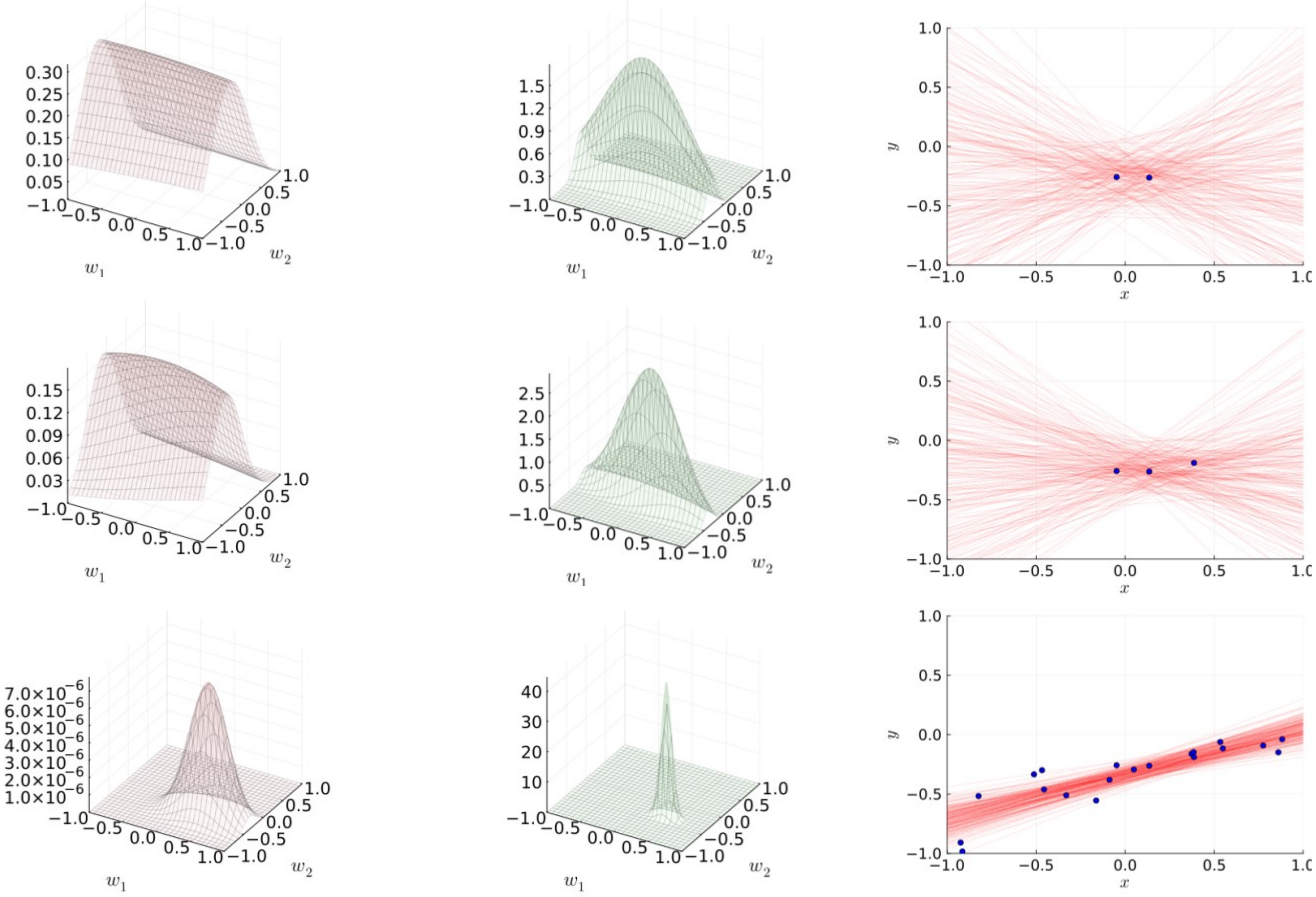
Bayesian Inference in Pictures

Likelihood | **Posterior** | **Input Space**

$n = 2$

$n = 3$

$m = 20$



$$f(x; \boldsymbol{w}) = w_1 x + w_2$$

$$p(y|x; \boldsymbol{w}) = \mathcal{N}(y; f(x), 0.2^2)$$

$$p(w_j) = \mathcal{N}(w_j; 0, 0.5)$$

Overview

1. Bayesian Linear Regression
2. **Bayesian Linear Regression via Message Passing**
   - **Normal Distribution Revisited**
   - Posterior and Predictive Distribution
3. Fast Bayesian Linear Regression

Multivariate Normal Distribution

- **Multivariate Normal Distribution**. *A continuous random variable* $\boldsymbol{X} \in \mathbb{R}^M$ *is said to have a multivariate normal distribution if the density is given by*

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

  where $\boldsymbol{\Sigma}$ *must be a positive definite* $M \times M$ *matrix and* $\boldsymbol{\mu} \in \mathbb{R}^M$.

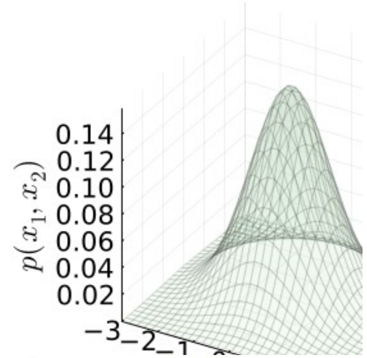
- **Properties**:

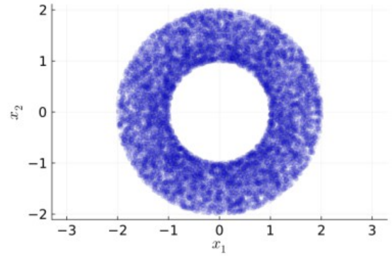
$$E[\boldsymbol{X}] = \boldsymbol{\mu}$$
$$\operatorname{cov}[\boldsymbol{X}] = \boldsymbol{\Sigma}$$

- **Covariance**. *For any two random variables* X_1 *and* X_2 *the covariance expresses the extent to which* X_1 *and* X_2 *vary together* **linearly** *and is given by*

$$\operatorname{cov}[X_1, X_2] = E[(X_1 - E[X_1]) \cdot (X_2 - E[X_2])] = E[X_1 X_2] - E[X_1] \cdot E[X_2]$$

  - Generalization of the variance to two random variables: $\operatorname{var}[X] = \operatorname{cov}[X, X]$
  - **Theorem**. *If two random variables* X_1 *and* X_2 *are independent, then* $\operatorname{cov}[X_1, X_2] = 0$. *The converse is not true!*





Multivariate Normal Distribution: Representations

- **Two Parameterizations (for different purposes):**
  - **Scale-Location Parameters**

$$\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = (2\pi)^{-\frac{M}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\cdot\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

  - **Natural Parameters**

$$\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau},\boldsymbol{P}) = (2\pi)^{-\frac{M}{2}}|\boldsymbol{P}|^{\frac{1}{2}}\cdot\exp\left(-\frac{1}{2}\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{P}^{-1}\boldsymbol{\tau}\right)\cdot\exp\left(\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{x}-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}\right)$$

- **Conversions**

$$\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{G}(\boldsymbol{x};\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu},\boldsymbol{\Sigma}^{-1})$$

Matrix inverse

$$\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau},\boldsymbol{P}) = \mathcal{N}(\boldsymbol{x};\boldsymbol{P}^{-1}\boldsymbol{\tau},\boldsymbol{P}^{-1})$$

Multivariate Normal Distributions: Products & Divisions

- **Theorem (Multiplication).** *Given two multi-dimensional Gaussian distributions* $\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_1,\boldsymbol{P}_1)$ *and* $\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_2,\boldsymbol{P}_2)$ *we have*

Gaussian density

$$\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_1,\boldsymbol{P}_1)\cdot\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_2,\boldsymbol{P}_2)=\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_1+\boldsymbol{\tau}_2,\boldsymbol{P}_1+\boldsymbol{P}_2)\cdot\mathcal{N}(\boldsymbol{\mu}_1;\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)$$

Additive updates!

- **Theorem (Division).** *Given two multi-dimensional Gaussian distributions* $\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_1,\boldsymbol{P}_1)$ *and* $\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_2,\boldsymbol{P}_2)$ *where* $\boldsymbol{P}_1-\boldsymbol{P}_2$ *is positive definite we have*

Correction factor

$$\frac{\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_1,\boldsymbol{P}_1)}{\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_2,\boldsymbol{P}_2)}=\frac{\mathcal{G}(\boldsymbol{x};\boldsymbol{\tau}_1-\boldsymbol{\tau}_2,\boldsymbol{P}_1-\boldsymbol{P}_2)}{\mathcal{N}(\boldsymbol{\mu}_1;\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2-\boldsymbol{\Sigma}_1)}\cdot\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_2-\boldsymbol{\Sigma}_1|}$$

Subtractive updates!

Gaussian density

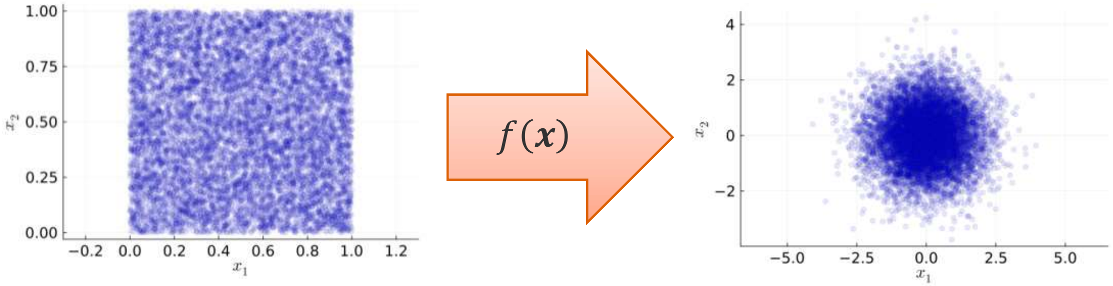
- Natural extension of the one-dimensional multiplication and division rules!

Sampling Multivariate Normal Distribution

- **Assumption**: We have access to a random number generator $X \sim \mathrm{Unif}([0,1])$
- **Box-Mueller**: If $X_1 \sim \mathrm{Unif}([0,1])$ and $X_2 \sim \mathrm{Unif}([0,1])$ then $f(\boldsymbol{X}) \sim N(\cdot;\mathbf{0},\boldsymbol{I})$ for

$$f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} \sqrt{-2\ln(x_1)} \cdot \cos(2\pi x_2) \\ \sqrt{-2\ln(x_1)} \cdot \sin(2\pi x_2) \end{bmatrix}$$

  - **In pictures:**



- **Sampling a multivariate Gaussian**. If $\boldsymbol{X} \sim \mathcal{N}(\cdot;\boldsymbol{\mu},\boldsymbol{\Sigma})$ then for $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{b}$

$$\boldsymbol{Y} \sim \mathcal{N}\left(\cdot;\boldsymbol{A\mu}+\boldsymbol{b},\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathrm{T}}\right)$$

  - For sampling a multivariate distribution, we require either the SVD or Cholesky decomposition of the covariance matrix, $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$
  - Can be easily proven from the properties of expectation and covariance

George Box
(1919 – 2013)

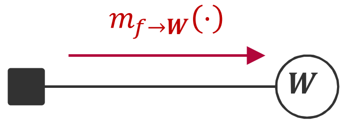
Mervin Mueller
(1928 – 2018)

Overview

1. Bayesian Linear Regression
2. **Bayesian Linear Regression via Message Passing**
   - Normal Distribution Revisited
   - **Posterior and Predictive Distribution**
3. Fast Bayesian Linear Regression

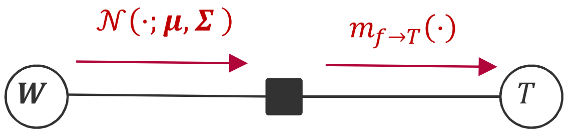
Multivariate Message Update Equations

Gaussian Factor $\mathcal{N}(\boldsymbol{W};\boldsymbol{\mu},\boldsymbol{\Sigma})$

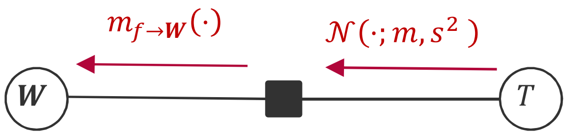


$$m_{f\to\boldsymbol{W}}(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w};\boldsymbol{\mu},\boldsymbol{\Sigma})$$

Gaussian Projection Factor $\delta\left(T - \boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}\right)$

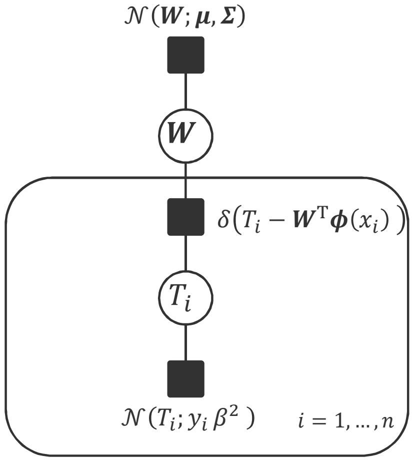


$$m_{f\to T}(t) = \int \delta\left(t - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\right)\cdot\mathcal{N}(\boldsymbol{w};\boldsymbol{\mu},\boldsymbol{\Sigma})\ d\boldsymbol{w} = \mathcal{N}\left(t;\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x},\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{x}\right)$$



$$m_{f\to\boldsymbol{W}}(\boldsymbol{w}) = \int \delta\left(t - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\right)\cdot\mathcal{N}(t;m,s^2)\ dt \propto \mathcal{G}\left(\boldsymbol{w};\frac{m}{s^2}\boldsymbol{x},\frac{1}{s^2}\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\right)$$

Factor Graph



Bayesian Linear Regression by Message Passing

- **Message**: Simple factor tree where each training example is summarized in an M-dimensional message
  - Prior Message $m_{1,0}(\boldsymbol{w}) = \mathcal{G}(\boldsymbol{w}; \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}) = p(\boldsymbol{w})$
  - Target Message $m_{2,i}(t_i) = \mathcal{N}(t_i; y_i, \beta^2) = p(y_i|t_i)$
  - Data Message $m_{1,i}(\boldsymbol{w}) = \mathcal{G}\left(\boldsymbol{w}; \beta^{-2} y_i \boldsymbol{\phi}(x_i), \beta^{-2}\boldsymbol{\phi}(x_i)\boldsymbol{\phi}^{\mathrm{T}}(x_i)\right) = p(y_i|\boldsymbol{w})$
- **Posterior**: Multiplying prior and data messages we have

$$p(\boldsymbol{w}|D) = \mathcal{G}\left(\boldsymbol{w}; \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \beta^{-2}\sum_{i=1}^{n} y_i \boldsymbol{\phi}(x_i), \boldsymbol{\Sigma}^{-1} + \beta^{-2}\sum_{i=1}^{n} \boldsymbol{\phi}(x_i)\boldsymbol{\phi}^{\mathrm{T}}(x_i)\right)$$

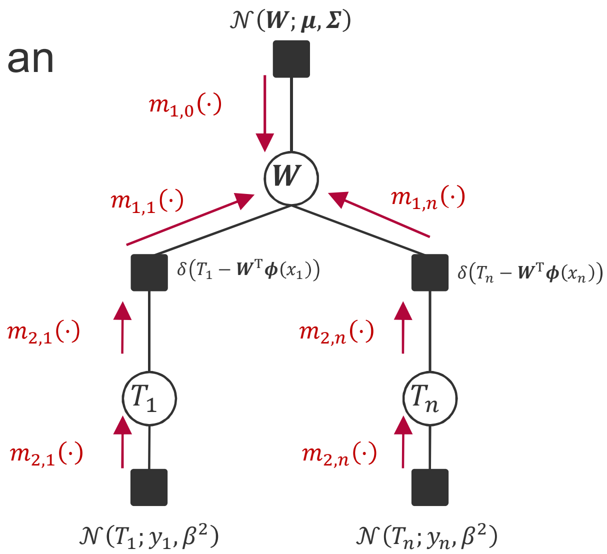
- **Feature Matrix**: All feature vectors are stacked on top of each other in a *feature matrix*

feature vector

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_M(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_M(x_n) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}^{\mathrm{T}}(x_1) \\ \vdots \\ \boldsymbol{\phi}^{\mathrm{T}}(x_n) \end{bmatrix}$$

$$\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y} = [\boldsymbol{\phi}(x_1) \quad \cdots \quad \boldsymbol{\phi}(x_n)] \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} y_i \boldsymbol{\phi}(x_i)$$

$$\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} = [\boldsymbol{\phi}(x_1) \quad \cdots \quad \boldsymbol{\phi}(x_n)] \begin{bmatrix} \boldsymbol{\phi}^{\mathrm{T}}(x_1) \\ \vdots \\ \boldsymbol{\phi}^{\mathrm{T}}(x_n) \end{bmatrix} = \sum_{i=1}^{n} \boldsymbol{\phi}(x_i)\boldsymbol{\phi}^{\mathrm{T}}(x_i)$$



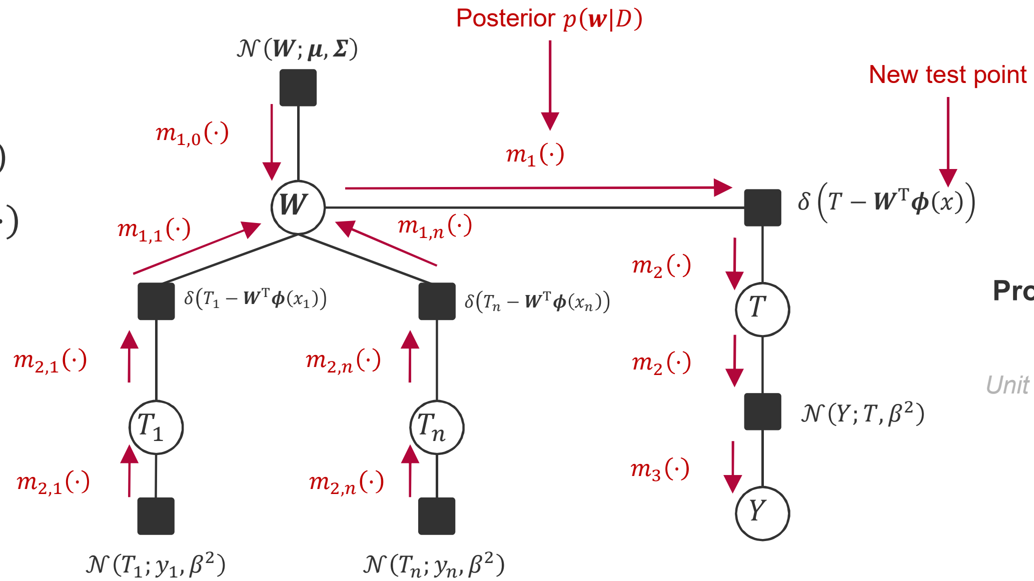
Bayesian Linear Regression: Training & Prediction

- **Posterior**: In terms of the feature matrix, it can be written as

$$
\begin{aligned}
p(\boldsymbol{w}|D) &= \mathcal{G}\left(\boldsymbol{w}; \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \beta^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y}, \boldsymbol{\Sigma}^{-1} + \beta^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right) \\
&= \mathcal{N}(\boldsymbol{w}; \boldsymbol{m}, \boldsymbol{S})
\end{aligned}
$$

$\boldsymbol{S}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \beta^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y}\right)$ $\quad$ $\left(\boldsymbol{\Sigma}^{-1} + \beta^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}$

- **Data model for prediction**:
  - Prediction at new test point x is $m_3(y)$
  - Posterior $p(\boldsymbol{w}|D)$ is the message $m_1(\cdot)$ to the Gaussian projection factor at the test point x
  - To avoid recomputing this message for every test point x we simply store the message $m_1(\boldsymbol{w}) = p(\boldsymbol{w}|D)$ as the "model"



Predictions

- **Predicition Tree**: Simple factor chain given posterior $p(\boldsymbol{w}|D) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{m}, \boldsymbol{S})$
  - Posterior Message $m_1(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{m}, \boldsymbol{S}) = p(\boldsymbol{w}|D)$
  - Projection Message $m_2(t) = \mathcal{N}\left(t; \boldsymbol{m}^{\mathrm{T}}\boldsymbol{\phi}(x), \boldsymbol{\phi}^{\mathrm{T}}(x) \cdot \boldsymbol{S} \cdot \boldsymbol{\phi}(x)\right) = p(t|x, D)$
  - Prediction Message $m_3(y) = \mathcal{N}\left(y; \boldsymbol{m}^{\mathrm{T}}\boldsymbol{\phi}(x), \beta^2 + \boldsymbol{\phi}^{\mathrm{T}}(x) \cdot \boldsymbol{S} \cdot \boldsymbol{\phi}(x)\right) = p(y|x, D)$
- **Bayesian Linear Regression in Matrix Notation**

Training

$$\boldsymbol{S} = \left(\boldsymbol{\Sigma}^{-1} + \beta^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}$$

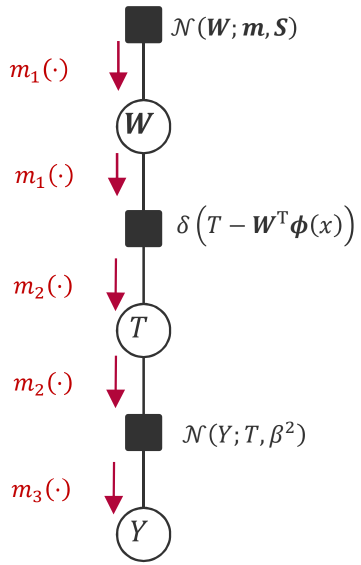
$$\boldsymbol{m} = \boldsymbol{S} \cdot \left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \beta^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y}\right)$$

Prediction

$$p(y|x, D) = \mathcal{N}\left(y; \boldsymbol{m}^{\mathrm{T}}\boldsymbol{\phi}(x), \beta^2 + \boldsymbol{\phi}^{\mathrm{T}}(x) \cdot \boldsymbol{S} \cdot \boldsymbol{\phi}(x)\right)$$

data uncertainty

model uncertainty



Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Bayesian Linear Regression: Example

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{0}, \lambda^2 \boldsymbol{I})$$

$\lambda = 10$

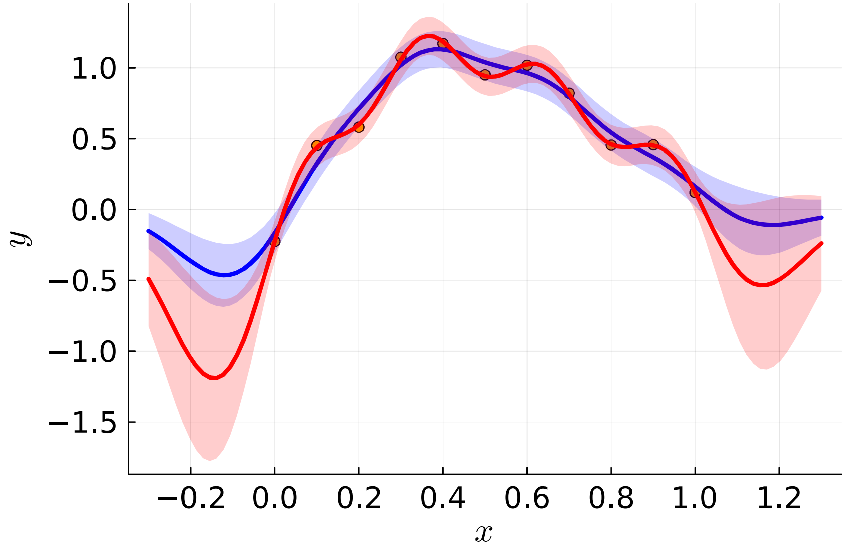
$\lambda = 1$

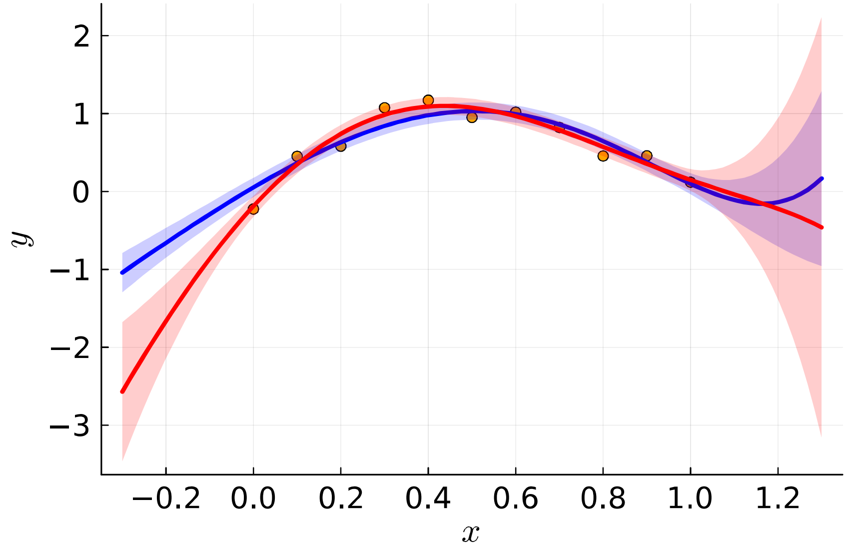
Gaussian Basis

$\phi_j(x) = \mathcal{N}(x; j, 0.15^2)$

Polynomial Basis

$\phi_j(x) = x^j$





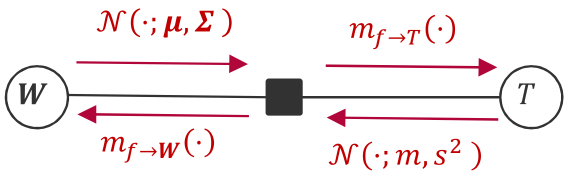
Overview

1. Bayesian Linear Regression
2. Bayesian Linear Regression via Message Passing
   - Normal Distribution Revisited
   - Posterior and Predictive Distribution
3. **Fast Bayesian Linear Regression**

Gaussian Projection Factor Revisited

Identical factors but different assumptions on $p(\boldsymbol{W})$

Gaussian Projection Factor $\delta(T - \boldsymbol{W}^{\mathrm{T}}\boldsymbol{x})$

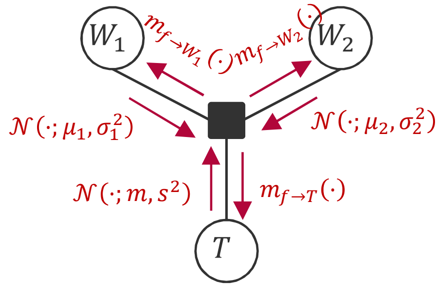


$$m_{f\to T}(t) = \int \delta(t - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) \cdot \mathcal{N}(\boldsymbol{w};\boldsymbol{\mu},\boldsymbol{\Sigma})\ d\boldsymbol{w} = \mathcal{N}(t;\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x},\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{x})$$

$$m_{f\to\boldsymbol{W}}(\boldsymbol{w}) = \int \delta(t - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) \cdot \mathcal{N}(t;m,s^2)\ dt \propto \mathcal{G}\left(\boldsymbol{w};\frac{m}{s^2}\boldsymbol{x},\frac{1}{s^2}\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\right)$$

Inverting the sum of these matrices is $\mathcal{O}(M^3)$

Weighted Sum Factor $\delta(T - \boldsymbol{W}^{\mathrm{T}}\boldsymbol{x})$



$$m_{f\to T}(t) = \mathcal{N}(t;x_1\mu_1 + x_2\mu_2, x_1^2\sigma_1^2 + x_2^2\sigma_2^2)$$

$$m_{f\to W_2}(w_2) = \mathcal{N}\left(w_2;\frac{m - x_1\mu_1}{x_2},\frac{s^2 + x_1^2\sigma_1^2}{x_2^2}\right)$$

$$m_{f\to W_1}(w_1) = \mathcal{N}\left(w_1;\frac{m - x_2\mu_2}{x_1},\frac{s^2 + x_2^2\sigma_2^2}{x_1^2}\right)$$

Computing the 1D-messages is $\mathcal{O}(M)$!

Fast Bayesian Linear Regression

- **Speeding up Bayesian Linear Regression**: Factorize the prior **and** posterior over the weight vector and then use message passing
  - Since x is fixed, we used $\boldsymbol{\phi} := \boldsymbol{\phi}(x)$
  - Message $m_{1,i}(w_i) = \mathcal{N}\left(w_i; \mu_i, \sigma_i^2\right)$
  - Message $m_3(t) = \mathcal{N}(t; y, \beta^2)$
  - Message $m_{2,i}(w_i) = \mathcal{N}\left(w_i; \phi_i^{-1} \cdot \left(y - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\phi} + \mu_i\phi_i\right), \phi_i^{-2} \cdot \left(\beta^2 + \sum_{j=1}^{M} \phi_j^2\sigma_j^2 - \phi_i^2\sigma_i^2\right)\right)$
- One can show that the product of $m_{1,i}(w_i)$ and $m_{2,i}(w_i)$ gives

$$\mu_i \leftarrow \mu_i + \frac{y - \boldsymbol{\mu}^T\boldsymbol{\phi}(x)}{\phi_i(x)} \cdot \left[\frac{\phi_i^2(x)\sigma_i^2}{\beta^2 + \sum_{j=1}^{M}\phi_j^2(x)\sigma_j^2}\right]$$

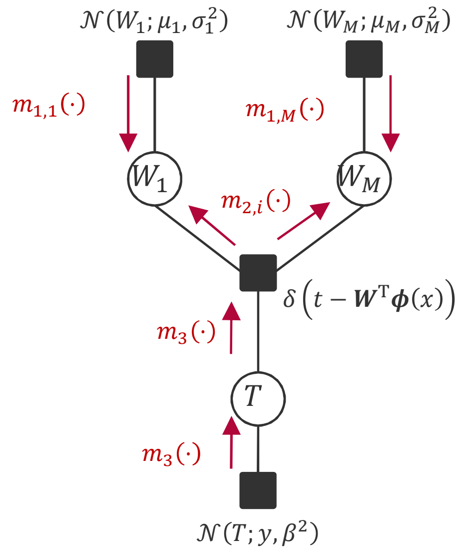
$$\sigma_i^2 \leftarrow \sigma_i^2 \cdot \left[1 - \frac{\phi_i^2(x)\sigma_i^2}{\beta^2 + \sum_{j=1}^{M}\phi_j^2(x)\sigma_j^2}\right]$$

target mismatch is measured in units of $\phi_i(x)$

multiplicative update

largest for parameter with largest uncertainty so far

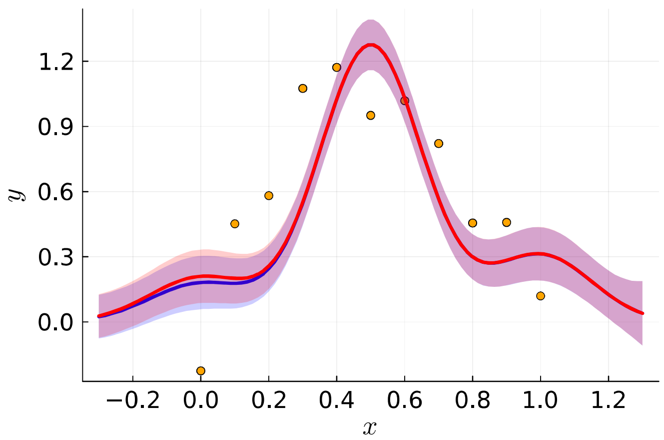
- In practice, each training example is only processed once.
  - Complexity reduces from $\mathcal{O}(n \cdot M^2 + M^3)$ to $\mathcal{O}(n \cdot M)$



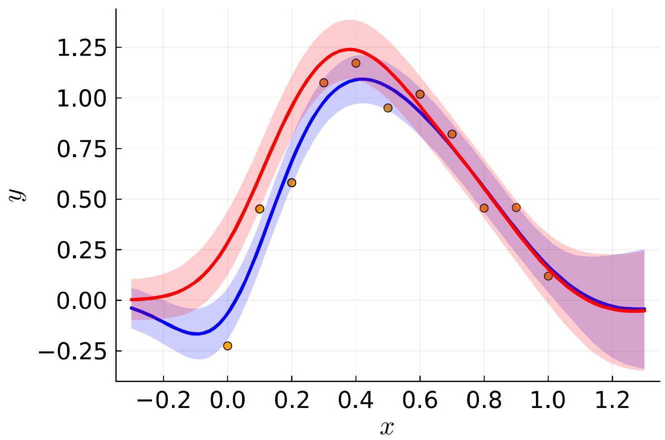
Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Speeding up Bayesian Linear Regression

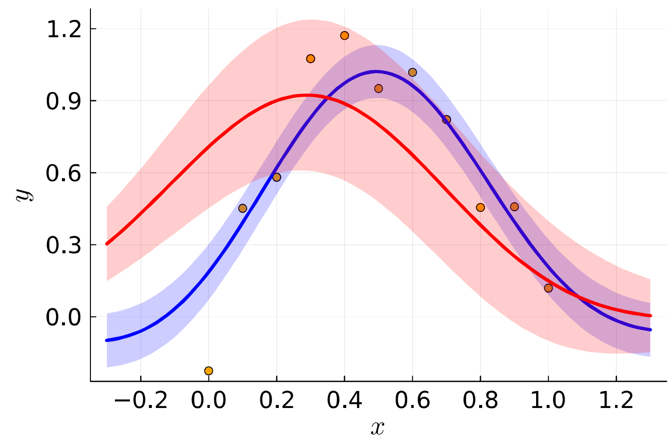
Nearly orthogonal features



Weakly correlated features



Strongly correlated features



Summary

1. **Bayesian Linear Regression**
   - Averaging over all functions weighting them by their posterior probability gives both a smoother mean and confidence intervals for each prediction (predictive distribution)
   - Message passing on the Bayesian Regression factor graph involves no loops and is exact
   - For linear basis function models with Normal noise, the posterior can be computed in closed form
   - Variance of Bayesian regression accounts for model uncertainty

2. **Fast Bayesian Linear Regression**
   - The Bayesian linear regression algorithm is of cubic complexity in the features and quadratic in the training set size
   - By factorizing *both* the prior and posterior distribution over the weight vector, we get a completely linear-complexity algorithm!

See you next week!