Lecture 04: Statistics

Introduction to Machine Learning [25737]

Sajjad Amini

Sharif University of Technology

Contents

1. Basic Problem
2. Maximum Likelihood Estimation (MLE)
3. Empirical Risk Minimization (ERM)
4. Maximum a Posteriori (MAP) or Regularization
5. Bayesian Model Averaging
6. Approximate Posterior Inference

References

Except explicitly cited, the reference for the material in slides is:

- Murphy, K. P. (2022). *Probabilistic machine learning: an introduction.* MIT press.

Section 1

Basic Problem

Exploring Model Fitting

Model Fitting (Training)

Machine learning generally deals with finding parameterized mapping $f(\cdot;\boldsymbol{\theta})$ (Task) based on dataset $\mathcal{D}$ (Experience). This is known as model fitting (training). Training is generally formulated as:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{\theta})$$

where $\mathcal{L}(\boldsymbol{\theta})$ is known as loss function.

Model Parameters

There are two important points considering model parameters:

- Point estimate $\widehat{\boldsymbol{\theta}}$
- Uncertainty or confidence in the estimate (*Inference*)

Section 2

Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation (MLE)

The MLE for supervised learning is defined as:

$$\widehat{\boldsymbol{\theta}}_{\text{mle}} \triangleq \underset{\boldsymbol{\theta}}{\text{argmax}} \; p(\overbrace{\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^{N}}^{\mathcal{D}} | \boldsymbol{\theta})$$

adding the independency of training example we have:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{y}_n|\boldsymbol{\theta})$$

The above distribution can be reformulated as:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta}) \overbrace{p(\boldsymbol{x}_n|\boldsymbol{\theta})}^{p(\boldsymbol{x}_n)}$$

Thus we can find MLE using the following problem:

$$\widehat{\boldsymbol{\theta}}_{\text{mle}} \triangleq \underset{\boldsymbol{\theta}}{\text{argmax}} \; \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta})$$

Negative Log Likelihood (NLL)

From Slide 7 we have:

$$\widehat{\boldsymbol{\theta}}_{\text{mle}} \triangleq \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta})$$

Log likelihood (LL) is defined as:

$$LL(\boldsymbol{\theta}) \triangleq \log \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta})$$

Adding a negative sign we reach the Negative Log Likelihood (NLL) as:

$$NLL(\boldsymbol{\theta}) \triangleq -\log \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{x}, \boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta})$$

And optimization problem to find $\widehat{\boldsymbol{\theta}}_{mle}$ becomes minimization as:

$$\widehat{\boldsymbol{\theta}}_{\text{mle}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \; NLL(\boldsymbol{\theta})$$

MLE Justification

Equivalent to MAP estimation

Under uniform prior distribution ($p(\boldsymbol{\theta}) \propto 1$), $\widehat{\boldsymbol{\theta}}_{mle} = \widehat{\boldsymbol{\theta}}_{map}$

Equivalence of MAP and MLE Under Uniform Prior

The maximum a Posteriori estimation for model parameters is:

$$\widehat{\boldsymbol{\theta}}_{map} = \underset{\boldsymbol{\theta}}{\text{argmax}} \; p(\boldsymbol{\theta}|\mathcal{D}) = \underset{\boldsymbol{\theta}}{\text{argmax}} \; \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}$$

where $p(\mathcal{D})$ is independent of $\boldsymbol{\theta}$ and we assume uniform prior ($p(\boldsymbol{\theta}) \propto 1$). Thus:

$$\widehat{\boldsymbol{\theta}}_{map} = \underset{\boldsymbol{\theta}}{\text{argmax}} \; p(\mathcal{D}|\boldsymbol{\theta}) = \widehat{\boldsymbol{\theta}}_{mle}$$

Kullback Leibler (KL) divergence

KL divergence is a common function for comparing two distributions p and q defined over a random variable Y. It is formulated as:

- Discrete RV: $D_{\mathrm{KL}}(p\|q) \triangleq \sum_{y\in\mathcal{Y}} p(y)\log\frac{p(y)}{q(y)} = \mathbb{E}_p[\log\frac{p(y)}{q(y)}]$
- Continuous RV: $D_{\mathrm{KL}}(p\|q) \triangleq \int_{y\in\mathcal{Y}} p(y)\log\frac{p(y)}{q(y)}dy = \mathbb{E}_p[\log\frac{p(y)}{q(y)}]$

Kullback Leibler (KL) divergence

- KL divergence is is not a distance measure because:
  - It is not symmetric: $D_{\mathrm{KL}}(p\|q) \neq D_{\mathrm{KL}}(q\|p)$
  - It need not satisfy triangular inequality.
- $D_{\mathrm{KL}}(p\|q) \geq 0$
- $D_{\mathrm{KL}}(p\|q) = 0$ iff $p = q$

Empirical Data Distribution

In a supervised problem over dataset $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}$, the empirical distribution is defined as:

$$p_D(\boldsymbol{x}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_n)$$

$$p_D(\boldsymbol{y}|\boldsymbol{x}) = \begin{cases} \delta(\boldsymbol{y} - \boldsymbol{y}_n) & \text{if } \boldsymbol{x} = \boldsymbol{x}_n, n = 1, \ldots, N \\ \text{ND} & \text{O.W.} \end{cases}$$

$$p_D(\boldsymbol{x}, \boldsymbol{y}) = p_D(\boldsymbol{y}|\boldsymbol{x}) p_D(\boldsymbol{x}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_n) \delta(\boldsymbol{y} - \boldsymbol{y}_n)$$

MLE Justification

Minimizaing the Distance Between Model and Data Distributions

Assume the conditional empirical distribution $p_D(\boldsymbol{y}|\boldsymbol{x})$ and model distribution $p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})$. The expected KL divergence between these distributions over epirical distribution $p_D(\boldsymbol{x})$ is:

$$\mathbb{E}_{p_D(\boldsymbol{x})}[D_{\mathrm{KL}}(p_D(\boldsymbol{y}|\boldsymbol{x})\|p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}))] = \int_{\boldsymbol{x}} p_D(\boldsymbol{x}) \left[\int_{\boldsymbol{y}} p_D(\boldsymbol{y}|\boldsymbol{x}) \log \frac{p_D(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})} d\boldsymbol{y}\right] d\boldsymbol{x}$$

$$= \overbrace{\int_{\boldsymbol{x}}\int_{\boldsymbol{y}} p_D(\boldsymbol{x},\boldsymbol{y}) \log p_D(\boldsymbol{y}|\boldsymbol{x})}^{\text{constant}} - \int_{\boldsymbol{x}}\int_{\boldsymbol{y}} \frac{1}{N}\sum_{n=1}^{N} \delta(\boldsymbol{x}-\boldsymbol{x}_n)\delta(\boldsymbol{y}-\boldsymbol{y}_n) \log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})$$

$$= \text{constant} - \frac{1}{N}\sum_{n=1}^{N} \log p(\boldsymbol{y}_n|\boldsymbol{x}_n,\boldsymbol{\theta}) = \text{constant} + NLL(\boldsymbol{\theta})$$

Thus we have:

$$\widehat{\boldsymbol{\theta}}_{mle} = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{p_D(\boldsymbol{x})}[D_{\mathrm{KL}}(p_D(\boldsymbol{y}|\boldsymbol{x})\|p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}))]$$

MLE Example

MLE for Bernolli Distribution

Suppose:

- $Y \sim Ber(\theta)$ representing coin toss ($Y = 1$ represents head)
- $\theta = p(Y = 1)$
- $\mathcal{D} = \{y_1, \ldots, y_N\}$

Compute $\widehat{\theta}_{mle}$.

MLE Example

MLE for Bernolli Distribution

Suppose:

- $Y \sim Ber(\theta)$ representing coin toss ($Y = 1$ represents head)
- $\theta = p(Y = 1)$
- $\mathcal{D} = \{y_1, \ldots, y_N\}$

Compute $\widehat{\theta}_{mle}$.

Solution

$$NNL(\theta) = -\log \prod_{n=1}^{N} p(y_n|\theta) = -\log \prod_{n=1}^{N} \theta^{\mathbb{I}(y_n=1)}(1-\theta)^{\mathbb{I}(y_n=0)}$$

$$= -\sum_{n=1}^{N} \mathbb{I}(y_n = 1)\log\theta + \mathbb{I}(y_n = 0)\log(1-\theta) = -[N_1 \log\theta + N_0 \log(1-\theta)]$$

where $N_1 = \sum_{n=1}^{N} \mathbb{I}(y_n = 1)$ (number of heads) and $N_0 = \sum_{n=1}^{N} \mathbb{I}(y_n = 0)$ (number of tails). N_1 and N_2 are called the *Sufficient Statistics* of the data, since they summarize everything we need to know about $\mathcal{D}$. $N = N_1 + N_2$ is called the *Sample Size*. $\widehat{\theta}_{mle}$ cab be found as:

$$\frac{d}{d\theta} NLL(\theta) = 0 \Rightarrow \widehat{\theta}_{mle} = \frac{N_1}{N_1 + N_0} (\text{ Empirical fraction of heads})$$

MLE Example

MLE for Gaussian Distribution

Suppose:

- $Y \sim \mathcal{N}(\mu, \sigma^2)$
- $\boldsymbol{\theta} = (\mu, \sigma^2)$
- $\mathcal{D} = \{y_1, \ldots, y_N\}$

Compute $\widehat{\theta}_{mle} = \{\widehat{\mu}_{mle}, \widehat{\sigma}^2_{mle}\}$.

MLE Example

MLE for Gaussian Distribution

Suppose:

- $Y \sim \mathcal{N}(\mu, \sigma^2)$
- $\boldsymbol{\theta} = (\mu, \sigma^2)$
- $\mathcal{D} = \{y_1, \ldots, y_N\}$

Compute $\widehat{\theta}_{mle} = \{\widehat{\mu}_{mle}, \widehat{\sigma}^2_{mle}\}$.

Solution

$$NLL(\mu, \sigma^2) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mu)^2 + \frac{N}{2} \log(2\pi\sigma^2)$$

$\bar{y} = \frac{1}{N} \sum_{n=1}^{N} y_n$ and $s^2 = \frac{1}{N} \sum_{n=1}^{N} y_n^2$ are called the *Sufficient Statistics* of the data, since they summarize everything we need to know about $\mathcal{D}$ to calculate $\widehat{\mu}_{mle}$ and $\widehat{\sigma}^2_{mle}$ as:

$$\begin{cases} \frac{d}{d\mu} NLL(\mu, \sigma^2) = 0 \\ \frac{d}{d\sigma^2} NLL(\mu, \sigma^2) = 0 \end{cases} \Rightarrow \begin{cases} \widehat{\mu}_{mle} = \frac{1}{N} \sum_{n=1}^{N} y_n = \bar{y} \\ \widehat{\sigma}^2_{mle} = \frac{1}{N} \sum_{n=1}^{N} (y_n - \widehat{\mu})^2 = s^2 - \bar{y}^2 \end{cases}$$

MLE Example

MLE for MVN

Suppose:

- $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- $\mathcal{D} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$

Compute $\widehat{\boldsymbol{\theta}}_{mle} = \{\widehat{\boldsymbol{\mu}}_{mle}, \widehat{\boldsymbol{\Sigma}}_{mle}\}$.

MLE Example

MLE for MVN

Suppose:

- $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- $\mathcal{D} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$

Compute $\widehat{\boldsymbol{\theta}}_{mle} = \{\widehat{\boldsymbol{\mu}}_{mle}, \widehat{\boldsymbol{\Sigma}}_{mle}\}$.

Solution

$$LL(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{N}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2} \sum_{n=1}^{N} (\boldsymbol{y}_n - \boldsymbol{\mu})^T \boldsymbol{\Lambda} (\boldsymbol{y}_n - \boldsymbol{\mu})$$

$\bar{\boldsymbol{y}} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{y}_n$ and $\boldsymbol{S} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{y}_n \boldsymbol{y}_n^T$ are called the *Sufficient Statistics* of the data, since they summarize everything we need to know about $\mathcal{D}$ to calculate $\widehat{\boldsymbol{\mu}}_{mle}$ and $\widehat{\boldsymbol{\Sigma}}_{mle}$ as:

$$\begin{cases} \frac{\partial}{\partial \boldsymbol{\mu}} NLL(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = 0 \\ \frac{\partial}{\partial \boldsymbol{\Sigma}} NLL(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = 0 \end{cases} \Rightarrow \begin{cases} \widehat{\boldsymbol{\mu}}_{mle} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{y}_n = \bar{\boldsymbol{y}} \\ \widehat{\boldsymbol{\Sigma}}_{mle} = \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{y}_n - \bar{\boldsymbol{y}})(\boldsymbol{y}_n - \bar{\boldsymbol{y}})^T = \boldsymbol{S} - \bar{\boldsymbol{y}} \bar{\boldsymbol{y}}^T \end{cases}$$

Section 3

Empirical Risk Minimization (ERM)

Empirical Risk Minimization (ERM)

ERM

Remember MLE where we have the following problem:

$$\widehat{\boldsymbol{\theta}}_{mle} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{n=1}^{N} \overbrace{-\log p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta})}^{l(\boldsymbol{y}_n;\boldsymbol{x}_n,\boldsymbol{\theta})}$$

We can generalize this result by replacing conditional log loss with any other loss, to get:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} l(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{\theta})$$

The above is known as Empirical Risk Minimization (ERM). It is loss expectation with respect to empirical distribution.

Sample Loss Functions

Assume:

- A probabilistic binary classifier as: $p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \sigma(y\eta) = \frac{1}{1+e^{-y\eta}}$ where $\eta = f(\boldsymbol{x}; \boldsymbol{\theta})$ is logg odds and $y \in \{-1, +1\}$.

We can define different loss functions as:

Name	$l(y; \boldsymbol{x}_n, \boldsymbol{\theta})$
Misclassification	$\mathbb{I}(y\eta < 0)$
NLL	$-\log_2 p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \log_2(1 + e^{-y\eta})$
Hing loss	$\max(0, 1 - y\eta) = (1 - y\eta)_+$
Exp loss	$e^{y\eta}$

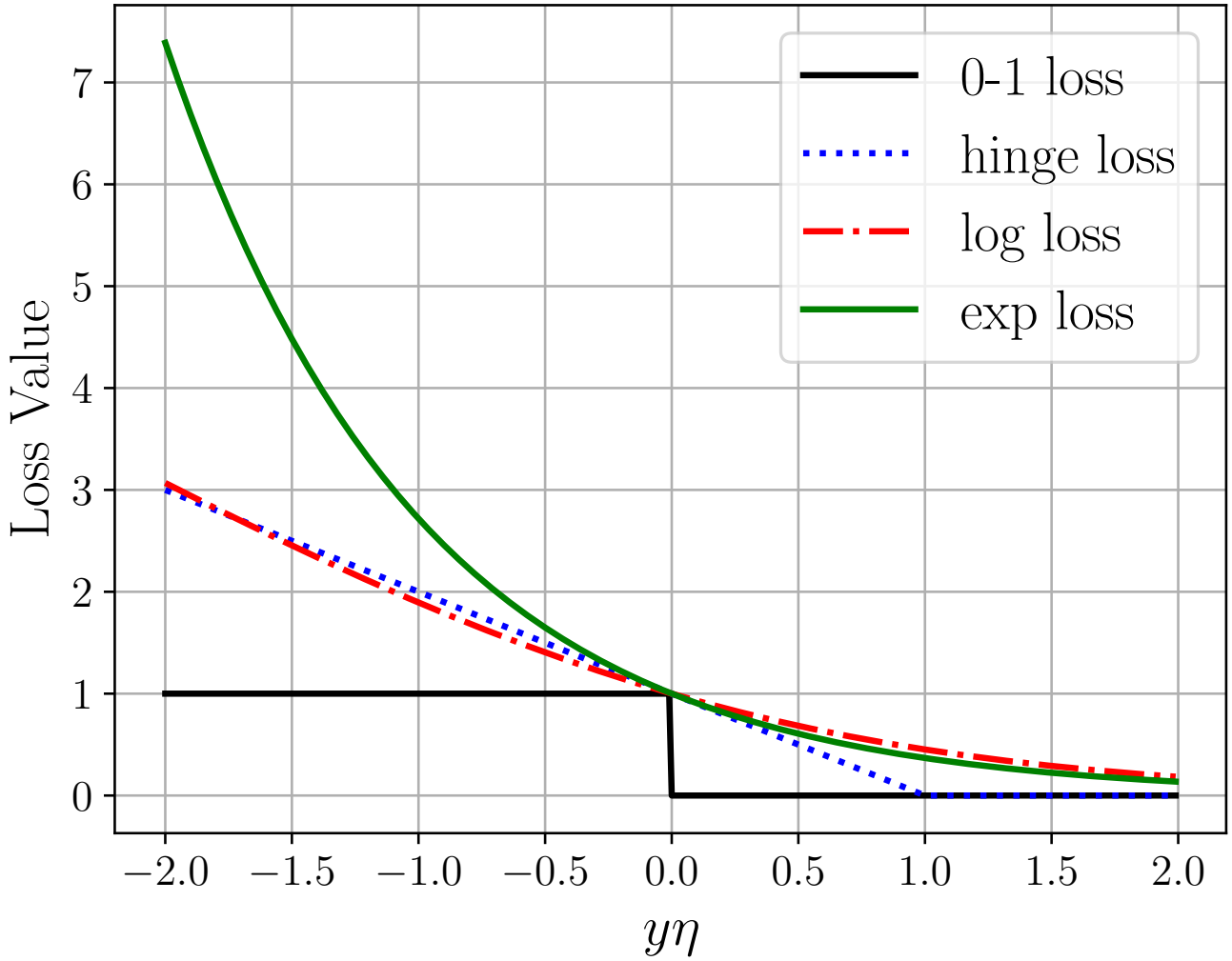


Figure: Loss functions for binary classification

Section 4

Maximum a Posteriori (MAP) or Regularization

Problem with MLE

Overfitting in MLE

Suppose the example of coin tossing with $N = 3$ where we observe 3 heads. Thus we have:

$$\widehat{\theta}_{mle} = \frac{N_1}{N_1 + N_0} = 1$$

In this case, overfitting has occurred.

Regularization

Regularization is the process of designing and adding a penalty term to NLL (or empirical risk) so as to control overfitting. Thus we have:

$$\mathcal{L}(\boldsymbol{\theta}; \lambda) = \left[\frac{1}{N} \sum_{n=1}^{N} l(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{\theta}) \right] + \lambda C(\boldsymbol{\theta})$$

where:

- $\lambda \geq 0$ is the regularization parameter
- $C(\boldsymbol{\theta})$ is some form of complexity penalty

From Regularization to MAP

Assume $C(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta})$ and $\lambda = 1$. Then:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}; 1) &= -\left[\sum_{n=1}^{N} \log p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right] \\ &= -\log p(\boldsymbol{\theta}|\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^{N}) + \text{const}\end{aligned}$$

Thus minimizing the $\mathcal{L}(\boldsymbol{\theta}; 1)$ is equivalent to maximizing the posterior and we have:

$$\widehat{\boldsymbol{\theta}}_{map} = \underset{\boldsymbol{\theta}}{\text{argmin}} \ \mathcal{L}(\boldsymbol{\theta}; 1)$$

MAP Example

MAP for Bernolli Distribution

Suppose:

- $Y \sim Ber(\theta)$ representing coin toss ($Y = 1$ represents head)
- $\theta = p(Y = 1)$
- $\mathcal{D} = \{y_1, \ldots, y_N\}$
- $p(\theta) = \text{Beta}(\theta|a, b) = \frac{\theta^{a-1}(1-\theta)^{b-1}}{\text{B}(a,b)}$ (Prior Distribution)

Compute $\widehat{\theta}_{map}$.

MAP Example

MAP for Bernolli Distribution

Suppose:

- $Y \sim Ber(\theta)$ representing coin toss ($Y = 1$ represents head)
- $\theta = p(Y = 1)$
- $\mathcal{D} = \{y_1, \ldots, y_N\}$
- $p(\theta) = \text{Beta}(\theta|a, b) = \frac{\theta^{a-1}(1-\theta)^{b-1}}{\text{B}(a,b)}$ (Prior Distribution)

Compute $\widehat{\theta}_{map}$.

Solution

$$\widehat{\theta}_{map} = \underset{\theta}{\text{argmin}} - \log \prod_{n=1}^{3} p(y_n|\theta) - \log p(\theta)$$

$$= (N_1 + a - 1)\log(\theta) + (N_0 + b - a)\log(1 - \theta) = \frac{N_1 + a - 1}{N_1 + N_0 + a + b - 2}$$

MAP Example

MAP for Bernolli Distribution (Continue)

$$\widehat{\theta}_{map} = \frac{N_1 + a - 1}{N_1 + N_0 + a + b - 2}$$

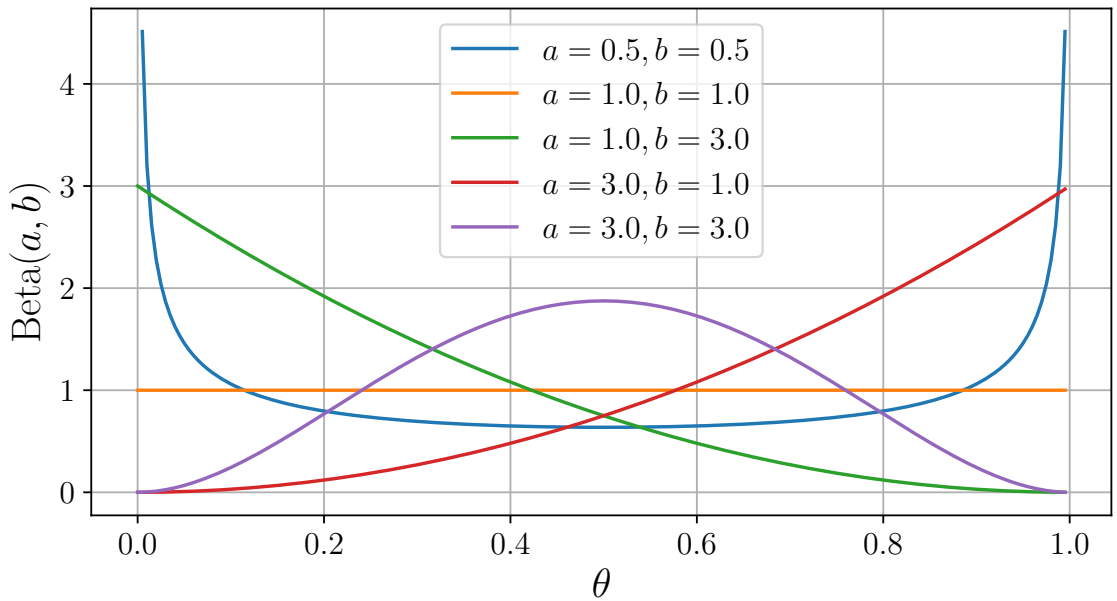


Figure: Probability density function for Beta distribution

Regularization Parameter

Challenge in Selecting λ

As we see before the regularized loss is defined as:

$$\mathcal{L}(\boldsymbol{\theta}; \lambda) = \left[\frac{1}{N} \sum_{n=1}^{N} l(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{\theta}) \right] + \lambda C(\boldsymbol{\theta})$$

But how to Select λ:

- Large value of $\lambda \Rightarrow$ Staying near prior (*Underfitting*)
- Small value of $\lambda \Rightarrow$ Focus on minimizing empirical risk (*Overfitting*)

Selecting Regularization Parameter

Using Validation Set

Define $R_\lambda(\boldsymbol{\theta}, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} l(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{\theta}) + \lambda C(\boldsymbol{\theta})$. Then we select λ as:

- Partition data into two disjoint set $\mathcal{D}_{\text{train}}$ (training set) and $\mathcal{D}_{\text{valid}}$ (validation or development set). Usually we put 80% for training and 20% for validation
- For each value of λ compute: $\widehat{\boldsymbol{\theta}}_\lambda(\mathcal{D}_{\text{train}}) = \text{argmin}_{\boldsymbol{\theta}}\, R_\lambda(\boldsymbol{\theta}, \mathcal{D}_{\text{train}})$
- Compute the validation risk: $R_\lambda^{\text{val}} = R_0(\widehat{\boldsymbol{\theta}}_\lambda(\mathcal{D}_{\text{train}}), \mathcal{D}_{\text{valid}})$
- Select: $\lambda^\star = \text{argmin}_\lambda \;\; R_\lambda^{\text{val}}$

Fit the model to entire dataset: $\widehat{\boldsymbol{\theta}}^\star = \text{argmin}_{\boldsymbol{\theta}}\, R_{\lambda^\star}(\boldsymbol{\theta}, \mathcal{D})$

Small Size Dataset

If the size of dataset is small, leaving aside 20% for a validation set can result in an unreliable estimate of the model parameters.

Selecting Regularization Parameter

Using Cross-Validation

Define $R_\lambda(\boldsymbol{\theta}, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} l(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{\theta}) + \lambda C(\boldsymbol{\theta})$. Then we select λ as:

- Split data into K folds.
- For each fold $k \in \{1, \ldots, K\}$, we train the model on all the folds but the k-th, and test on the k-th. So we calculate:

$$\text{Cross-validated risk: } R_\lambda^{\text{CV}} \triangleq \sum_{k=1}^{K} R_0(\widehat{\boldsymbol{\theta}}_\lambda(\mathcal{D}_{-k}), \mathcal{D}_k)$$

- Select: $\lambda^\star = R_\lambda^{\text{CV}}$

Fit the model to entire dataset: $\widehat{\boldsymbol{\theta}}^\star = \text{argmin}_{\boldsymbol{\theta}} R_{\lambda^\star}(\boldsymbol{\theta}, \mathcal{D})$

Avoid Overfitting

Early Stopping

Model parameters ($\boldsymbol{\theta}$) are learned in iterative optimization algorithm. In this method, the optimization is stopped as signs of overfitting are observed.

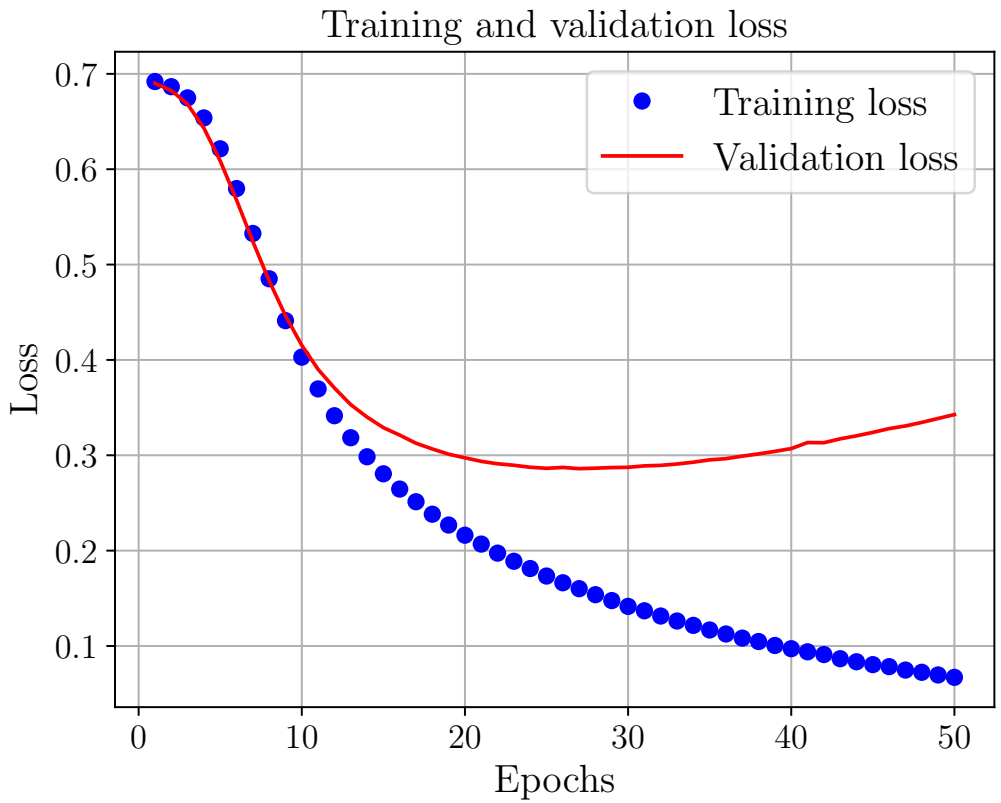


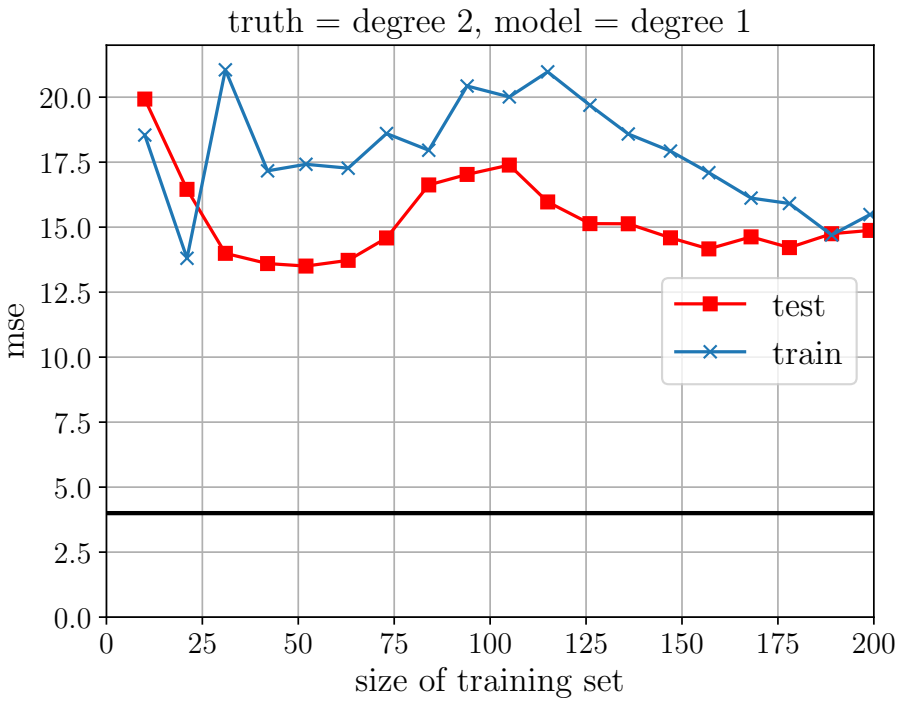
Figure: Tracking overfitting through iterations

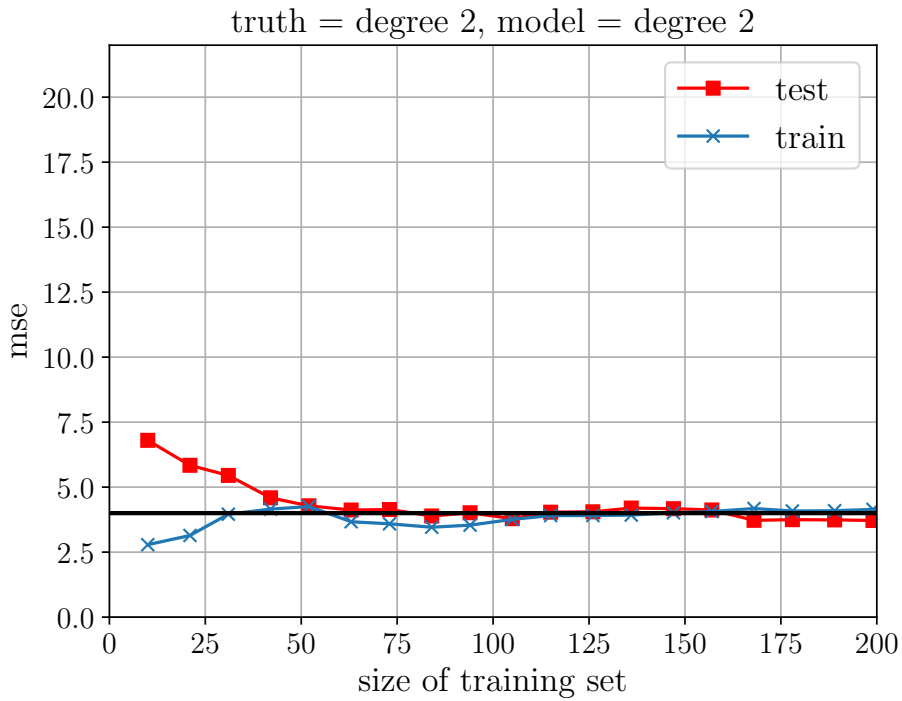
Avoid Overfitting

Using More Data

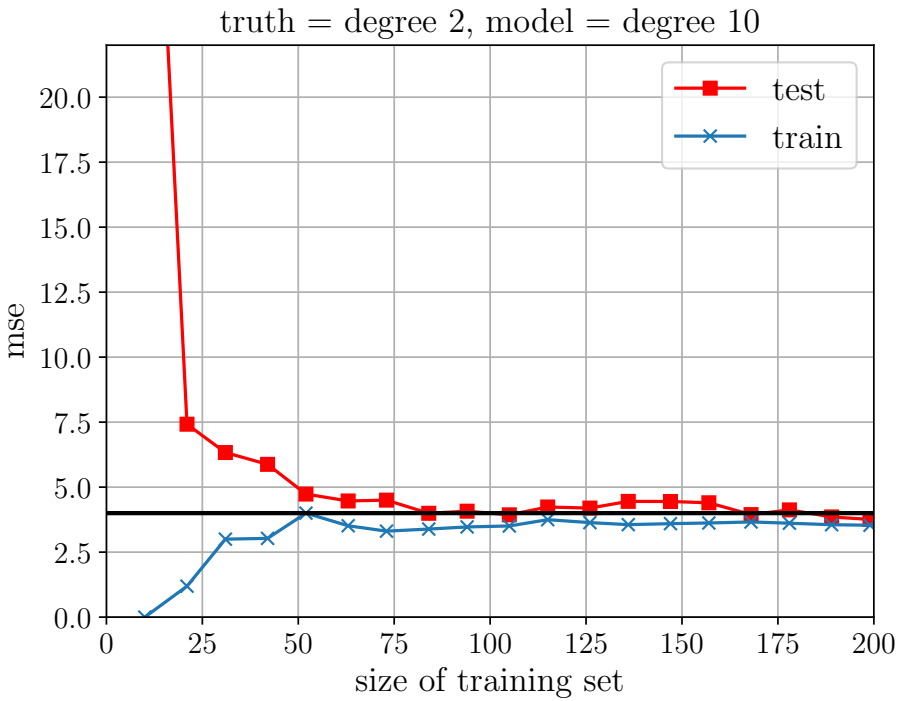
As the amount of data increases, the chance of overfitting (for a model of fixed complexity) decreases (assuming the data contains suitably informative examples, and is not too redundant).

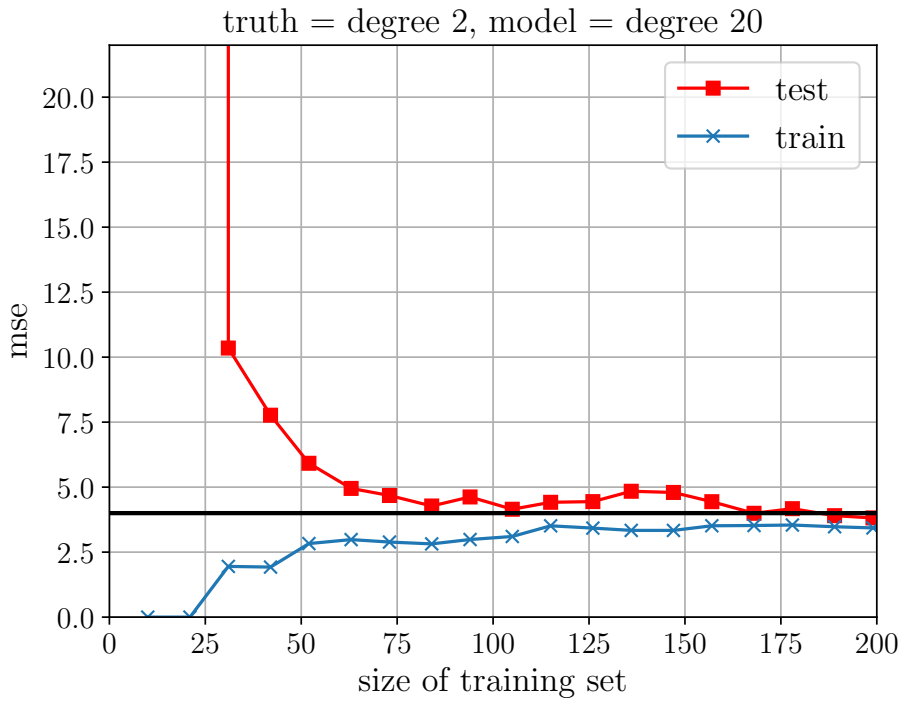
Avoid Overfitting





Avoid Overfitting





Avoid Overfitting

Marginal Likelihood

Using Bayes rule, we can compute the posterior over parameters $p(\boldsymbol{\theta}|\mathcal{D})$ as:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{p(\mathcal{D})} = \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta}')p(\mathcal{D}|\boldsymbol{\theta}')d\boldsymbol{\theta}'}$$

$p(\mathcal{D})$ in the denominator is called *marginal likelihood* since it is computed by marginalizing over the unknown parameters $\boldsymbol{\theta}$. This can be interpreted as:

$$p(\mathcal{D}) = \mathbb{E}_{p(\boldsymbol{\theta})}[p(\mathcal{D}|\boldsymbol{\theta})]$$

Bayes Model Averaging (BMA)

In Bayes Model Averaging, we compute the *Posterior Predictive Distribution* over outputs given inputs by marginalizing out $\boldsymbol{\theta}$ parameters as:

$$p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}) = \int p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$

Section 5

Bayesian Model Averaging

Bayesian Model Averaging

Challenge

The possibility to compute posterior probability is one of the main challenges for BMA. The solution is to use conjugate prior to likelihood function.

Bernoulli Likelihood - Beta Prior

Assume dataset samples are independent and identically distributed and comes from Bernoulli distribution where $\theta = P(Y = 1)$. Then:

- The likelihood is: $p(\mathcal{D}|\theta) = \prod_{n=1}^{N} \theta^{N_1}(1-\theta)^{N_0}$
  - $N_1 = \sum_{n=1}^{N} \mathbb{I}(y_n = 1)$ and $N_0 = \sum_{n=1}^{N} \mathbb{I}(y_n = 0)$
- Beta is conjugate prior to Bernoulli likelihood thus: $p(\theta) \propto \theta^{\breve{\alpha}-1}(1-\theta)^{\breve{\beta}-1} = \text{Beta}(\theta|\breve{\alpha}, \breve{\beta})$

The posterior can be calculated as:

$$p(\theta|\mathcal{D}) = \text{Beta}(\theta|\hat{\alpha}, \hat{\beta}), \begin{cases} \hat{\alpha} \triangleq \breve{\alpha} + N_1 \\ \hat{\beta} \triangleq \breve{\beta} + N_0 \end{cases}$$

Bernoulli Likelihood - Beta Prior (Continue)

- The parameters of the prior are called *hyper-parameters*
- Hyper-parameters play a role analogous to the sufficient statistics (N_1 and N_2); they are therefore often called *pseudo counts*.
- The strength of the prior is controlled by $\breve{N} = \breve{\alpha} + \breve{\beta}$; this is called the *equivalent sample size* (analogous to $N = N_0 + N_1$).

Bayesian Model Averaging

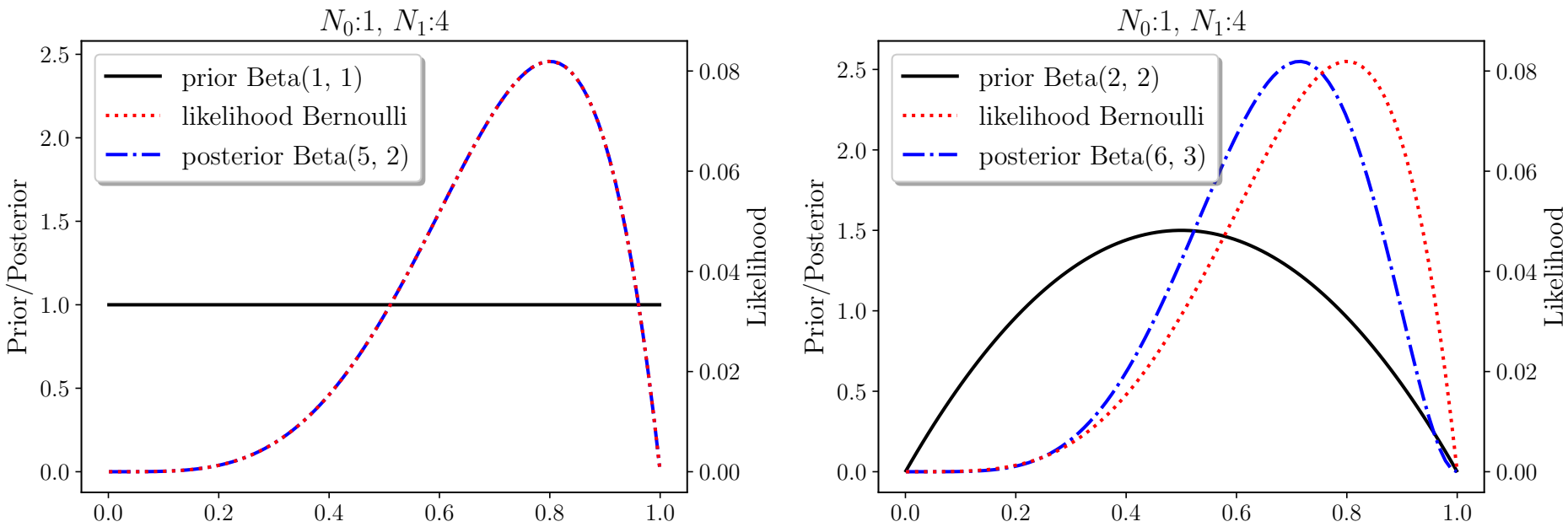


Figure: Uniform and non-Uniform prior distribution for small dataset size

Bayesian Model Averaging

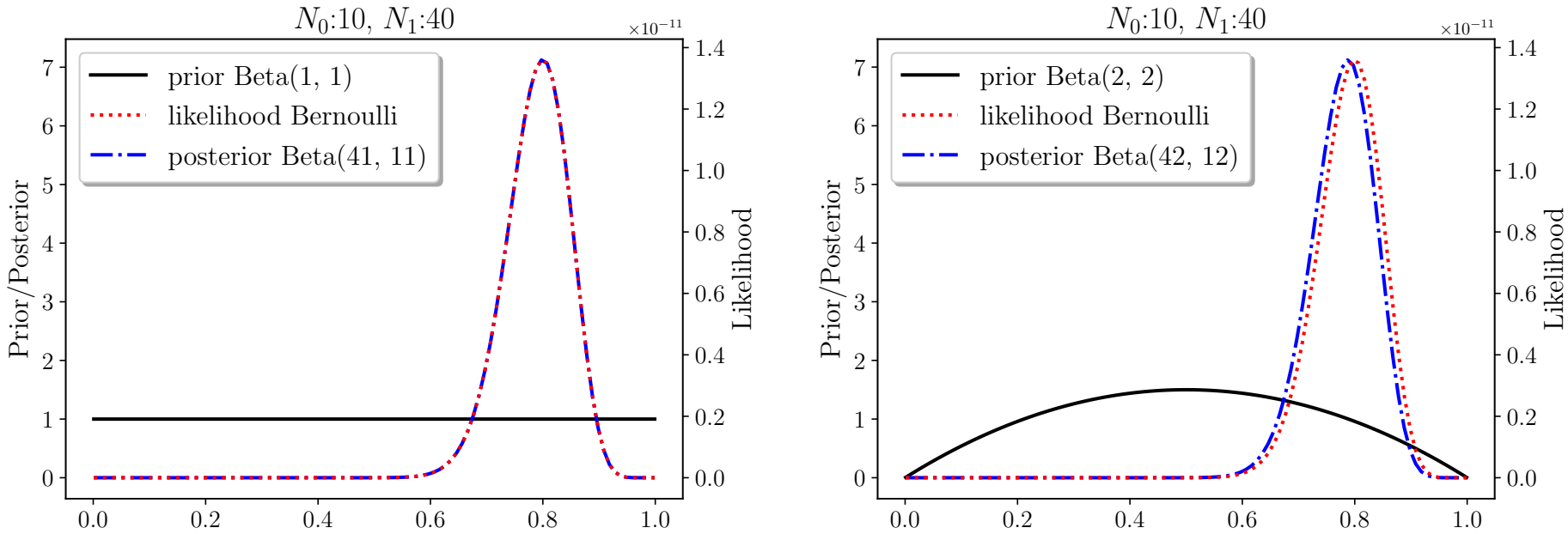


Figure: Uniform and non-Uniform prior distribution for large dataset size

Bernoulli Likelihood - Beta Prior (Continue)

We can use different point estimates (*Plug-in approximation*) as:

$$\widehat{\boldsymbol{\theta}}_{\text{map}} = \frac{\breve{\alpha} + N_1 - 1}{\breve{\alpha} + N_1 - 1 + \breve{\beta} + N_0 - 1}$$

$$\widehat{\boldsymbol{\theta}}_{\text{mle}} = \frac{N_1}{N_1 + N_0}$$

$$\bar{\boldsymbol{\theta}} \triangleq \mathbb{E}[\boldsymbol{\theta}|\mathcal{D}] = \frac{\hat{\alpha}}{\hat{\alpha} + \hat{\beta}} = \lambda \overbrace{\frac{\breve{\alpha}}{\breve{N}}}^{\bar{\theta}_p} + (1-\lambda) \overbrace{\frac{N_1}{N}}^{\widehat{\theta}_{\text{mle}}}, \lambda = \frac{\breve{N}}{N + \breve{N}}$$

The posterior variance can show the uncertainty in our estimate and can be calculated as:

$$\mathbb{V}[\theta|\mathcal{D}] = \frac{\hat{\alpha}\hat{\beta}}{(\hat{\alpha} + \hat{\beta})^2(\hat{\alpha} + \hat{\beta} + 1)}$$

If $N \gg \breve{\alpha} + \breve{\beta}$, then the posterior variance can be simplified to:

$$\mathbb{V}[\theta|\mathcal{D}] \approx \frac{\widehat{\theta}_{\text{mle}}(1 - \widehat{\theta}_{\text{mle}})}{N}$$

Bernoulli Likelihood - Beta Prior (Continue)

Using posterior predictive distribution we have:

$$p(y=1|\mathcal{D}) = \int_0^1 p(y=1|\theta)p(\theta|\mathcal{D})d\theta$$
$$= \int_0^1 \theta Beta(\theta|\hat{\alpha},\hat{\beta})d\theta = \mathbb{E}[\theta|\mathcal{D}] = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}}$$

Now compare these two cases:

- Plug-in approximation with $p(\theta) = \text{Beta}(\theta|2,2)$ then $p(y=1|\widehat{\theta}_{\text{map}}) = \frac{N_1+1}{N_1+N_2+2}$
- Posterior predictive distribution with $p(\theta) = \text{Beta}(\theta|1,1)$ then $p(y=1) = \frac{N_1+1}{N_1+N_2+2}$

Section 6

Approximate Posterior Inference

Grid Approximation

Basis: Discretizing parameter space

- Partitioning the parameters space into finite set of possibilities, denoted $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$
- Approximate the posterior using brute-force enumeration:

$$p(\boldsymbol{\theta} = \boldsymbol{\theta}_k|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k)}{p(\mathcal{D})} \approx \frac{p(\mathcal{D}|\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k)}{\sum_{k'=1}^{K} p(\mathcal{D}|\boldsymbol{\theta}_{k'})p(\boldsymbol{\theta}_{k'})}$$

Notes:

- Not scalable with respect to parameter vector dimension (Exponential grow)

Quadratic (Laplace) Approximation

Basis: Approximating posterior using MVN

- Rewrite the posterior as:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{Z} e^{-\epsilon(\boldsymbol{\theta})}, \quad \begin{cases} \epsilon(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta}, \mathcal{D}) \\ Z = p(\mathcal{D}) \end{cases}$$

- Approximate $\epsilon(\boldsymbol{\theta})$ around its mode ($\widehat{\boldsymbol{\theta}}_{\text{map}}$) using Taylor expansion:

$$\epsilon(\boldsymbol{\theta}) \approx \epsilon(\widehat{\boldsymbol{\theta}}_{\text{map}}) + (\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{\text{map}})^T \boldsymbol{g} + \frac{1}{2}(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{\text{map}})^T \boldsymbol{H} (\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{\text{map}})$$

- $\boldsymbol{g}(\widehat{\boldsymbol{\theta}}_{\text{map}}) = \mathbf{0}$ and we can compute $\boldsymbol{H}$, thus: $\widehat{p}(\boldsymbol{\theta}|\mathcal{D}) = \mathcal{N}(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}_{\text{map}}, \boldsymbol{H}^{-1})$

Notes:

- $\boldsymbol{H}$ is assumed to be diagonal
- Not suitable for skewed posterior
- Not suitable for constrained parameters

Approximate Posterior Inference

Variational Approximation

Basis:

- Assuming approximate posterior distribution comes from family $\mathcal{Q}$, denoted q
- Find $q^\star$ as: $q^\star = \text{argmin}_{q \in \mathcal{Q}} \; D(q(\boldsymbol{\theta}), p(\boldsymbol{\theta}|\mathcal{D}))$ where:
  - D is a discrepancy measure such as KL divergence

Notes:

- The approximation can be biased if due to the limitation $q \in \mathcal{Q}$

Markov Chain Monte Carlo (MCMC)

Basis: Generating samples from posterior

- Generate samples $\boldsymbol{\theta}^s \sim p(\boldsymbol{\theta}|\mathcal{D})$ efficiently without having to evaluate normaization constant $p(\mathcal{D})$
- Evaluate posterior by:

$$q(\boldsymbol{\theta}) \approx \frac{1}{S} \sum_{s=1}^{S} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^s)$$

Notes:

- Low convergence speed