Principles of Machine Learning

Lecture 4: Linear Regression

Sharif University of Technology
Dept. of Aerospace Engineering

March 18, 2025

Table of Contents

Overfitting – MAP Estimation

We often observe that the magnitude of the parameter values becomes relatively large if we run into overfitting (Bishop, 2006).

To mitigate the the effect of huge parameter values $\longrightarrow$ placing a *prior* distribution $p(\boldsymbol{\theta})$ on the parameters

A prior $p(\theta) = \mathcal{N}(0, 1) \longrightarrow$ params expected to lie in $[-2, 2]$

maximum a posteriori (MAP) estimation:
Find params that maximize the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y})$ rather than maximizing the likelihood

$$\boxed{p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \frac{p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{Y} \mid \mathcal{X})}} \tag{18}$$

Overfitting – MAP Estimation

The posterior over the parameters $\boldsymbol{\theta}$, given the training data $\mathcal{X}, \mathcal{Y}$, is obtained by applying Bayes' theorem as (18).

The parameter vector $\boldsymbol{\theta}_{\mathrm{MAP}}$ that maximizes the posterior (18) is the MAP estimate.

To find the MAP estimate, we follow steps that are similar in flavor to maximum likelihood estimation.

Overfitting – MAP Estimation

Finding the MAP estimate

$$\log p(\boldsymbol{\theta} \mid \mathcal{X}, \mathcal{Y}) = \log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) + \text{const}$$

$\rightarrow$ MAP estimate will be a "compromise" between the prior and the data-dependent likelihood.

To find the MAP estimate $\boldsymbol{\theta}_{\text{MAP}}$ is to minimize the negative log-posterior wrt to $\boldsymbol{\theta}$, i.e., solving

$$\boldsymbol{\theta}_{\text{MAP}} \in \arg\min_{\boldsymbol{\theta}} \{-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})\}.$$

Finding the MAP estimate

The gradient of the negative log-posterior with respect to $\boldsymbol{\theta}$ is

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

With a Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, b^2\mathbf{I})$ on the params $\boldsymbol{\theta}$,

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

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

Finding the MAP estimate

Setting the gradient to $\mathbf{0}^\top$ and solving for $\boldsymbol{\theta}_{\text{MAP}}$:

$$\frac{1}{\sigma^2}(\boldsymbol{\theta}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} - \mathbf{y}^\top \boldsymbol{\Phi}) + \frac{1}{b^2}\boldsymbol{\theta}^\top = \mathbf{0}^\top$$

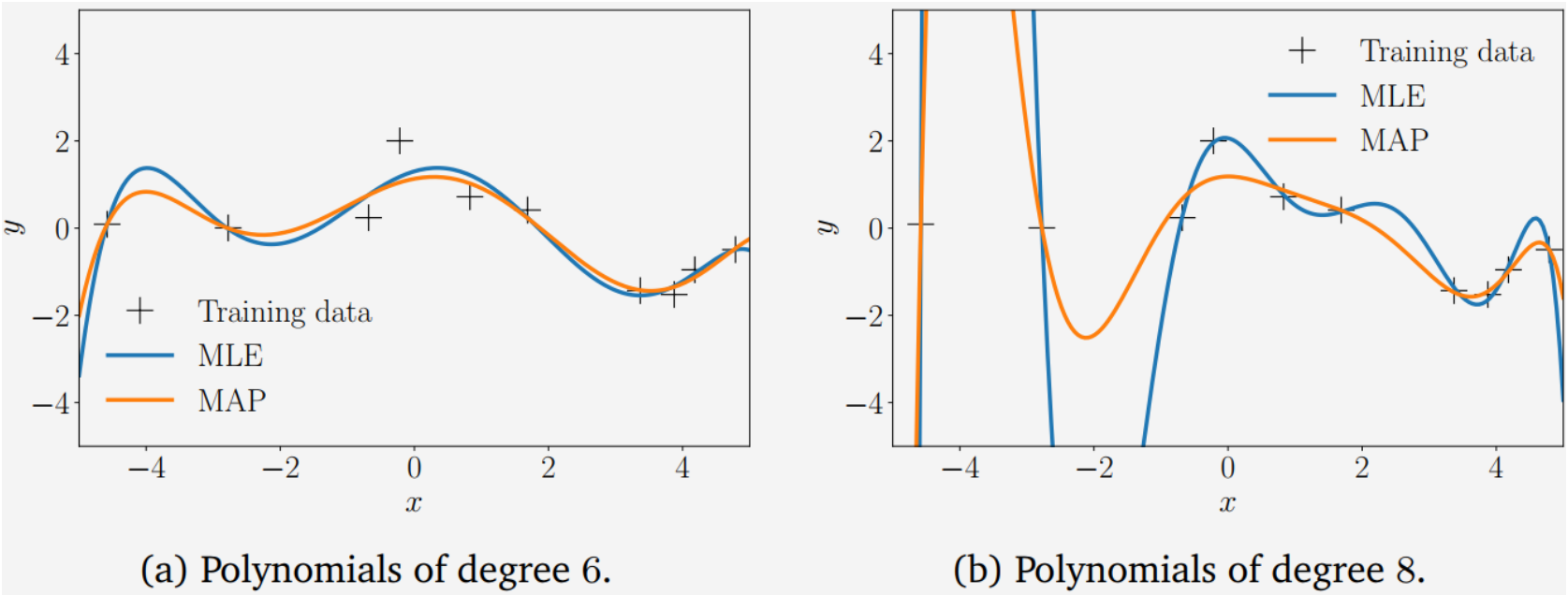
$$\iff \boldsymbol{\theta}^\top \left(\frac{1}{\sigma^2}\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{1}{b^2}\mathbf{I}\right) - \frac{1}{\sigma^2}\mathbf{y}^\top \boldsymbol{\Phi} = \mathbf{0}^\top$$

$$\iff \boldsymbol{\theta}^\top \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\mathbf{I}\right) = \mathbf{y}^\top \boldsymbol{\Phi}$$

$$\iff \boldsymbol{\theta}^\top = \mathbf{y}^\top \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\mathbf{I}\right)^{-1}$$

$$\boxed{\boldsymbol{\theta}_{\text{MAP}} = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\mathbf{I}\right)^{-1} \boldsymbol{\Phi}^\top \mathbf{y}} \quad (19)$$

Overfitting – MAP Estimation



(a) Polynomials of degree 6. (b) Polynomials of degree 8.

- The prior (regularizer) does not play a significant role for the low-degree polynomial, but keeps the function relatively smooth for higher-degree polynomials.

- Although the MAP estimate can push the boundaries of overfitting, it is not a general solution to this problem.

Overfitting – MAP Estimation as Regularization

Regularization are methods to mitigate the the effect of overfitting.

Rather than placing a prior distribution on the params $\boldsymbol{\theta}$, penalize the amplitude of the parameter by means of *regularization*.

For instance, in *regularized least squares* (RLS), the loss function is

$$\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \tag{20}$$

- The first term is a *data-fit* term (or *misfit term*): proportional to NLL.
- The second term is *regularizer*: the *regularization parameter* $\lambda \geq 0$ controls the "the strictness" of the regularization.

Overfitting – MAP Estimation as Regularization

The regularizer $\lambda \|\boldsymbol{\theta}\|_2^2$ in (20) can be interpreted as negative log-Gaussian prior that we used in MAP estimation.

With a Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, b^2\mathbf{I})$, the negative log-Gaussian prior is

$$-\log p(\boldsymbol{\theta}) = \frac{1}{2b^2} \|\boldsymbol{\theta}\|_2^2 + \text{const.}$$

- Note that for $\lambda = \frac{1}{2b^2}$,the regularization term and the negative log-Gaussian prior are the same.

Overfitting – MAP Estimation as Regularization

Given the regularized least-squares loss function in (19), minimizing it yields

$$\boxed{\boldsymbol{\theta}_{\mathrm{RLS}} = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda \mathbf{I})^{-1} \boldsymbol{\Phi}^\top \mathbf{y}},$$

which is the same as the MAP estimate in (19) for $\lambda = \frac{\sigma^2}{b^2}$,
where σ^2 is the noise variance and b^2 is the variance of the (isotropic) Gaussian prior $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, b^2\mathbf{I})$.

Table of Contents

Ridge Regression

ℓ_2 Regularization

MLE and MAP can result in overfitting.

A solution: use MAP estimation with a zero-mean Gaussian prior on the weights $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{0}, \lambda^{-1}\mathbf{I}) \longrightarrow$ *Ridge Regression*

$$\begin{aligned}\boldsymbol{\theta}^*_{\text{map}} &= \arg\min \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) + \frac{1}{2\tau^2}\boldsymbol{\theta}^\top\boldsymbol{\theta} \\ &= \arg\min \mathrm{RSS}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_2^2\end{aligned}$$

where $\lambda = \frac{\sigma^2}{\tau^2}$ is proportional to the strength of the prior and

Ridge Regression

$$\|\boldsymbol{\theta}\|_2 \triangleq \sqrt{\sum_{\mathrm{d}=1}^{\mathrm{D}} |\theta_{\mathrm{d}}|^2} = \sqrt{\boldsymbol{\theta}^\top \boldsymbol{\theta}}$$

is the ℓ_2 norm of the vector $\boldsymbol{\theta}$

$\rightarrow$ the weights that become too large in magnitude are getting penalized

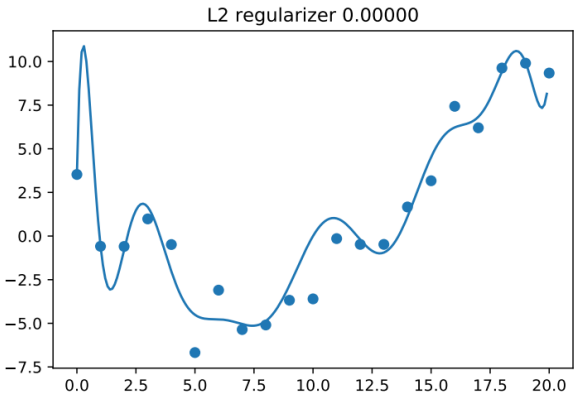
This is called ℓ_2 **regularization** or **weight decay**.

- Note that the offset term θ_0 is not penalized because

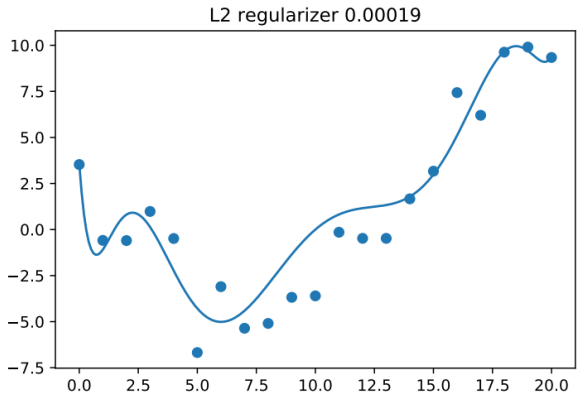
(1) Only affects the global mean of the output.

(2) Does not contribute to overfitting.

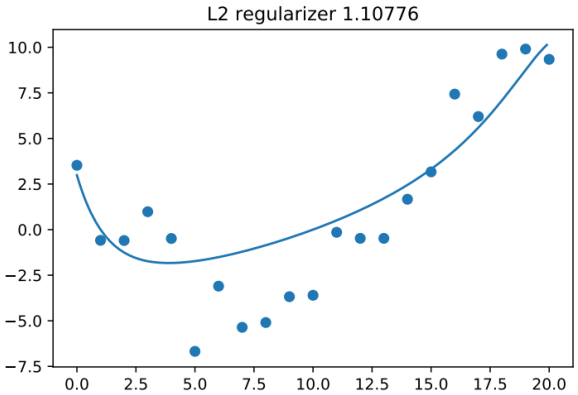
Ridge Regression



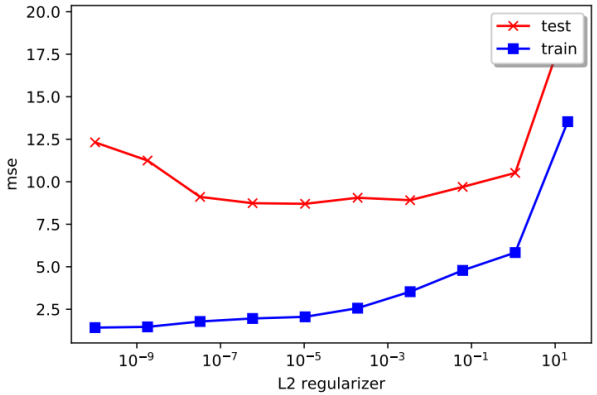
(a)



(b)



(c)



(d)

Figure: (a-c) Ridge regression applied to a degree 14 polynomial fit to 21 data points. (d) MSE vs strength of regularizer.

Table of Contents

Lasso Regression

Previous section:
assuming a Gaussian prior for the regression coefficients when fitting linear regression $\rightarrow$ encourages the params to be small $\rightarrow$ preventing overfitting.

May want some params to be exactly zero and not simply small $\rightarrow$ we want $\boldsymbol{\theta}^*$ be **sparse**, so that we minimize the **L0-norm**:

$$\|\boldsymbol{\theta}\|_0 = \sum_{\mathrm{d}=1}^{\mathrm{D}} \mathbb{I}(|\theta_d| > 0)$$

This is useful for **feature selection** where the prediction has the form

$$f(\mathbf{x}; \boldsymbol{\theta}) = \sum_{\mathrm{d}=1}^{\mathrm{D}} \theta_d x_d$$

To ignore a feature $x_d \Longrightarrow \theta_d = 0$

Lasso Regression

MAP estimation with a Laplace prior (ℓ_1 regularization)

To compute such sparse estimates, we focus on MAP estimation using the Laplace distribution as the prior:

$$p(\boldsymbol{\theta} \mid \lambda) = \prod_{\mathrm{d}=1}^{\mathrm{D}} \mathrm{Lap}(\theta_d \mid 0, 1/\lambda) \propto \prod_{\mathrm{d}=1}^{\mathrm{D}} e^{-\lambda|\theta_d|}$$

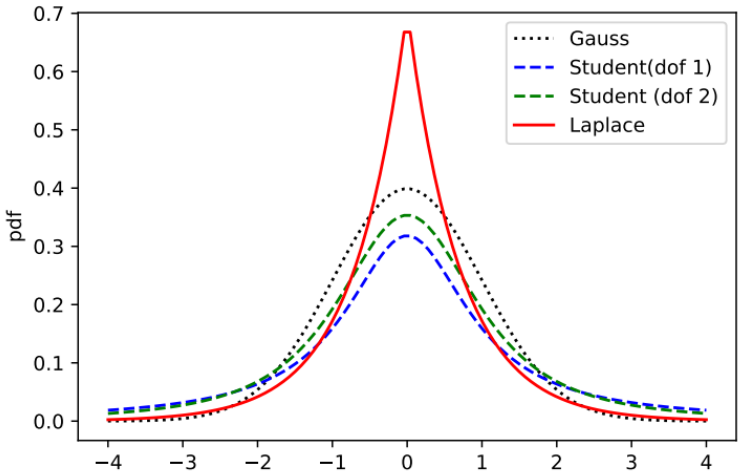
where λ is the sparsity parameter, and

$$\mathrm{Lap}(\theta \mid \mu, b) \triangleq \frac{1}{2b}\exp(-\frac{|\theta - \mu|}{b})$$

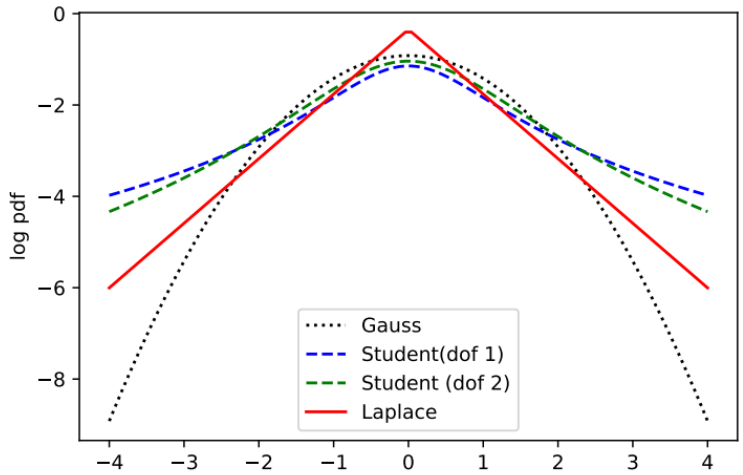
where μ is a location parameter and $b > 0$ is a sacle parameter.

MAP estimation with a Laplace prior (ℓ_1 regularization)

$\text{Lap}(\theta \mid 0, b)$ puts more density on 0 than $\mathcal{N}(\theta \mid 0, \sigma^2)$, even when we fix the variance to be the same.



(a)



(b)

Lasso Regression

MAP estimation with a Laplace prior (ℓ_1 regularization)

To perform MAP estimation with the discussed prior, we minimize

$$\mathrm{PNLL}(\boldsymbol{\theta}) = -\log p(\mathcal{D} \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta} \mid \lambda) = \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1$$

where $\|\boldsymbol{\theta}\| \triangleq \sum_{d=1}^{D} |\theta_d|$ is the ℓ_1 norm of $\boldsymbol{\theta}$.

- This called Lasso (least absolute shrinkage and selection prior) Regression.
- More generally, MAP estimation with a Laplace prior is called ℓ_1-**regularization**.

Lasso Regression

MAP estimation with a Laplace prior (ℓ_1 regularization)

Note that we could use other norms such as

$$\|\boldsymbol{\theta}\|_{\mathrm{q}} = \left(\sum_{\mathrm{d}=1}^{\mathrm{D}} |\theta_d|^q\right)^{1/q}$$

- Sparser solutions for $q < 1$
- ℓ_0-**norm** for $q = 0$
- For any $q < 1$, the problem becomes non-convex
  - Thus, ℓ_1-norm is the tightest **convex relaxation** of the ℓ_0-norm

Lasso Regression

Why does ℓ_1 regularization yield sparse solutions?

The lasso objective is the following non-smooth objective:

$$\min_{\boldsymbol{\theta}} \mathrm{NLL}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_1$$

which is the Lagrangian for the following quadratic program:

$$\min_{\boldsymbol{\theta}} \mathrm{NLL}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \|\boldsymbol{\theta}\|_1 \leq B$$

where B is an upper bound on the ℓ_1-norm of the weights.

Similarly, for ridge regression objective:

$$\min_{\boldsymbol{\theta}} \mathrm{NLL}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \|\boldsymbol{\theta}\|_2^2 \leq B$$

Lasso Regression

Why does ℓ_1 regularization yield sparse solutions?

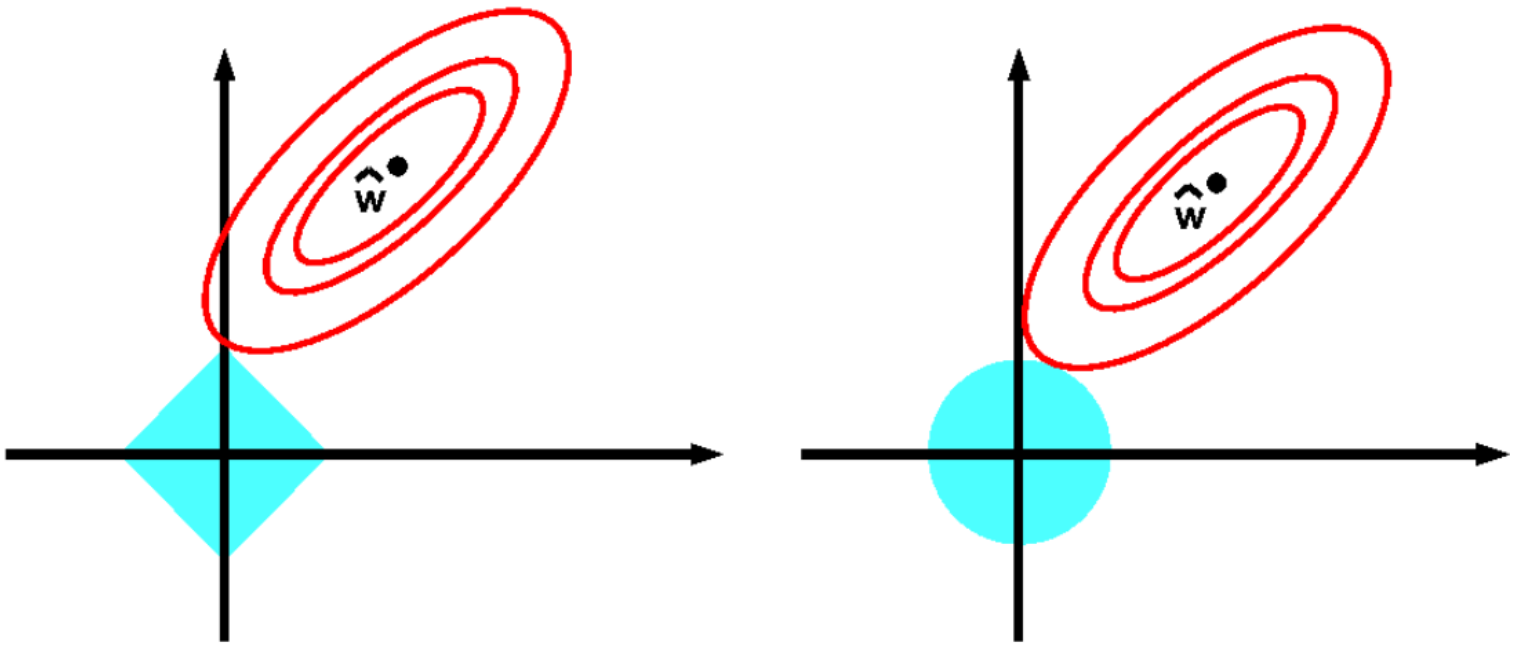


Figure: Illustration of ℓ_1 (left) vs ℓ_2 (right) regularization of a least squares problem

As constraint B is relaxed, the ℓ_1 ball grows until it meets the objective with corners more likely to intersect the ellipse than sides. The ℓ_2 ball can intersect the objective at any point, without a preference for sparsity.

Ridge vs Lasso

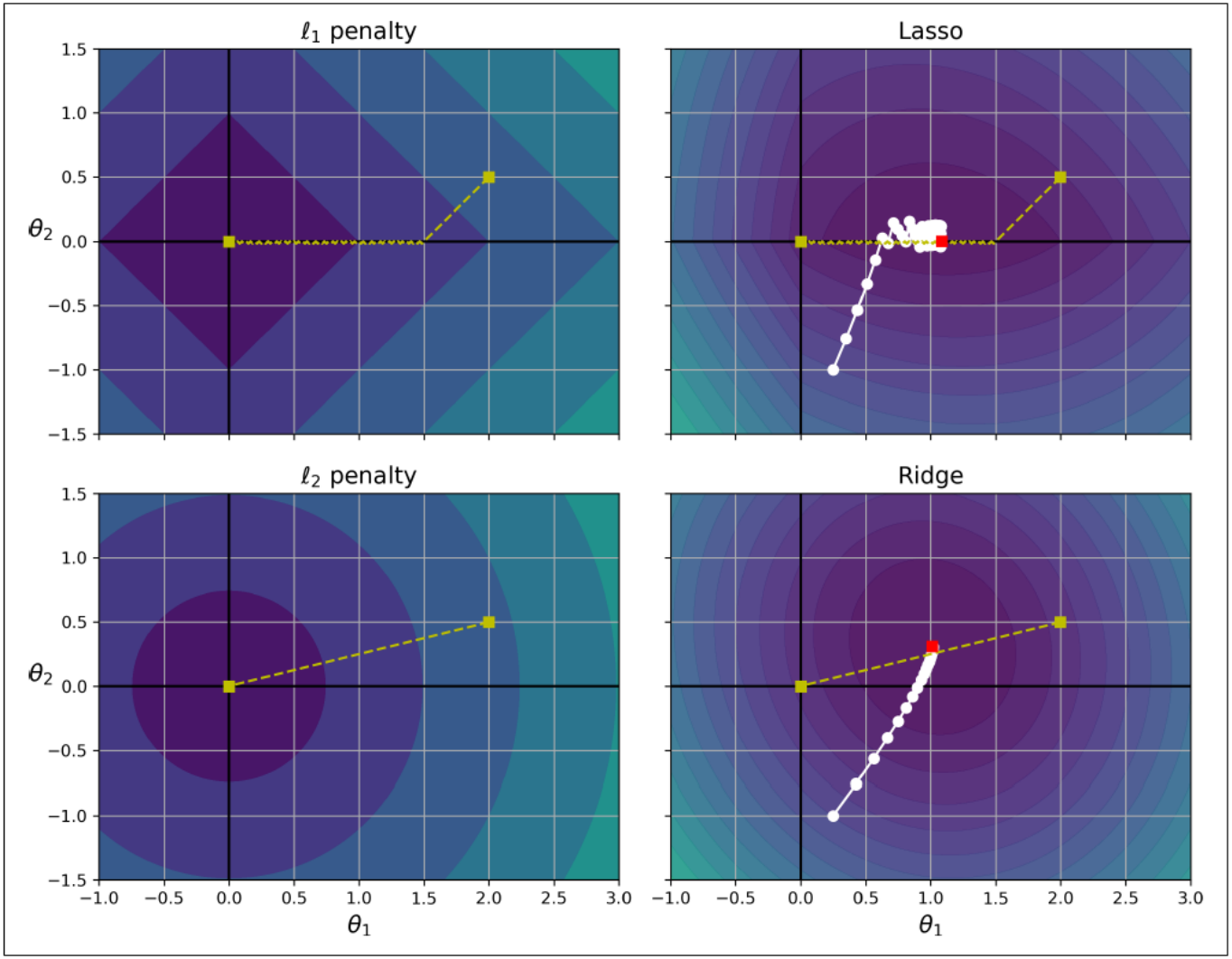


Figure: Ridge vs Lasso

Ridge vs Lasso

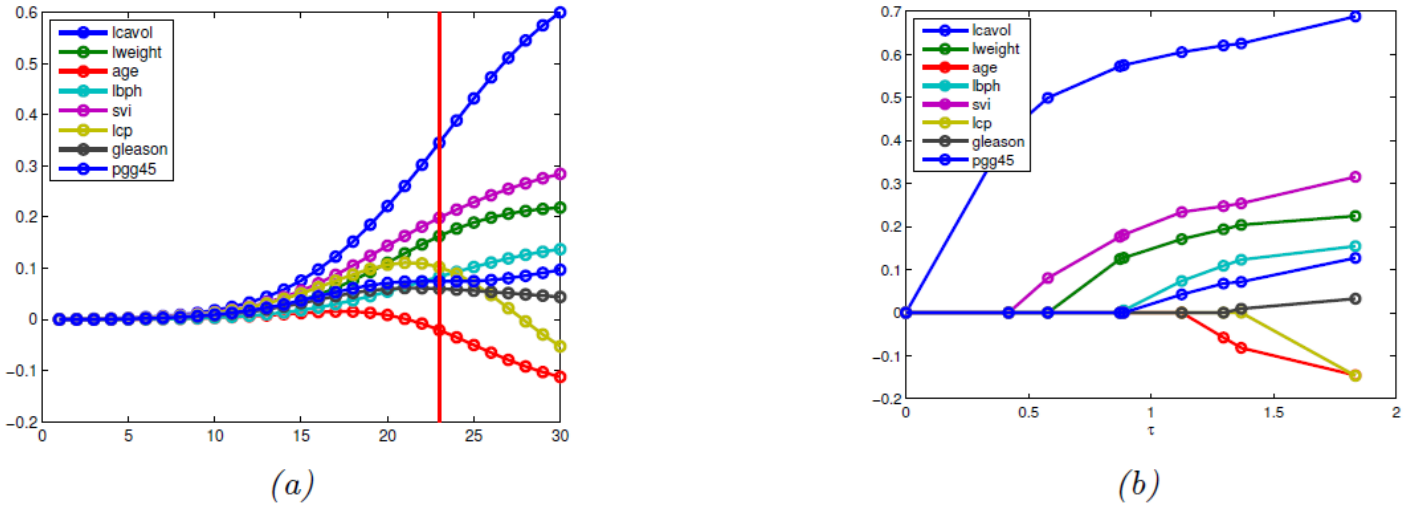


Figure: Effect of Regularization coef. on the parameters

Term	OLS	Best Subset	Ridge	Lasso
intercept	2.465	2.477	2.467	2.465
lcalvol	0.676	0.736	0.522	0.548
lweight	0.262	0.315	0.255	0.224
age	-0.141	0.000	-0.089	0.000
lbph	0.209	0.000	0.186	0.129
svi	0.304	0.000	0.259	0.186
lcp	-0.287	0.000	-0.095	0.000
gleason	-0.021	0.000	0.025	0.000
pgg45	0.266	0.000	0.169	0.083
Test error	0.521	0.492	0.487	0.457

Lasso Regression

Group Lasso

- Standard ℓ_1 regularization $\rightarrow$ assumes $1:1$ correspondence between parameters and variables
- More complex models $\rightarrow$ many parameters associated with a variable:
  - Each variable d associated with a vector of weights $\boldsymbol{\theta}_d$, then

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_D]$$

To exclude a variable d is to force the whole subvector $\boldsymbol{\theta}_d$ to go to zero $\rightarrow$ **group sparsity**

Group Lasso – Applications

- **Linear regression with categorical inputs**
  - If the d'th variable is categorical with K possible levels, then it will be represented as a one-hot vector of length K
- **Multinomial logistic regression**
  - The d'th variable will be associated with C different weights, one per class
- **Neural networks**
  - The k'th neuron will have multiple inputs
- **Multi-task learning**
  - Each input feature is associated with C different weights, one per output task

Group Lasso – Penalizing the two-norm

For group sparsity $\rightarrow$ partition the parameter vector into G groups
$\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G]$
The objective to minimize is

$$\mathrm{PNLL}(\boldsymbol{\theta}) = \mathrm{NLL}(\boldsymbol{\theta}) + \lambda \sum_{\mathrm{g}=1}^{\mathrm{G}} \|\boldsymbol{\theta}_g\|_2 \tag{21}$$

where $\|\boldsymbol{\theta}_g\|_2 = \sqrt{\sum_{d \in g} \theta_d^2}$ is the 2-norm of the group weight vector.

If the NLL is least squares $\rightarrow$ (21) is called **group lasso**

Elastic Net (ridge and lasso combined)

- Group lasso $\rightarrow$ specify the group structure ahead of time
- If the structure is not known & highly correlated coefficients are to be treated as an implicit group $\rightarrow$ **elastic net**

A hybrid between lasso and ridge regression:

$$\mathcal{L}(\boldsymbol{\theta}, \lambda_1, \lambda_2) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2 + \lambda_2 \|\boldsymbol{\theta}\|_2^2 + \lambda_1 \|\boldsymbol{\theta}\|_1 \tag{22}$$

The (22) is *strictly convex* (assuming $\lambda_2 > 0$) $\rightarrow$ there exists a unique global minimum even if the $\mathbf{X}$ is not full-rank

Elastic Net – Advantages

- **grouping effect**: regression coefficients of highly correlated variables tend to be equal
- If two features are identical ($\mathbf{X}_{:j} = \mathbf{X}_{:k}$)$\rightarrow$ their estimates are also equal ($\theta_j^* = \theta_k^*$)
- If $D > N \rightarrow$ elastic net can select more than N non-zero variables on its path to the dense estimate $\rightarrow$ the max number of non-zero elements that can be selected is N (excluding the MLE) $\rightarrow$ exploring more possible subsets of variables

Principles of Machine Learning

Lecture 5: Bayesian Linear Regression

Sharif University of Technology
Dept. of Aerospace Engineering

March 18, 2025

Outline

From Point Estimates to Bayesian Inference

Key Limitations of MLE/MAP:

- Point estimates (*e.g.*, $\boldsymbol{\theta}_{\text{MLE}}$ or $\boldsymbol{\theta}_{\text{MAP}}$) can lead to overfitting.
- No inherent uncertainty quantification in predictions.

Bayesian Linear Regression (BLR):

- Computes a **posterior distribution** over parameters $p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y})$.
- Predictions average over *all plausible parameters*, weighted by their posterior probability.
- Naturally incorporates regularization via priors.

The Perils of Maximum Likelihood

- **Problem**: MLE finds parameters maximizing $p(\mathcal{D}|\theta)$
- **Risk**: Perfect fit to training data $\Rightarrow$ poor generalization

Small Data Challenges

Key Issues:

- High parameter-to-data ratio
- Noise mistaken for signal
- No uncertainty quantification

Numerical Example

Training Data:

- $X = [0.5, 1.2, 2.3]$
- $y = [1.1, 1.8, 3.9]$

MLE fit: $y = 0.8 + 1.5x - 0.3x^2$

True relationship: $y = 1 + 1x + \epsilon$

Outline

Probabilistic Model

Prior: Gaussian distribution over parameters

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{S}_0)$$

Likelihood: Gaussian noise assumption

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\boldsymbol{\phi}^\top(\boldsymbol{x})\boldsymbol{\theta}, \sigma^2)$$

Joint Distribution:

$$p(\boldsymbol{y}, \boldsymbol{\theta}|\boldsymbol{X}) = p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})$$

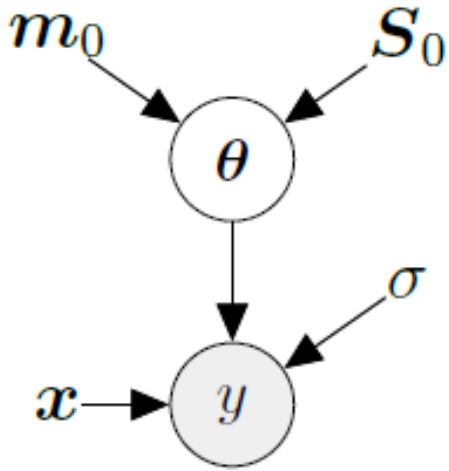


Figure: Graphical Model

Outline

Predictions Before Observing Data

Prior Predictive Distribution:

$$p(y_*|\boldsymbol{x}_*) = \int \underbrace{p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta})}_{\text{likelihood}} \underbrace{p(\boldsymbol{\theta})}_{\text{prior}} d\boldsymbol{\theta}$$

Closed-Form Solution (Conjugate Prior):

$$p(y_*|\boldsymbol{x}_*) = \mathcal{N}\left(\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_0,\ \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_0\boldsymbol{\phi}(\boldsymbol{x}_*) + \sigma^2\right)$$

- Uncertainty includes parameter prior ($\boldsymbol{S}_0$) and noise (σ^2).

Outline

Parameter Posterior via Bayes' Theorem

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

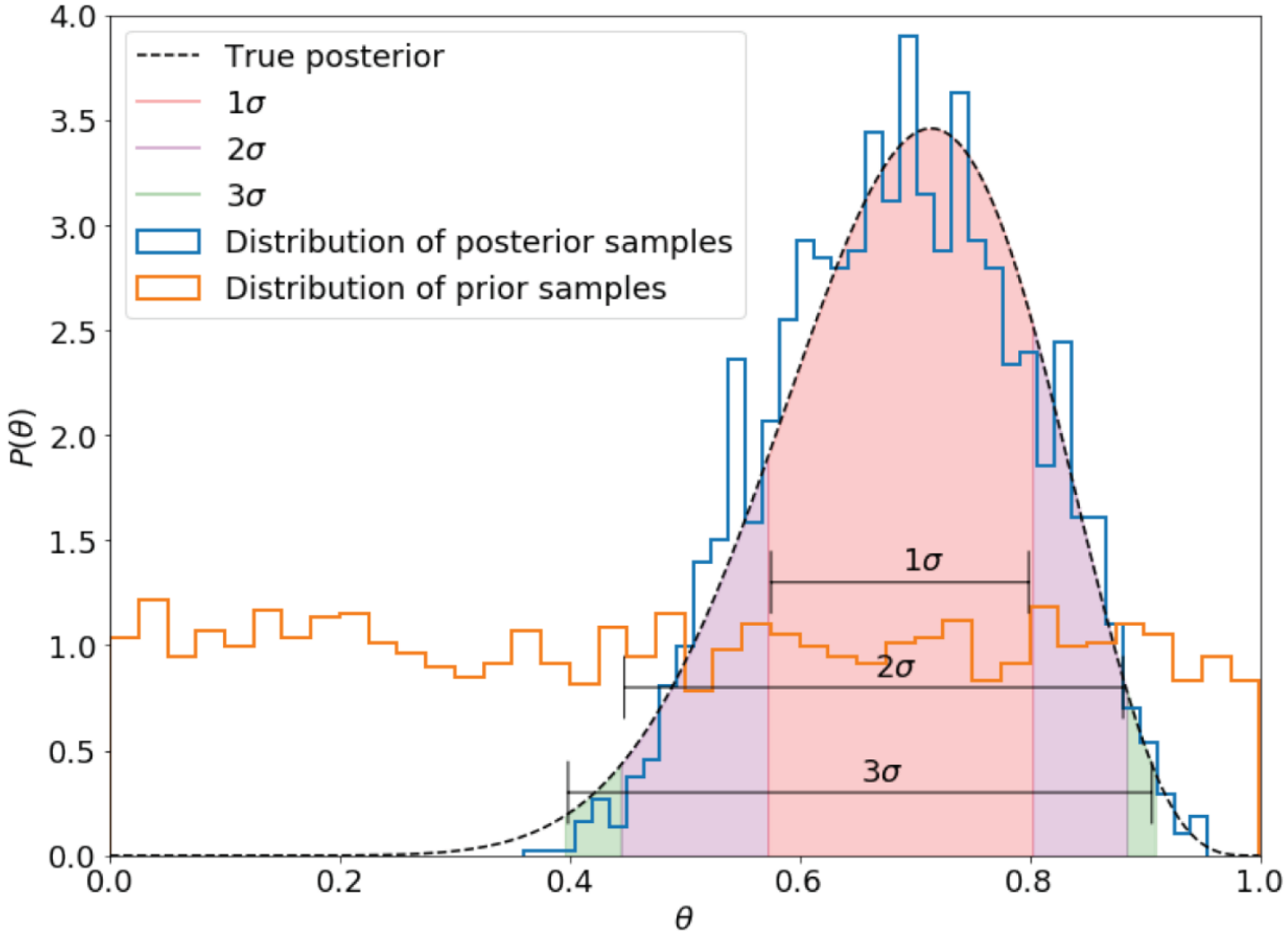
Closed-Form Posterior:

$$p(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y}) = \mathcal{N}\left(\boldsymbol{\theta}|\boldsymbol{m}_N,\boldsymbol{S}_N\right)$$

$$\boldsymbol{S}_N = \left(\boldsymbol{S}_0^{-1} + \sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\right)^{-1}$$

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

Posterior Visualization



Outline

Predictions with Parameter Posterior

Posterior Predictive Distribution:

$$p(y_*|\boldsymbol{x}_*, \mathcal{X}, \mathcal{Y}) = \mathcal{N}\left(\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_N,\ \boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}_*) + \sigma^2\right)$$

Key Observations:

- Predictive mean $\boldsymbol{\phi}^\top(\boldsymbol{x}_*)\boldsymbol{m}_N$ coincides with MAP estimate.
- Predictive variance decomposes into parameter uncertainty ($\boldsymbol{S}_N$) and noise (σ^2).

Example: Distribution Over Functions

Key Insight

Posterior over parameters θ induces distribution over functions

Setup

- True function: $f(x) = 0.5x + 0.3\sin(2\pi x)$
- Prior: $p(\theta) = \mathcal{N}(0, 0.5I)$ where $\phi(x) = [1, x, x^2]$
- Likelihood: $\sigma^2 = 0.1$, Observed data: 5 noisy points in $[0, 1]$

Posterior Samples:

$$\theta_1 \sim \mathcal{N}(m_N, S_N)$$
$$\theta_2 \sim \mathcal{N}(m_N, S_N)$$
$$\vdots$$
$$\theta_k \sim \mathcal{N}(m_N, S_N)$$

Resulting Functions

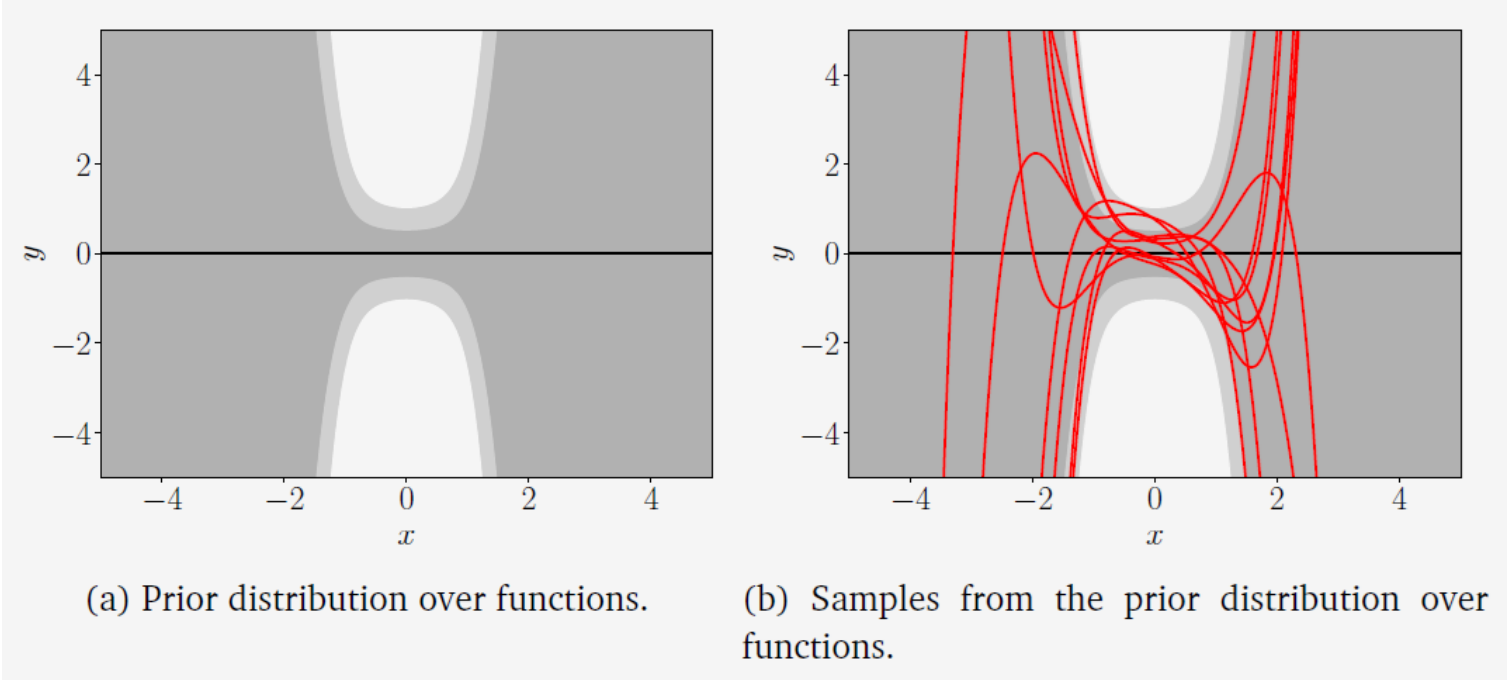
Each sample generates a different function:

$$f_i(x) = \theta_{i0} + \theta_{i1}x + \theta_{i2}x^2$$

Uncertainty bands emerge from parameter covariance S_N

Outline

Prior/Posterior Over Functions



(a) Prior distribution over functions.

(b) Samples from the prior distribution over functions.

Figure: Prior over functions: (a) Mean and confidence bounds; (b) Sampled functions.

Posterior Over Functions (After Training)

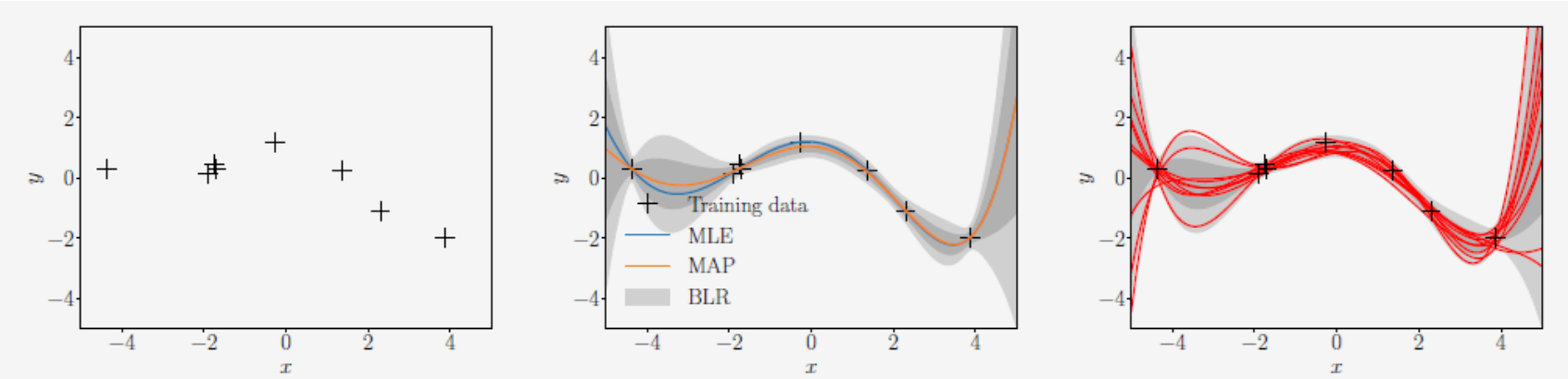


Figure: Posterior over functions: (a) Training data; (b) Predictive confidence bounds; (c) Sampled functions.

Outline

Key Takeaways

- BLR provides **full uncertainty quantification** via posterior distributions.
- Avoids overfitting by averaging over parameters.
- Conjugate Gaussian priors enable closed-form solutions.
- Predictive variance highlights regions of high uncertainty (*e.g.*, extrapolation).

Principles of Machine Learning

Lecture 5: Classification

Sharif University of Technology
Dept. of Aerospace Engineering

March 18, 2025

Table of Contents

Table of Contents

Introduction

- In regression (Chapter 3) we predict a **quantitative** response.
- Many real-world problems involve **qualitative** or **categorical** responses.
- **Classification** is the process of predicting a categorical outcome.

Quantitative vs. Qualitative Responses

- **Quantitative:** e.g. predicting house prices, temperatures, etc.
- **Qualitative:** e.g. predicting eye color (blue, brown, green), disease type, or default status.

Key Point: The methods for regression are not directly applicable to classification tasks.

What is Classification?

- Classification assigns an observation to a **category** or **class**.
- Many classifiers first estimate *class probabilities* before making a final decision.
- This probabilistic approach makes them somewhat similar to regression methods.

Common Classification Techniques

Some widely-used classification methods include:

- **Logistic Regression**
- **Linear Discriminant Analysis (LDA)**
- **K-Nearest Neighbors (KNN)**

More complex methods (discussed in later chapters) include tree-based methods, random forests, boosting, and support vector machines.

Why Not Use Linear Regression?

- Linear regression predicts continuous outputs.
- When the response is categorical, using linear regression can lead to:
  - Predictions that are not valid probabilities.
  - Poor interpretation and performance.

Thus, tailored classification methods are needed.

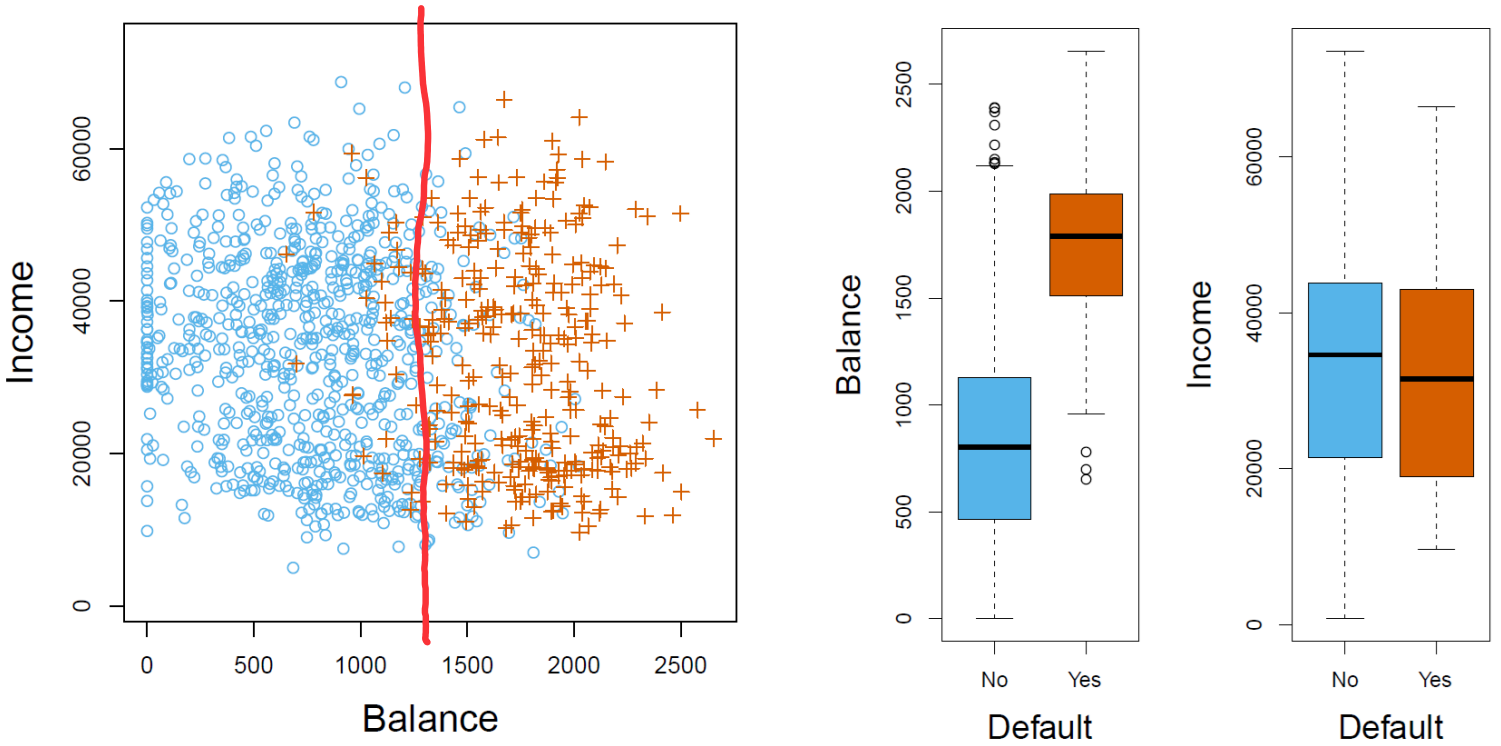
Real-World Examples of Classification

- **Medical Diagnosis:** Identifying which disease a patient has based on symptoms.
- **Fraud Detection:** Classifying whether a transaction is fraudulent.
- **Genetic Studies:** Determining if specific DNA mutations are deleterious.

The Default Dataset Example

- We illustrate classification using the **Default** data set.
- **Goal:** Predict whether an individual will default on a credit card payment.
- **Predictors:** Annual income and monthly credit card balance.

Visualizing the Default Dataset



- The annual incomes and monthly credit card balances of a number of individuals.

Building a Classifier

- Given training observations $(\mathbf{x}_i, y_i)$, our aim is to build a classifier.
- The classifier should perform well on both training and unseen test data.

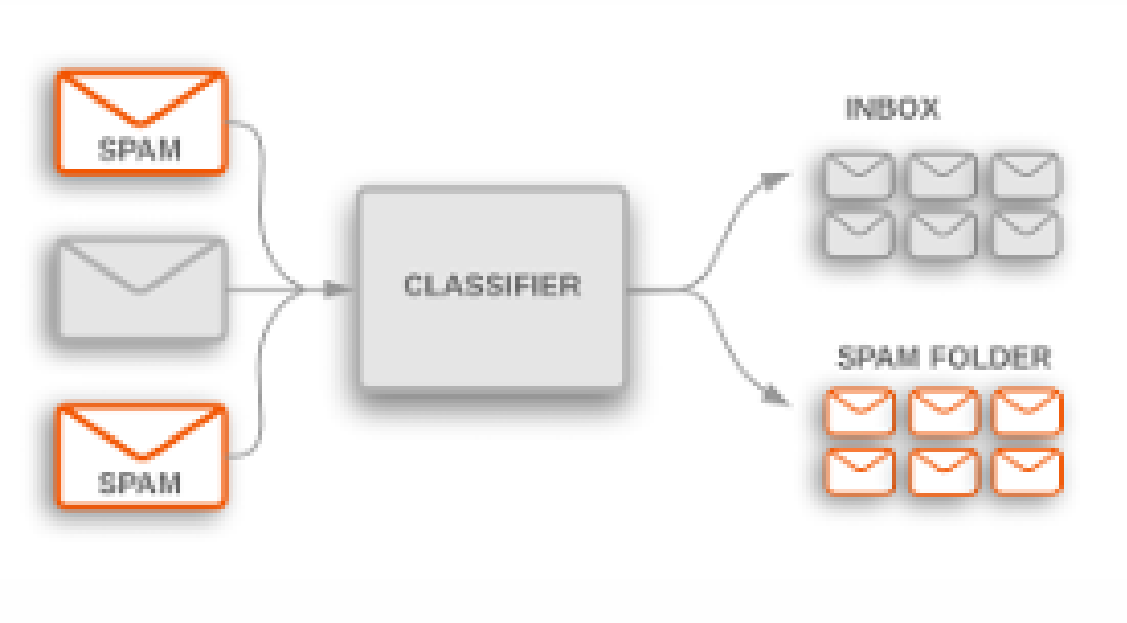


Table of Contents

Introduction

- Linear regression is designed for predicting quantitative responses.
- What happens when we try to use it for qualitative (categorical) outcomes?
- In this section, we discuss its limitations for classification.

A Motivating Example

- Imagine predicting a patient's medical condition in the emergency room.
- Possible diagnoses: **stroke**, **drug overdose**, **epileptic seizure**.

Encoding Categorical Responses

One might consider encoding the diagnoses as:

$$Y = \begin{cases} 1, & \text{if stroke} \\ 2, & \text{if drug overdose} \\ 3, & \text{if epileptic seizure} \end{cases}$$

This encoding allows the use of least squares to predict Y from predictors $X_1, \ldots, X_p$.

The Implicit Ordering Problem

- The numerical coding implies an ordering:

$$\text{stroke} < \text{drug overdose} < \text{epileptic seizure}$$

- It assumes the difference between stroke and drug overdose equals that between drug overdose and epileptic seizure.

Why the Ordering is Problematic

- In practice, there is no inherent ordering or uniform gap between such diagnoses.
- The imposed ordering might bias the model into learning relationships that do not exist.

Alternative Codings Yield Different Models

Consider an alternative coding:

$$Y = \begin{cases} 1, & \text{if epileptic seizure} \\ 2, & \text{if stroke} \\ 3, & \text{if drug overdose} \end{cases}$$

This coding implies a totally different ordering and relationships among the conditions.

Fundamental Consequence

- Different encodings yield fundamentally different linear models.
- Test predictions will vary with the choice of coding.

Natural Ordering vs. Arbitrary Categories

- If the response had a natural ordering (e.g., mild, moderate, severe), a numerical coding like 1, 2, 3 might be reasonable.
- For qualitative responses with more than two levels, there is generally no natural ordering.

Dummy Variable Approach for Binary Responses

For a binary qualitative response, the situation improves.

Example:

$$Y = \begin{cases} 0, & \text{if stroke} \\ 1, & \text{if drug overdose} \end{cases}$$

This is similar to the dummy variable coding.

Using Linear Regression for Binary Responses

- With the 0/1 coding, one could fit a linear regression.
- The decision rule: predict drug overdose if $Y > 0.5$, otherwise stroke.

Using Linear Regression for Binary Responses

- With the 0/1 coding, one could fit a linear regression.
- The decision rule: predict drug overdose if $Y > 0.5$, otherwise stroke.
- For binary responses, even if the coding is reversed, the final predictions remain the same.
- In this special case, the linear regression estimate $X\hat{\beta}$ approximates $\Pr(\text{drug overdose}|X)$.

Limitations for Binary Responses

- Linear regression can produce estimates outside the interval $[0, 1]$.
- Such estimates are difficult to interpret as probabilities.

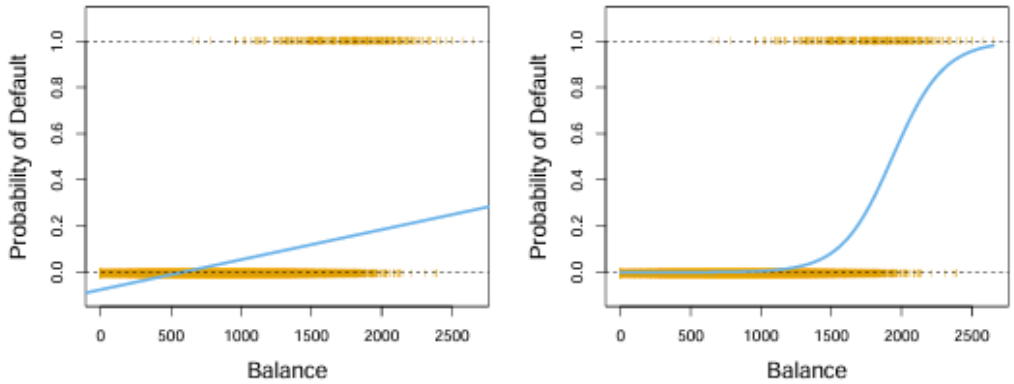
Why Linear Regression Fails for Classification

Linear Regression:

- Predictions outside [0,1]
- Example: Balance \$500 $\rightarrow$ p = -0.02
- Orange ticks show 0/1 outcomes

Logistic Regression:

- Bounded probabilities
- Same \$500 balance $\rightarrow$ p = 0.03



Fundamental Issues with Linear Regression

- **Ordering Imposition:**
  - Artificial spacing between categories
  - No default (0) vs Default (1) $\neq$ 1-unit difference
- **Encoding Ambiguity:**
  - Different codings $\Rightarrow$ different models
  - E.g., (No=0/Yes=1) vs (No=100/Yes=200)
- **Invalid Probabilities:**
  - Predictions like 1.2 or -0.3 are uninterpretable

Table of Contents

Prediction Rule

Threshold-based Prediction

$$\hat{y} = \begin{cases} 0 & \text{if } \widehat{p} < 0.5 \\ 1 & \text{if } \widehat{p} \geq 0.5 \end{cases}$$

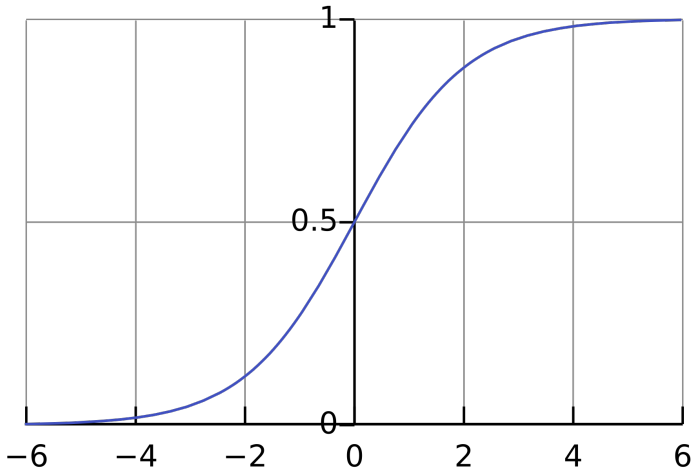
- Decision boundary at $t = \theta^{\mathsf{T}}\mathbf{x} = 0$
- Predicts 1 when logit $t \geq 0$
- Predicts 0 when logit $t < 0$
- Threshold can be adjusted for different risk tolerances

The Logistic Function

Definition

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

- S-shaped (sigmoid) curve
- Bounded between 0 and 1
- Example:
  - Balance \$1000: p = 0.0058
  - Balance \$2000: p = 0.586



Logit and Logistic Function

Key Relationships

- Logit function: $\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$
- Inverse of logistic function: $p = \sigma(t) = \frac{1}{1+e^{-t}}$
- $t = \theta^{\mathsf{T}}\mathbf{x}$ (linear combination)

Example

Practical Interpretation If $\widehat{p} = 0.7$: log-odds $= \log(0.7/0.3) \approx 0.85$