

Machine Learning Basics: Maximum Likelihood Estimation

Sargur N. Srihari
srihari@cedar.buffalo.edu

This is part of lecture slides on [Deep Learning](http://www.cedar.buffalo.edu/~srihari/CSE676):
<http://www.cedar.buffalo.edu/~srihari/CSE676>

Topics

1. Learning Algorithms
2. Capacity, Overfitting and Underfitting
3. Hyperparameters and Validation Sets
4. Estimators, Bias and Variance
5. Maximum Likelihood Estimation
6. Bayesian Statistics
7. Supervised Learning Algorithms
8. Unsupervised Learning Algorithms
9. Stochastic Gradient Descent
10. Building a Machine Learning Algorithm
11. Challenges Motivating Deep Learning

Topics in Maximum Likelihood

0. The maximum likelihood principle

- Maximizing likelihood is minimizing KL divergence

1. Conditional log-likelihood & MSE

- Minimizing negative log-likelihood is equivalent to minimizing MSE in linear regression with Gaussian noise

2. Properties of maximum likelihood

- No consistent estimator has lower MSE than maximum likelihood estimator

How to obtain good estimators?

- We have seen some definitions of common estimators and their properties
 - Ex: sample mean, bias-variance
- Where do they come from?
- Rather than guessing some function and determining its bias and variance, better to have some principle from which we can derive specific functions that are good estimators
- The most common such principle is the maximum likelihood principle

Maximum Likelihood Principle

- Consider set of m examples $\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}\}$
 - Drawn independently from the true but unknown data generating distribution $p_{\text{data}}(\mathbf{x})$
- Let $p_{\text{model}}(\mathbf{x}; \theta)$ be a parametric family of distributions over same space indexed by θ
 - i.e., $p_{\text{model}}(\mathbf{x}; \theta)$ maps any configuration of \mathbf{x} to a real no. estimating the true probability $p_{\text{data}}(\mathbf{x})$
- The maximum likelihood estimator for θ is:

$$\begin{aligned}\theta_{\text{ML}} &= \arg \max_{\theta} p_{\text{model}}(\mathbb{X}; \theta) \\ &= \arg \max_{\theta} \prod_{i=1}^m p_{\text{model}}(\mathbf{x}^{(i)}; \theta)\end{aligned}$$

- This product over many probabilities is inconvenient
 - ex: underflow

Alternative form of max likelihood

- An equivalent optimization problem is to take logarithm of the likelihood

$$\theta_{\text{ML}} = \arg \max_{\theta} \sum_{i=1}^m \log p_{\text{model}}(\mathbf{x}^{(i)}; \theta).$$

- Since dividing by m does not change the problem
- This maximization can be written as

$$\theta_{\text{ML}} = \arg \max_{\theta} \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(\mathbf{x}; \theta).$$

- The expectation is wrt the empirical distribution \hat{p}_{data} defined by the training data
- One way to interpret maximum likelihood estimation is to view it as minimizing the dissimilarity between the empirical distribution \hat{p}_{data} defined by the training set and the model distribution $p_{\text{model}}(\mathbf{x})$ as seen next

Maximizing likelihood is minimizing KL divergence

- Max likelihood = minimizing KL diverg. between:
 - empirical distribution \hat{p}_{data} and model distribution $p_{\text{model}}(\mathbf{x})$

- The K-L divergence is

$$D_{\text{KL}}(\hat{p}_{\text{data}} \| p_{\text{model}}) = \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} [\log \hat{p}_{\text{data}}(\mathbf{x}) - \log p_{\text{model}}(\mathbf{x})]$$

– First Term a function of data generation, not model

– Thus we only need to minimize $-E_{\mathbf{x} \sim \hat{p}_{\text{data}}} [\log p_{\text{model}}(\mathbf{x})]$

- i.e., cross entropy between distribution of training set and probability distribution defined by model

– *Definition of cross entropy* between distributions p and q is

$$H(p, q) = E_p[-\log q] = H(p) + D_{\text{KL}}(p \| q)$$

– For discrete distributions $H(p, q) = -\sum_{\mathbf{x}} p(\mathbf{x}) \log q(\mathbf{x})$

Summary of Max Likelihood

- It is an attempt to make model distribution p_{model} match the empirical distribution \hat{p}_{data}
 - Ideally we would like to match the unknown \hat{p}
- The optimal θ is the same whether we maximize likelihood or minimize KL divergence
- In software both are phrased as minimization
 - Maximum likelihood becomes minimization of negative log-likelihood (NLL)
 - Equivalently minimization of cross entropy
 - KL divergence has a minimal value 0
 - Negative log-likelihood is negative with no bound

Conditional Log-likelihood and MSE

- Maximum likelihood estimator can be readily generalized to parameters of an input-output relationship
 - Goal is prediction: conditional probability $P(\mathbf{y}|\mathbf{x};\theta)$
 - Which forms the basis of most supervised learning
- If \mathbf{X} represents all our inputs and \mathbf{Y} represents all our targets then the conditional maximum likelihood estimator is $\theta_{\text{ML}} = \arg \max_{\theta} P(\mathbf{Y} | \mathbf{X}; \theta)$
 - If examples are i.i.d. then

$$\theta_{\text{ML}} = \arg \max_{\theta} \sum_{i=1}^m \log P(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}; \theta)$$

Linear Regression as Maximum Likelihood

- Basic linear regression:
 - Takes an input \mathbf{x} and produce an output \hat{y}
 - Mapping from \mathbf{x} to \hat{y} is chosen to minimize MSE
- Revisit linear regression as maximum likelihood
 - Think of model as producing a conditional distribution $p(\mathbf{y}|\mathbf{x})$
- To derive same algorithm, define $p(y | x) = N(y; \hat{y}(\mathbf{x}, \mathbf{w}), \sigma^2)$
 - Function $\hat{y}(\mathbf{x}, \mathbf{w})$ predicts mean of Gaussian
 - Since samples are i.i.d.

$$\sum_{i=1}^m \log p(y^{(i)} | \mathbf{x}^{(i)}; \boldsymbol{\theta})$$
$$= -m \log \sigma - \frac{m}{2} \log(2\pi) - \sum_{i=1}^m \frac{\|\hat{y}^{(i)} - y^{(i)}\|^2}{2\sigma^2},$$

Thus maximizing the log-likelihood is same as minimizing MSE

$$\text{MSE}_{\text{train}} = \frac{1}{m} \sum_{i=1}^m \|\hat{y}^{(i)} - y^{(i)}\|^2$$

Properties of Maximum Likelihood

- Main appeal of maximum likelihood estimator:
 - It is the best estimator asymptotically
 - In terms of its rate of converges, as $m \rightarrow \infty$
 - Under some conditions, it has consistency property
 - As $m \rightarrow \infty$ it converges to the true parameter value
 - Conditions for consistency
 - p_{data} must lie within model family $p_{\text{model}}(\cdot, \theta)$
 - p_{data} must correspond to exactly one value of θ

Statistical Efficiency

- Several estimators, based on inductive principles other than MSE, can be consistent
- Define *Statistical efficiency*: estimator has lower generalization error for fixed no of samples
 - Or equivalently, needs fewer examples to obtain a fixed generalization error
- Needs measuring closeness to true parameter
 - MSE between estimated and true parameter values
 - Parameter MSE decreases as m increases
 - Using Cramer-Rao bound
 - No consistent estimator has lower MSE than Maximum Likelihood Estimator