Machine Learning in Communications
Lecture 3: Statistical Estimation and its Role in Machine Learning

Harpreet S. Dhillon

Wireless@VT, Department of Electrical & Computer Engineering
Virginia Tech, Blacksburg, VA

https://www.dhillon.ece.vt.edu
hdhillon@vt.edu

JTG/IEEE Information Theory Society Summer School
IIT Kanpur

Lecture Objectives

The main objective of this and the next lecture is to explore connections between estimation theory and popular machine learning algorithms. In this specific lecture, we will cover:

- Some basics of statistical estimation.
- Interpretation of least squares regression as a maximum likelihood estimator.
- Interpretation of regularized linear regression as a MAP estimator.
- Basis expansion or feature augmentation to perform polynomial regression.

Let us Start with a Coin Toss

- Consider a biased coin with $P(H) = \theta^*$, where θ^* is unknown to you.
- You toss the coin n times and get a specific sequence of H and T, say $H, T, T, \ldots H$.
- Let the fraction of heads in this sequence be $\hat{\theta}$.
- Question: Can $\hat{\theta}$ be very different from θ^*?
- Answer: *Possible* but not *probable* if n is large.
- Hoeffding's inequality provides a more formal answer:

$$P(|\hat{\theta} - \theta^*| > \epsilon) \leq 2 \exp\left(-2\epsilon^2 n\right).$$

  In words, $\hat{\theta}$ is probably close to θ^* if n is large.
- Take away: One can learn unknown out-of-sample (or true) θ^* from in-sample $\hat{\theta}$. Therefore, endowing a distribution on the dataset is necessary to ensure generalization to unseen data.
- This is explored more formally in Probably Approximately Correct (PAC) learning (part of "Computational Learning Theory").

Statistical Setting

- Think of features X and labels Y as random variables.
- All test and training samples $(\boldsymbol{x}, y) \sim p(\boldsymbol{x}, y)$.
  - For the discrete case, $p(\boldsymbol{x}, y)$ would represent pmf.
  - For the continuous case, $p(\boldsymbol{x}, y)$ would represent pdf.
  - It is advisable to think in terms of the discrete case to understand key concepts.
- We *assume* that $(\boldsymbol{x}_i, y_i)$ are sampled i.i.d. from $p(\boldsymbol{x}, y)$.
- This reduces the learning problem to learning (or estimating) the parameters of this (unknown) distribution.
- Not surprisingly, estimation theory is going to play a role in this development, which is the main topic of this lecture.

Bayes' Rule

- Let X and Y be two random variables with joint distribution $p(x, y)$. Bayes's rule states that:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}.$$

- Posterior: $p(y|x)$. Knowledge of the outputs after the data has been revealed to you.
- Prior: $p(y)$. Your belief about the outputs before the data is revealed to you. This often comes from domain knowledge.
- Likelihood: $p(x|y)$. Likelihood of observing a specific value of input for a given value of the output. For instance, $p(x = j|y = i)$ quantifies how likely is $x = j$ when you know the output is $y = i$.
- Normalization term: $p(x)$. Not very useful for this development since it is just a normalization term. We will often just say that $p(y|x) \propto p(x|y)p(y)$.

Statistical Estimation

- Frequentist approach: Maximum likelihood estimation.
- Bayesian approach: Maximum a posteriori estimation or Bayesian estimation.
- We will describe these approaches through the simple coin toss example that we introduced earlier in the lecture.
- Remember that we are eventually interested in $p(y|x)$. For now, we will just focus on a single random variable. All the ideas will be applicable to the conditional distribution as well.

Statistical Estimation

- You toss a coin few times and observe the following outcomes: $\mathbf{y} = \{T, H, T, H, T, T\}$. Think of this as your dataset.
- Our goal is to estimate the underlying distribution.
- We make the following reasonable observations:
  - The tosses are independent.
  - The outcome is binary.
- This means each coin toss can be thought of as an outcome of a Bernoulli trial; $y_i \sim \text{Bernoulli}(\theta)$.
  - This becomes your *model class*.
- Now from this model class, our objective is to find $\hat{\theta}$ (which is essentially *our* estimate of true θ^*).
  - This is where ML and MAP come into the picture.

Maximum Likelihood Estimation

- The question we try to answer here is: are there any specific values of θ that make our dataset $\mathbf{y}$ more "likely"?
- This is done by maximizing the likelihood function as follows:

$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} \underbrace{p(\mathbf{y}|\theta)}_{L(\theta)}.$$

- Likelihood function:
  - Note that it is a function of θ for a *given* $\mathbf{y}$.
  - Since all entries in our dataset will be assumed to be i.i.d., we will always have:

$$L(\theta) = \prod_{i=1}^{n} p(y_i|\theta).$$

Maximum Likelihood Estimation: Coin Toss Example

- Likelihood function for our example: $L(\theta) = \theta^{n_{\mathrm{H}}}(1-\theta)^{n_{\mathrm{T}}}$, where n_H and n_T represent the number of heads and tails observed in the dataset, respectively. Note that $n = n_H + n_T$.
- The maximum likelihood estimation problem for this case is:

$$\begin{aligned}\hat{\theta}_{\mathrm{ML}} &= \arg\max_{\theta} L(\theta) = \arg\max_{\theta} \log L(\theta) \\ &= \arg\max_{\theta} \{n_{\mathrm{H}} \log\theta + n_{\mathrm{T}} \log(1-\theta)\}\end{aligned}$$

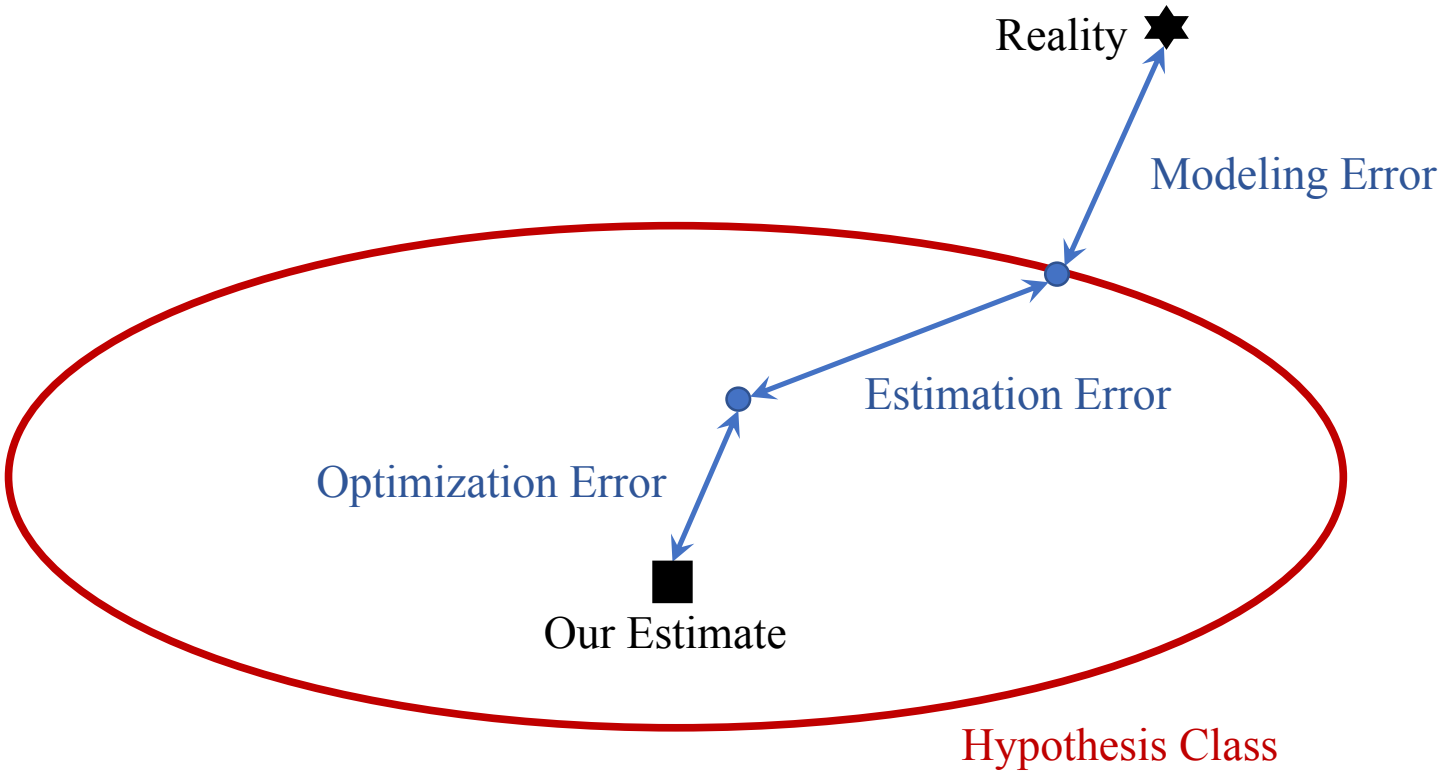
- Now if you take the partial derivative with respect to θ and set it to 0, you get

$$\hat{\theta}_{\mathrm{ML}} = \frac{n_{\mathrm{H}}}{n_{\mathrm{H}} + n_{\mathrm{T}}}.$$

Optimality of Maximum Likelihood Estimation

- As we will show now, maximum likelihood estimation minimizes KL divergence. It is "optimal" when our choice of model class is correct (i.e., θ^* lies in the model class).
- Reminder: Note that we assumed infinite data to establish this result.
- Let us now see why maximum likelihood may not always be sufficient when you have finite data.

Error Decomposition



- Modeling Error: Reality lies outside your chosen hypothesis class.
- Estimation Error: Estimate may be off because of limited data.
- Optimization Error: For instance, the problem could be np hard because of which some optimization error is unavoidable.
- You may be able to trade one error for the other.

Bayesian Estimation: MAP

- The general idea of Bayesian statistics is to treat θ as a random variable instead of being deterministic. Recall that in maximum likelihood, we treated it as deterministic.
- From Bayes' rule, we can write:

$$p(\theta|\mathbf{y}) = \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})}$$

  - Here $p(\theta)$ is our prior belief in θ, which was assumed constant before. As we will see in later lectures, this will also act as a regularizer.
- Specific estimator of interest is the so-called maximum *a posteriori* probability (MAP) estimator, which is defined as:

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} p(\theta|\mathbf{y}) = \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})}$$

  which is just the mode of the posterior.
  - MAP and maximum likelihood estimators are the same when θ is uniform.

Conjugate Priors

- It is useful to select prior distribution such that when it multiplies with the likelihood function, the posterior is of the same type (i.e., it has the same distribution as the prior but with potentially different parameters).
- The conjugate prior for the Bernoulli likelihood is the Beta distribution:

$$p(\theta) = \frac{\theta^{m_{\mathrm{H}}-1}(1-\theta)^{m_{\mathrm{T}}-1}}{B(m_{\mathrm{H}}, m_{\mathrm{T}})} \sim \mathrm{Beta}(m_{\mathrm{H}}, m_{\mathrm{T}}),$$

  where $B(m_{\mathrm{H}}, m_{\mathrm{T}}) = \frac{\Gamma(m_{\mathrm{H}})\Gamma(m_{\mathrm{T}})}{\Gamma(m_{\mathrm{H}}+m_{\mathrm{T}})}$ and $\Gamma(\cdot)$ is the Gamma function.
  - Here, m_{H} and m_{T} are the hyper parameters (parameters of the prior).
  - We will provide a useful interpretation of these parameters shortly.

MAP Estimation for our Coin Toss Example

- Choosing Beta distribution as our prior, the MAP estimate is:

$$\begin{aligned}\hat{\theta}_{\mathrm{MAP}} &= \arg\max_{\theta} p(\theta|\mathbf{y}) = \arg\max_{\theta} \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})} \\ &= \arg\max_{\theta} c\theta^{n_{\mathrm{H}}}(1-\theta)^{n_{\mathrm{T}}}\theta^{m_{\mathrm{H}}-1}(1-\theta)^{m_{\mathrm{T}}-1} \\ &= \arg\max_{\theta} c\theta^{n_{\mathrm{H}}+m_{\mathrm{H}}-1}(1-\theta)^{n_{\mathrm{T}}+m_{\mathrm{T}}-1} \\ &= \arg\max_{\theta} \mathrm{Beta}(n_{\mathrm{H}}+m_{\mathrm{H}}, n_{\mathrm{T}}+m_{\mathrm{T}}),\end{aligned}$$

  where we introduced a normalization constant c in the second step so that we do not have to explicitly track the terms that do not depend upon θ.
- Think of m_{H} and m_{T} as the "pseudo counts" that are known to us before we started the experiment (recall that this is a part of the "prior" distribution).

MAP Estimation for our Coin Toss Example

- Clearly, $\hat{\theta}_{\mathrm{MAP}}$ is nothing but the mode of $\mathrm{Beta}(n_{\mathrm{H}} + m_{\mathrm{H}}, n_{\mathrm{T}} + m_{\mathrm{T}})$, which is known to be:

$$\hat{\theta}_{\mathrm{MAP}} = \frac{n_{\mathrm{H}} + m_{\mathrm{H}} - 1}{n_{\mathrm{H}} + m_{\mathrm{H}} + n_{\mathrm{T}} + m_{\mathrm{T}} - 2}$$

- Note that if we had infinite data, the priors would not have even mattered. In that case, $\hat{\theta}_{\mathrm{MAP}} = \hat{\theta}_{\mathrm{ML}}$.

Generalization to Non-Binary Case: Setup and Likelihood Fn

- What if we roll a biased k sided die?
- The random variable X is no longer binary.
- Specifically, $P(X = j) = \theta_j$, for $1 \leq j \leq k$.
- Define: $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_k \end{bmatrix}$.
  - Example dataset for $k = 6$: $\mathbf{y} = \{3, 5, 2, 4, 6, 1, 1, 2\}$.
- Similar to the binary case, our objective here is to estimate $\hat{\boldsymbol{\theta}}$.
- The likelihood function for this case can be derived as:

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{j=1}^{k} \theta_j^{n_j}$$

where $n_j = \sum_{i=1}^{n} \mathbf{1}(y_i = j)$ is simply the number of times j appears in the dataset of length n.

Generalization to Non-Binary Case: MAP and ML Estimates

- Conjugate prior for this case is the Dirichlet distribution:

$$\boldsymbol{\theta} \sim \text{Dirichlet}(m_1, m_2, \ldots m_k) \propto \prod_{j=1}^{k} \theta_j^{m_j - 1}.$$

  - Simply a generalization of the Beta distribution.
- MAP estimate:

$$\hat{\theta}_{\text{MAP},j} = \frac{n_j + m_j - 1}{\sum_{l=1}^{k}(n_l + m_l) - k}$$

- There are two ways of obtaining the maximum likelihood estimator from the above result. Either drive the size of dataset to infinity or use the uniform prior ($m_l = 1, \forall l$ in this case). This gives

$$\hat{\theta}_{\text{ML},j} = \frac{n_j}{n}$$

where recall that $\sum_{l=1}^{k} n_l = n$.

Continuous Random Variables: Gaussian Case

- Consider a dataset $\mathbf{y} = \{y_1, y_2, \ldots y_n\}$.
- As noted on the previous slide, our model assumption here is $y_i \sim \mathcal{N}(\mu, \sigma^2)$:

$$p(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right).$$

- The ML estimates of μ and σ^2 can be easily shown to be:

$$\hat{\mu}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\hat{\sigma}^2_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}_{\mathrm{ML}})^2.$$

- Therefore, the maximum likelihood estimators for the mean and variance are nothing but the sample mean and (unadjusted or biased) sample variance.
  - For completeness, note that the adjusted (or unbiased) sample variance is: $\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \hat{\mu})^2$, where $\hat{\mu}$ is the sample mean.

Gaussian Model: MAP Estimator

- Recall from our earlier discussion that

$$\hat{\theta}_{\mathrm{MAP}} = \arg\max_{\theta} p(\theta|\mathbf{y}) = \arg\max_{\theta} \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})}.$$

- For our application, we just need to discuss $\hat{\mu}_{\mathrm{MAP}}$. We can therefore, consider σ^2 to be an *unknown* constant and estimate $\hat{\mu}_{\mathrm{MAP}}$.
- Conjugate prior: We assume $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$. Therefore:

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

- Now MAP estimate for μ can be directly expressed as:

$$\hat{\mu}_{\mathrm{MAP}} = \frac{\frac{\sum_{i=1}^{n} y_i}{\sigma^2} + \frac{\mu_0}{\sigma_0^2}}{\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}}$$

Linear Regression

- Assumes linear relationship between the inputs and outputs:

$$\hat{y} = \hat{f}(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j,$$

  where β_j's are the weights (also called the learning parameters). Here, β_0 is called the bias feature.
- It can also be expressed in terms of the vector notation:

$$\hat{y} = \begin{bmatrix} \beta_0 & \beta_1 \dots \beta_d \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} = \boldsymbol{\beta}^T \mathbf{x} = \mathbf{x}^T \boldsymbol{\beta}.$$

  Note the notation: $\mathbf{x} = \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix}$.
- Note that $\boldsymbol{\beta} \in \mathbb{R}^{d+1}$ due to the inclusion of the bias feature.
- While we will assume $y \in \mathbb{R}$, the results and insights can be easily generalized to the case where y is a vector.

Visualizing Linear Regression Solution

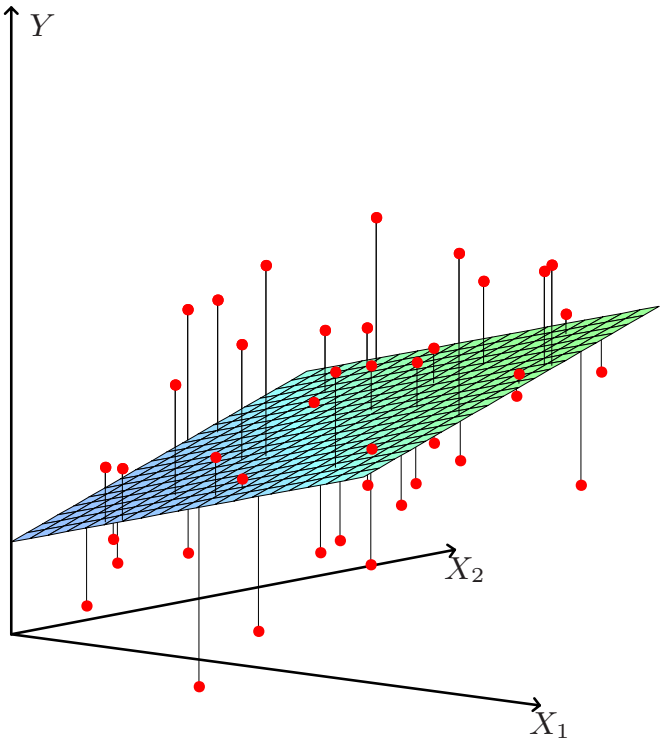


Figure: Linear least squares fit from Figure 3.1 of ESL.

- The vertical black line segments represent "error" $(y_i - \hat{y}_i)$.
- Linear regression essentially tries to minimize the "cumulative error" across all training points. This is formalized by defining a loss function, which we do on the next slide.

Linear (Least Squares) Regression: Loss Function

- We will use the squared loss function (least squares regression):

$$L(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - x_{i0}\beta_0 - x_{i1}\beta_1 \ldots - x_{id}\beta_d)^2$$

- This is a convex function in $\boldsymbol{\beta}$.
- Our learning objective is:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}) = \arg\min_{\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2.$$

- Can be obtained using gradient descent algorithm or directly as:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

where $\mathbf{X}^+ = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the Moore-Penrose pseudo-inverse of $\mathbf{X}$.

When Does Squared Loss Make Sense?

- Consider the following generative model.
- $\mathbf{x} \sim p(\mathbf{x})$: Features sampled from an arbitrary distribution. We do not need to know this distribution.
- Model assumption:

$$y_i|\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\beta}^T\mathbf{x}_i, \sigma^2).$$

- Parameters of this model: $\boldsymbol{\beta}$ and σ^2.
- Therefore, the output under this model is

$$y = \boldsymbol{\beta}^T\mathbf{x} + \mathcal{N}(0, \sigma^2).$$

- Assuming σ^2 to be an unknown constant, let us determine $\hat{\boldsymbol{\beta}}_{\mathrm{ML}}$.

Maximum Likelihood Estimate: $\hat{\boldsymbol{\beta}}_{\mathrm{ML}}$

- Maximizing the log likelihood function, we get:

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{\mathrm{ML}} &= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i, \boldsymbol{\beta}) \\
&= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \boldsymbol{\beta}^T \mathbf{x}_i)^2}{2\sigma^2} \right) \right] \\
&= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left[-\frac{(y_i - \boldsymbol{\beta}^T \mathbf{x}_i)^2}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2) \right] \\
&= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \mathbf{x}_i)^2.
\end{aligned}$$

- This is nothing but least squares regression.

Gaussian Assumption and Outliers

- Since Gaussian is not heavy-tailed, we are essentially assuming residual errors to be "small" whenever we apply least squares regression.
- If a few training points exhibit large errors, we call them outliers.
- It should not be surprising now that least squares will not work well in the presence of outliers.
- Simple fix: Model noise with a heavy-tailed distribution under which large "errors" are allowed. One possibility is the Laplace distribution:

$$y_i|\mathbf{x}_i \sim \text{Laplace}(\boldsymbol{\beta}^T\mathbf{x}_i, b)$$

$$p(y|\mathbf{x}_i, \boldsymbol{\beta}) = \frac{1}{2b}\exp\left(-\frac{|y_i - \boldsymbol{\beta}^T\mathbf{x}_i|}{b}\right)$$

$$\hat{\boldsymbol{\beta}}_{\text{ML}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} |y_i - \boldsymbol{\beta}^T\mathbf{x}_i|.$$

MAP Estimate $\boldsymbol{\beta}_{\text{MAP}}$: Regularization

- A natural question now is whether there is a similar connection between MAP and the loss function.
- This leads to the idea of Bayesian linear regression or regularized linear regression.
- As before, consider $y_i|\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\beta}^T\mathbf{x}_i, \sigma^2)$.
- We assume Gaussian prior: $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2\mathbf{I})$.
- In other words, β_i's are i.i.d. and each is $\sim \mathcal{N}(0, \sigma^2)$.
- The MAP estimate can now be expressed as:

$$\boldsymbol{\beta}_{\text{MAP}} = \arg\max_{\boldsymbol{\beta}} \frac{p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta})p(\boldsymbol{\beta})}{p(\mathbf{y}|\mathbf{X})}$$

- $\mathbf{X}$: Dataset of the feature vectors of all data points.

MAP Estimate $\boldsymbol{\beta}_{\text{MAP}}$

$$
\begin{aligned}
\boldsymbol{\beta}_{\text{MAP}} &= \arg\max_{\boldsymbol{\beta}} \ \frac{p(\mathbf{y}|\mathbf{X},\boldsymbol{\beta})p(\boldsymbol{\beta})}{p(\mathbf{y}|\mathbf{X})} \\
&= \arg\max_{\boldsymbol{\beta}} \ \log p(\mathbf{y}|\mathbf{X},\boldsymbol{\beta}) + \log p(\boldsymbol{\beta}) \\
&= \arg\max_{\boldsymbol{\beta}} \ \left[-\sum_{i=1}^{n} \frac{(y_i - \boldsymbol{\beta}^T\mathbf{x}_i)^2}{2\sigma^2} - \sum_{j=0}^{d} \frac{\beta_j^2}{2\sigma_0^2} \right] \\
&= \arg\max_{\boldsymbol{\beta}} \ \frac{1}{n} \left[-\sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T\mathbf{x}_i)^2 - \underbrace{\frac{\sigma^2}{\sigma_0^2}}_{\lambda} \sum_{j=0}^{d} \beta_j^2 \right] \\
&= \arg\min_{\boldsymbol{\beta}} \ \frac{1}{n} \left[\sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T\mathbf{x}_i)^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right]
\end{aligned}
$$

This is called regularized regression. Here, λ is the regularization parameter.

Polynomial Regression

- For simplicity of exposition, let us consider the case of a single input. Therefore, each training example is just (x, y).
- We can always define our hypothesis function as an m^{th} order polynomial of x:

$$\hat{y} = \hat{f}(x, \boldsymbol{\beta}) = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_m x^m.$$

- $\hat{f}(x, \boldsymbol{\beta})$ is linear in $\boldsymbol{\beta}$ even though it is not linear in x. However, it is just the linearity in terms of $\boldsymbol{\beta}$ that really matters. We can always treat each non-linear term as a new "feature", which is the reason it is called feature augmentation or basis expansion.
- In general, we can express $\hat{f}(\boldsymbol{x}, \boldsymbol{\beta})$ as $\boldsymbol{\beta}^T \phi(\boldsymbol{x})$, where $\phi(\boldsymbol{x}) : \mathbb{R}^{m_1} \rightarrow \mathbb{R}^{m_2}$.

Polynomial Regression

- One example of $\phi(\boldsymbol{x}) : \mathbb{R}^{m_1} \rightarrow \mathbb{R}^{m_2}$ is given below:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_1^2 \\ x_2^2 \\ x_3^2 \\ x_1 x_2 \\ \vdots \end{bmatrix}$$

- Interpretation: A lower dimensional vector is embedded in a higher dimensional space so as to facilitate linear processing. This is also what we implicitly do in advanced techniques such as neural networks.

Summary

- We provided a brief introduction to statistical estimation.
- Interpreted least squares regression as a maximum likelihood problem.
- Interpreted regularized linear regression as a MAP estimator.
- Basis expansion or feature augmentation to perform polynomial regression.