Lecture 9: Linear Regression

Yi, Yung (이융)

Mathematics for Machine Learning
KAIST EE

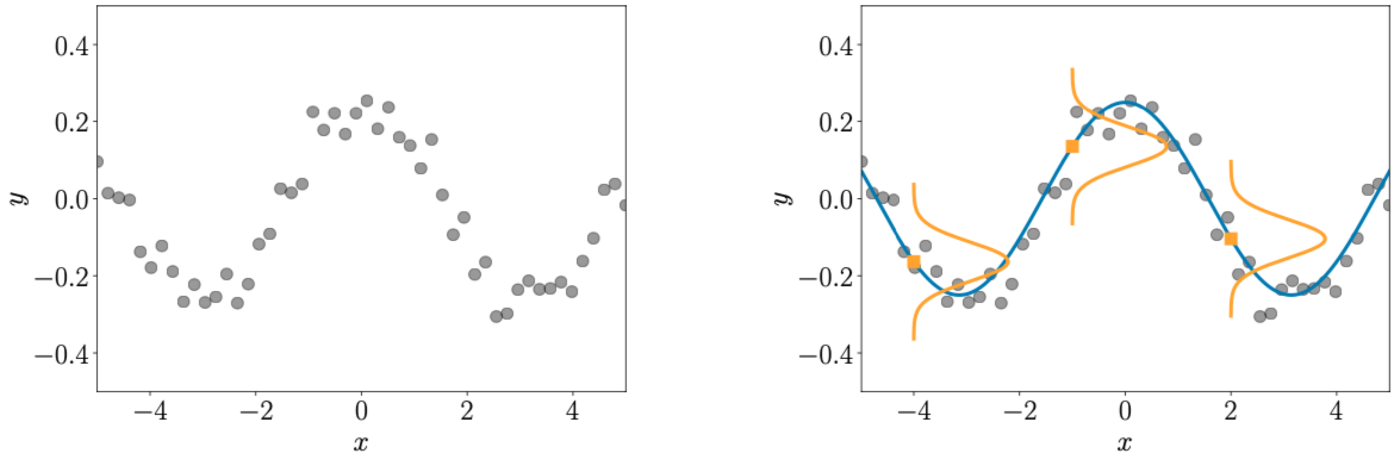
MONTH DAY, 2021

Roadmap

- Problem Formulation
- Parameter Estimation: ML
- Parameter Estimation: MAP
- Bayesian Linear Regression
- Maximum Likelihood as Orthogonal Projection

Roadmap

- Problem Formulation
- Parameter Estimation: ML
- Parameter Estimation: MAP
- Bayesian Linear Regression
- Maximum Likelihood as Orthogonal Projection



- For some input values x_n, we observe noisy function values $y_n = f(x_n) + \epsilon$
- Goal: infer the function f that generalizes well to function values at new inputs
- Applications: time-series analysis, control and robotics, image recognition, etc.

Formulation

- Linear regression, Gaussian noise
- Notation for simplification (this is how the textbook uses)

$$p(y|\mathbf{x}) = p_{Y|\mathbf{X}}(y|\mathbf{x}), \quad Y \sim \mathcal{N}(\mu, \sigma^2) \xrightarrow{\text{simplifies}} \mathcal{N}(y \mid f(\mathbf{x}), \sigma^2)$$

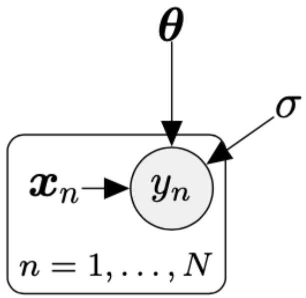
- Likelihood: for $\mathbf{x} \in \mathbb{R}^D$ and $y \in \mathbb{R}$, $p(y \mid \mathbf{x}) = \mathcal{N}(y \mid f(\mathbf{x}), \sigma^2)$
- $y = f(\mathbf{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$
- Linear regression with the parameter $\boldsymbol{\theta} \in \mathbb{R}^D$

$$p(y \mid \mathbf{x}) = \mathcal{N}(y \mid \mathbf{x}^\mathsf{T}\boldsymbol{\theta}, \sigma^2) \iff y = \mathbf{x}^\mathsf{T}\boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Prior with Gaussian nose: $p(y \mid \mathbf{x}) = \mathcal{N}(y \mid \mathbf{x}^\mathsf{T}\boldsymbol{\theta}, \sigma^2)$

Parameter Estimation

- Training set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$



- Assuming iid of N data, the likelihood is factorized into:

$$p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = \prod_{n=1}^{N} p(y_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}(y_n \mid \boldsymbol{x}_n^{\mathsf{T}}, \sigma^2),$$

  where $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and $\mathcal{Y} = \{y_1, \ldots, y_n\}$

- ML and MAP

Roadmap

- Problem Formulation
- Parameter Estimation: ML
- Parameter Estimation: MAP
- Bayesian Linear Regression
- Maximum Likelihood as Orthogonal Projection

MLE (Maximum Likelihood Estimation) (1)

- $\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \Big(-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) \Big)$
- For Gaussian noise with $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^{\mathsf{T}}$ and $\boldsymbol{y} = [y_1, \ldots, y_n]^{\mathsf{T}}$,

$$
\begin{aligned}
-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) &= -\log \prod_{n=1}^{N} p(y_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(y_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) \\
&= \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n^{\mathsf{T}} \boldsymbol{\theta})^2 + \text{ const} = \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 + \text{ const}
\end{aligned}
$$

Negative-log likelihood for $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{\theta} + \mathcal{N}(0, \sigma^2)$:

$$
-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 + \text{ const}
$$

MLE (Maximum Likelihood Estimation) (2)

- For Gaussian noise with $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\mathsf{T}$ and $\boldsymbol{y} = [y_1, \ldots, y_n]^\mathsf{T}$,

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2, \quad L(\boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2$$

- In this special case of Gaussian noise, finding MLE is equivalent to finding $\boldsymbol{\theta}$ that minimizes the empirical risk with squared loss function
  - Models as functions = Model as probabilistic models
- We find $\boldsymbol{\theta}$ such that $\frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{\theta}} = 0$

$$\frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{\theta}} = \frac{1}{2\sigma^2}\left(-2(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^\mathsf{T}\boldsymbol{X}\right) = \frac{1}{\sigma^2}\left(-\boldsymbol{y}^\mathsf{T}\boldsymbol{X} + \boldsymbol{\theta}^\mathsf{T}\boldsymbol{X}^\mathsf{T}\boldsymbol{X}\right) = 0$$

$$\iff \boldsymbol{\theta}_{\mathrm{ML}}^\mathsf{T}\boldsymbol{X}^\mathsf{T}\boldsymbol{X} = \boldsymbol{y}^\mathsf{T}\boldsymbol{X}$$

$$\iff \boldsymbol{\theta}_{\mathrm{ML}}^\mathsf{T} = \boldsymbol{y}^\mathsf{T}\boldsymbol{X}(\boldsymbol{X}^\mathsf{T}\boldsymbol{X})^{-1} \quad (\boldsymbol{X}^\mathsf{T}\boldsymbol{X} \text{ is positive definite if } \mathrm{rk}(\boldsymbol{X}) = D)$$

$$\iff \boldsymbol{\theta}_{\mathrm{ML}} = (\boldsymbol{X}^\mathsf{T}\boldsymbol{X})^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{y}$$

MLE with Features

- Linear regression: Linear in the parameters
  - $\phi(\boldsymbol{x})^\mathsf{T}\boldsymbol{\theta}$ is also fine, where $\phi(\boldsymbol{x})$ can be non-linear (we will cover this later)
  - $\phi(\boldsymbol{x})$ are the features
- Linear regression with the parameter $\boldsymbol{\theta} \in \mathbb{R}^K$, $\phi(\boldsymbol{x}) : \mathbb{R}^D \mapsto \mathbb{R}^K$:

$$p(y \mid \boldsymbol{x}) = \mathcal{N}(y \mid \phi(\boldsymbol{x})^\mathsf{T}\boldsymbol{\theta}, \sigma^2) \Longleftrightarrow y = \phi(\boldsymbol{x})^\mathsf{T}\boldsymbol{\theta} + \epsilon = \sum_{k=0}^{K-1} \theta_k \phi_k(\boldsymbol{x}) + \epsilon$$

- Example. Polynomial regression. For $x \in \mathbb{R}$ and $\boldsymbol{\theta} \in \mathbb{R}^K$, we lift the original 1-D input into K-D feature space with monomials x^k:

$$\phi(x) = \begin{pmatrix} \phi_0(x) \\ \vdots \\ \phi_{K-1}(x) \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ x^{K-1} \end{pmatrix} \in \mathbb{R}^K \quad \Longrightarrow \quad f(x) = \sum_{k=0}^{K-1} \theta_k x^k$$

Feature Matrix and MLE

- Now, for the entire training set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$,

$$\boldsymbol{\Phi} := \begin{pmatrix} \phi^{\mathsf{T}}(\boldsymbol{x}_1) \\ \vdots \\ \phi^{\mathsf{T}}(\boldsymbol{x}_N) \end{pmatrix} = \begin{pmatrix} \phi_0(\boldsymbol{x}_1) & \cdots & \phi_{K-1}(\boldsymbol{x}_1) \\ \vdots & \ldots & \vdots \\ \phi_0(\boldsymbol{x}_N) & \cdots & \phi_{K-1}(\boldsymbol{x}_N) \end{pmatrix} \in \mathbb{R}^{N\times K}, \quad \Phi_{ij} = \phi_j(\boldsymbol{x}_i),\ \phi_j : \mathbb{R}^D \mapsto \mathbb{R}$$

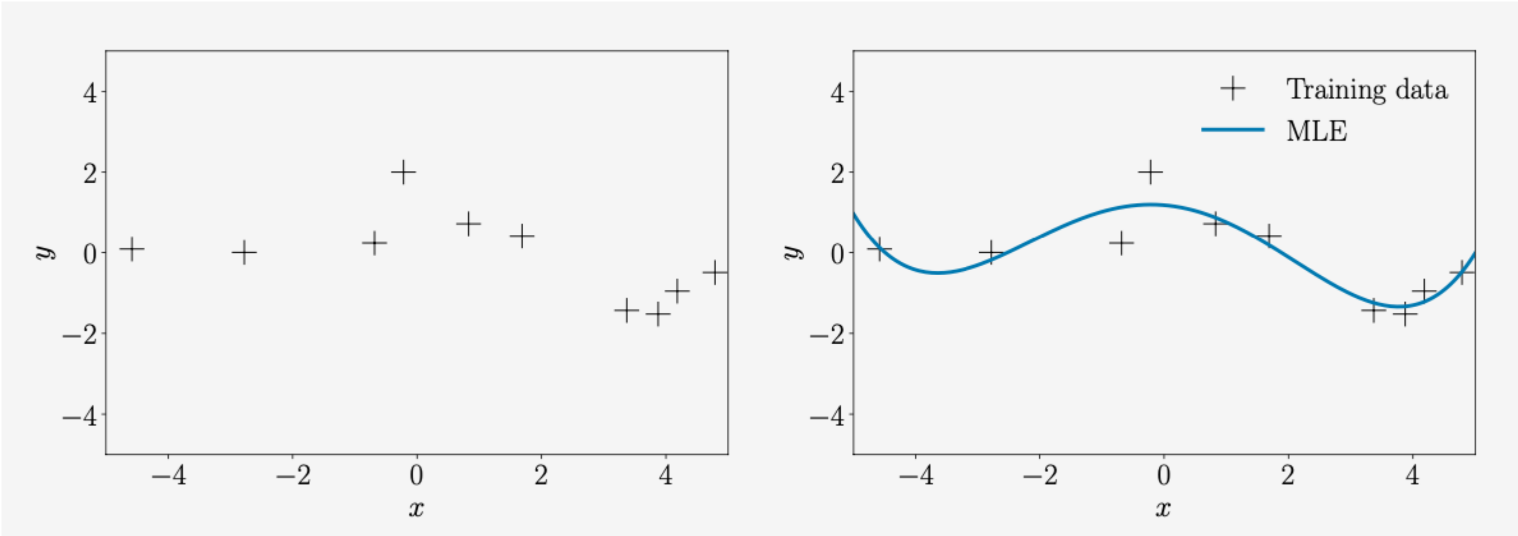
- Negative log-likelihood: Similarly to the case of $\boldsymbol{y} = \boldsymbol{X\theta}$,
  - $p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\Phi\theta}, \sigma^2 \boldsymbol{I})$
  - Negative-log likelihood for $f(\boldsymbol{x}) = \phi^{\mathsf{T}}(\boldsymbol{x})\boldsymbol{\theta} + \mathcal{N}(0, \sigma^2)$:

$$-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{\Phi\theta}\|^2 + \text{const}$$

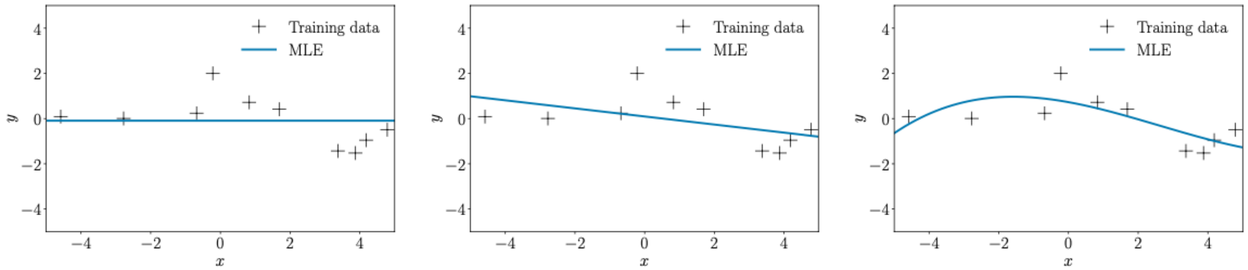
- MLE: $\boldsymbol{\theta}_{\text{ML}} = (\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{y}$

Polynomial Fit

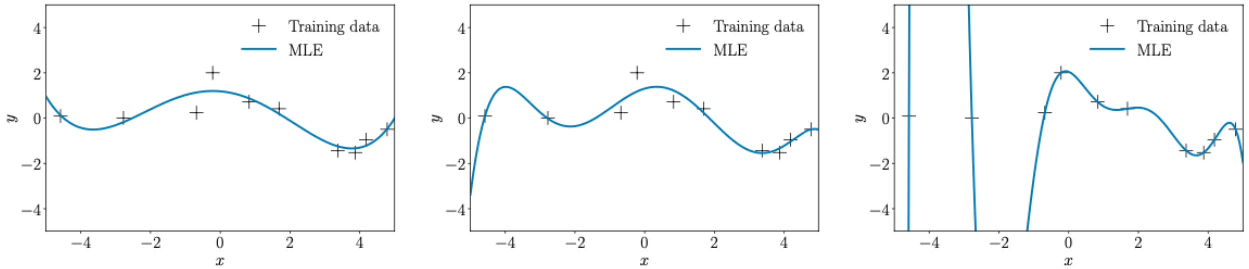
- $N = 10$ data, where $x_n \sim \mathcal{U}[-5, 5]$ and $y_n = -\sin(x_n/5) + \cos(x_n) + \epsilon$, $\epsilon \sim \mathcal{N}(0, 0.2^2)$
- Fit with poloynomial with degree 4 using ML



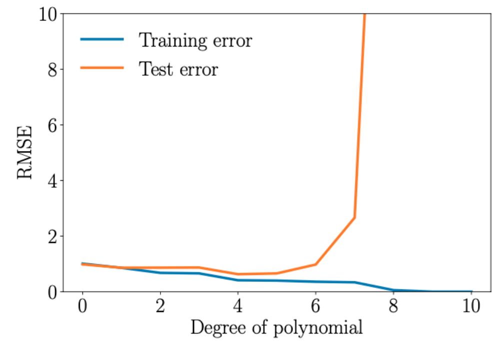
Overfitting in Linear Regression



(a) $M = 0$ (b) $M = 1$ (c) $M = 3$



(d) $M = 4$ (e) $M = 6$ (f) $M = 9$



- Higher polynomial degree is better (training error always decreases)
- Test error increases after some polynomial degree

Roadmap

- Problem Formulation
- Parameter Estimation: ML
- Parameter Estimation: MAP
- Bayesian Linear Regression
- Maximum Likelihood as Orthogonal Projection

MAPE (Maximum A Posteriori Estimation)

- MLE: prone to overfitting, where the magnitude of the parameters becomes large.
- a prior distribution $p(\boldsymbol{\theta})$ helps: what $\boldsymbol{\theta}$ is plausible
- MAPE and Bayes' theorem

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{Y} \mid \mathcal{X})} \implies \boldsymbol{\theta}_{\mathsf{MAP}} \in \arg\min_{\boldsymbol{\theta}} \Big(-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \Big)$$

- Gradient

$$-\frac{\mathrm{d} \log p(\boldsymbol{\theta} | \mathcal{X}, \mathcal{Y})}{\mathrm{d}\boldsymbol{\theta}} = -\frac{\mathrm{d} \log p(\mathcal{Y} | \mathcal{X}, \boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}} - \frac{\mathrm{d} \log p(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}$$

MAPE for Gausssian Prior (1)

- Example. A (conjugate) Gaussian prior $p(\boldsymbol{\theta}) \sim \mathcal{N}(0, b^2 \boldsymbol{I})$
  - For Gaussian likelihood, Gaussian prior $\implies$ Gaussian posterior
- Negative log-posterior

Negative-log posterior for $f(\boldsymbol{x}) = \phi^\mathsf{T}(\boldsymbol{x})\boldsymbol{\theta} + \mathcal{N}(0, \sigma^2)$ and $p(\boldsymbol{\theta}) \sim \mathcal{N}(0, b^2 \boldsymbol{I})$:

$$-\log p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y}) = \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{\Phi\theta})^\mathsf{T}(\boldsymbol{y} - \boldsymbol{\Phi\theta}) + \frac{1}{2b^2}\boldsymbol{\theta}^\mathsf{T}\boldsymbol{\theta} + \text{const}$$

- Gradient

$$-\frac{\mathrm{d} \log p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y})}{\mathrm{d}\boldsymbol{\theta}} = \frac{1}{\sigma^2}(\boldsymbol{\theta}^\mathsf{T}\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi} - \boldsymbol{y}^\mathsf{T}\boldsymbol{\Phi}) + \frac{1}{b^2}\boldsymbol{\theta}^\mathsf{T}$$

- MAP vs. ML

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \underbrace{\left(\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}\right)}_{(*)}^{-1}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{y}, \quad \boldsymbol{\theta}_{\mathrm{ML}} = (\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{y}$$

- The term $\frac{\sigma^2}{b^2}\boldsymbol{I}$
  - Ensures that $(*)$ is symmetric, strictly positive definite
  - Role of regularizer

Aside: MAPE for General Gausssian Prior (3)

- Example. A (conjugate) Gaussian prior $p(\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{S}_0)$
- Negative log-posterior

Negative-log posterior for $f(\boldsymbol{x}) = \phi^{\mathsf{T}}(\boldsymbol{x})\boldsymbol{\theta} + \mathcal{N}(0, \sigma^2)$ and $p(\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{S}_0)$:

$$-\log p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y}) = \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{m}_0)^{\mathsf{T}}\boldsymbol{S}_0{}^{-1}(\boldsymbol{\theta} - \boldsymbol{m}_0) + \text{const}$$

- We will use this later for computing the parameter posterior distribution in Bayesian linear regression.

Regularization: MAPE vs. Explicit Regularizer

- Explicit regularizer in regularized least squares (RLS)

$$\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2 + \lambda \|\boldsymbol{\theta}\|^2$$

- MAPE wth Gaussian prior $p(\boldsymbol{\theta}) \sim \mathcal{N}(0, b^2\boldsymbol{I})$
  - Negative log-Gaussian prior

$$-\log p(\boldsymbol{\theta}) = \frac{1}{2b^2}\boldsymbol{\theta}^\top\boldsymbol{\theta} + \text{const}$$

  - $\lambda = 1/2b^2$ is the regularization term

- Not surprising that we have

$$\boldsymbol{\theta}_{\text{RLS}} = \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y}$$

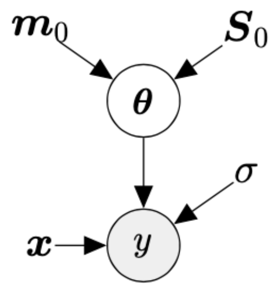
Roadmap

- Problem Formulation
- Parameter Estimation: ML
- Parameter Estimation: MAP
- Bayesian Linear Regression
- Maximum Likelihood as Orthogonal Projection

Bayesian Linear Regression

- Earlier, ML and MAP. Now, fully Bayesian
- Model

$$\begin{aligned} \text{prior} \quad & p(\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{S}_0) \\ \text{likelihood} \quad & p(y|\boldsymbol{x}, \boldsymbol{\theta}) \sim \mathcal{N}\left(y \mid \phi^{\mathsf{T}}(\boldsymbol{x})\boldsymbol{\theta}, \sigma^2\right) \\ \text{joint} \quad & p(y, \boldsymbol{\theta}|\boldsymbol{x}) = p(y \mid \boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{\theta}) \end{aligned}$$



- Goal: For an input $\boldsymbol{x}_*$, we want to compute the following posterior predictive distribution of y_*:

$$p(y_*|x_*, \mathcal{X}, \mathcal{Y}) = \int \overbrace{p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta})}^{\text{likelihood}} \overbrace{p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y})}^{(*)} \, \mathrm{d}\boldsymbol{\theta}$$

  - $(*)$: parameter posterior distribution that needs to be computed

- Parameter posterior distribution

$$p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{m}_N, \boldsymbol{S}_N), \quad \text{where}$$

$$\boldsymbol{S}_N = \left(\boldsymbol{S}_0{}^{-1} + \sigma^2 \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi}\right)^{-1}, \quad \boldsymbol{m}_N = \boldsymbol{S}_N\left(\boldsymbol{S}_0{}^{-1}\boldsymbol{m}_0 + \sigma^{-2}\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{y}\right)$$

(Proof of Sketch)

- From the negative-log posterior for general Gaussian prior,

$$-\log p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y}) = \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^\mathsf{T}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{m}_0)^\mathsf{T}\boldsymbol{S}_0{}^{-1}(\boldsymbol{\theta} - \boldsymbol{m}_0) + \text{const}$$

Parameter Posterior Distribution (2)

$$= \frac{1}{2}\Big(\sigma^{-2}\mathbf{y}^\top\mathbf{y} - 2\sigma^{-2}\mathbf{y}^\top\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\theta}^\top\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{\theta}^\top\mathbf{S}_0{}^{-1}\boldsymbol{\theta} - 2\mathbf{m}_0^\top\mathbf{S}_0{}^{-1}\boldsymbol{\theta} + \mathbf{m}_0^\top\mathbf{S}_0{}^{-1}\mathbf{m}_0\Big)$$
$$= \frac{1}{2}\Big(\boldsymbol{\theta}^\top(\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \mathbf{S}_0{}^{-1})\boldsymbol{\theta} - 2(\sigma^{-2}\boldsymbol{\Phi}^\top\mathbf{y} + \mathbf{S}_0{}^{-1}\mathbf{m}_0)^\top\boldsymbol{\theta}\Big) + \text{const}$$

- cyan color: quadratic term, orange color: linear term
- $p(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y}) \propto \exp($ quadratic in $\boldsymbol{\theta}$ $) \implies$ Gaussian distribution
- Assume that $p(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{m}_N,\mathbf{S}_N)$, and find $\mathbf{m}_N$ and $\mathbf{S}_N$.

$$-\log\mathcal{N}(\boldsymbol{\theta}|\mathbf{m}_N,\mathbf{S}_N) = \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_N)^\top\mathbf{S}_N{}^{-1}(\boldsymbol{\theta} - \mathbf{m}_N) + \text{const}$$
$$= \frac{1}{2}\Big(\boldsymbol{\theta}^\top\mathbf{S}_N{}^{-1}\boldsymbol{\theta} - 2\mathbf{m}_N{}^\top\mathbf{S}_N{}^{-1}\boldsymbol{\theta} + \mathbf{m}_N^\top\mathbf{S}_N{}^{-1}\mathbf{m}_N\Big) + \text{const}$$

Thus,

$$\mathbf{S}_N{}^{-1} = \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \mathbf{S}_0{}^{-1} \quad \text{and} \quad \mathbf{m}_N{}^\top\mathbf{S}_N{}^{-1} = (\sigma^{-2}\boldsymbol{\Phi}^\top\mathbf{y} + \mathbf{S}_0{}^{-1}\mathbf{m}_0{}^\top)$$

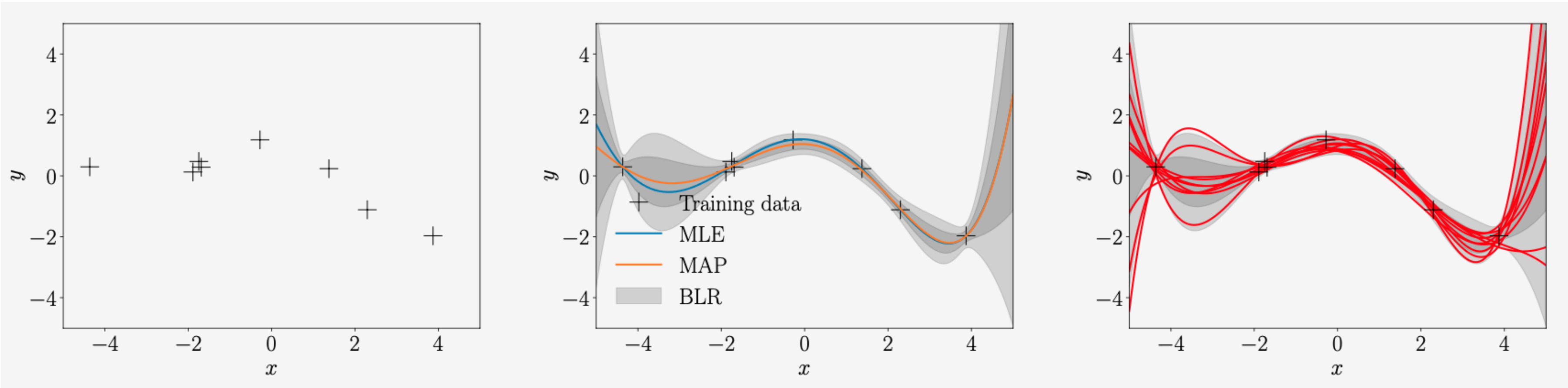
Posterior Predictions (1)

- Posterior predictive distribution

$$
\begin{aligned}
p(y_*|x_*, \mathcal{X}, \mathcal{Y}) &= \int p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y}) \mathrm{d}\boldsymbol{\theta} \\
&= \int \mathcal{N}\left(y_*|\phi^{\mathsf{T}}(\boldsymbol{x}_*)\boldsymbol{\theta}, \sigma^2\right) \mathcal{N}\left(\boldsymbol{\theta}|\boldsymbol{m}_N, \boldsymbol{S}_N\right) \mathrm{d}\boldsymbol{\theta} \\
&= \mathcal{N}\left(y_*|\phi^{\mathsf{T}}(\boldsymbol{x}_*)\boldsymbol{m}_N, \phi^{\mathsf{T}}(\boldsymbol{x}_*)\boldsymbol{S}_N \phi(\boldsymbol{x}_*) + \sigma^2\right)
\end{aligned}
$$

- The mean $\phi^{\mathsf{T}}(\boldsymbol{x}_*)\boldsymbol{m}_N$ coincides with the MAP estimate

Posterior Predictions (2)



- BLR: Bayesian Linear Regression

Computing Marginal Likelihood

- Likelihood: $p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta})$, Marginal likelihood: $p(\mathcal{Y}|\mathcal{X}) = \int p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$
- Recall that the marginal likelihood is important for model selection via Bayes factor:

$$\text{(Posterior odds)} = \frac{\mathbb{P}(M_1 \mid \mathcal{D})}{\mathbb{P}(M_2 \mid \mathcal{D})} = \frac{\frac{\mathbb{P}(\mathcal{D}|M_1)\mathbb{P}(M_1)}{\mathbb{P}(\mathcal{D})}}{\frac{\mathbb{P}(\mathcal{D}|M_2)\mathbb{P}(M_2)}{\mathbb{P}(\mathcal{D})}} = \underbrace{\frac{\mathbb{P}(M_1)}{\mathbb{P}(M_2)}}_{\text{Prior odds}} \underbrace{\frac{\mathbb{P}(\mathcal{D} \mid M_1)}{\mathbb{P}(\mathcal{D} \mid M_2)}}_{\text{Bayes factor}}$$

$$\begin{aligned} p(\mathcal{Y}|\mathcal{X}) &= \int p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \int \mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}\boldsymbol{\theta}, \sigma^2\boldsymbol{I})\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m}_0, \boldsymbol{S}_0)\,\mathrm{d}\boldsymbol{\theta} \\ &= \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\Phi}\boldsymbol{m}_0, \boldsymbol{\Phi}\boldsymbol{S}_0\boldsymbol{\Phi}^\top + \sigma^2\boldsymbol{I}) \end{aligned}$$

Roadmap

- Problem Formulation
- Parameter Estimation: ML
- Parameter Estimation: MAP
- Bayesian Linear Regression
- Maximum Likelihood as Orthogonal Projection

ML as Orthogonal Projection

- For $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\theta} + \mathcal{N}(0, \sigma^2)$, $\boldsymbol{\theta}_{\mathrm{ML}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y} = \dfrac{\boldsymbol{X}^\top \boldsymbol{y}}{\boldsymbol{X}^\top \boldsymbol{X}} \in \mathbb{R}$

$$\boldsymbol{X}\boldsymbol{\theta}_{\mathrm{ML}} = \frac{\boldsymbol{X}\boldsymbol{X}^\top}{\boldsymbol{X}^\top \boldsymbol{X}} \boldsymbol{y}$$

  - Orthogonal projection of $\boldsymbol{y}$ onto the one-dimensional subspace spanned by $\boldsymbol{X}$
- For $f(\boldsymbol{x}) = \phi^\top(\boldsymbol{x})\boldsymbol{\theta} + \mathcal{N}(0, \sigma^2)$, $\boldsymbol{\theta}_{\mathrm{ML}} = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{y} = \dfrac{\boldsymbol{\Phi}^\top \boldsymbol{y}}{\boldsymbol{\Phi}^\top \boldsymbol{\Phi}} \in \mathbb{R}$

$$\boldsymbol{\Phi}\boldsymbol{\theta}_{\mathrm{ML}} = \frac{\boldsymbol{\Phi}\boldsymbol{\Phi}^\top}{\boldsymbol{\Phi}^\top \boldsymbol{\Phi}} \boldsymbol{y}$$

  - Orthogonal projection of $\boldsymbol{y}$ onto the K-dimensional subspace spanned by columns of $\boldsymbol{\Phi}$

Summary and Other Issues (1)

- Linear regression for Gaussian likelihood and conjugate Gaussian priors. Nice analytical results and closed forms
- Other forms of likelihoods for other applications (e.g., classification)
- GLM (generalized linear model): $y = \sigma \circ f$ (σ: activation function)
  - No longer linear in $\boldsymbol{\theta}$
  - Logistic regression: $\sigma(f) = \dfrac{1}{1 + \exp(-f)} \in [0, 1]$ (interpreted as the probability of becoming 1)
  - Building blocks of (deep) feedforward neural nets
  - $\boldsymbol{y} = \sigma(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$. $\boldsymbol{A}$: weight matrix, $\boldsymbol{b}$: bias vector
  - K-layer deep neural nets: $\boldsymbol{x}_{k+1} = f_k(\boldsymbol{x}_k)$, $f_k(\boldsymbol{x}_k) = \sigma_k(\boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{b}_k)$

Summary and Other Issues (2)

- Gaussian process
  - A distribution over parameters $\rightarrow$ a distribution over functions
  - Gaussian process: distribution over functions without detouring via parameters
  - Closely related to BLR and support vector regression, also interpreted as Bayesian neural network with a single hidden layer and the infinite number of units
- Gaussian likelihood, but non-Gaussian prior
  - When $N << D$ (small training data)
  - Prior that enforces sparsity, e.g., Laplace prior
  - A linear regression with teh Laplace prior $=$ linear regression with LASSO (L1 regularization)

Questions?

Review Questions

1)