ECE 449/590 – OOP and Machine Learning
Lecture 15 Machine Learning Basics II

Professor Jia Wang
Department of Electrical and Computer Engineering
Illinois Institute of Technology

October 17, 2022

Outline

Reading Assignment

- This lecture: Deep Learning 5
- Next lecture: Deep Learning 6

Outline

Point Estimation

- Input: a set of i.i.d. (independent and identically distributed) data points/examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$.
- Assume the data points are sampled from a parametric distribution with the parameter $\boldsymbol{\theta}$.
  - E.g. for Gaussian (normal) distribution, $\boldsymbol{\theta} = (\mu, \sigma)$.
  - The type of the distribution itself, e.g. Gaussian or uniform distribution, is not part of the parameter.
- A point estimator is any function of the input
  - $\hat{\boldsymbol{\theta}}_m = g(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)})$
  - A good estimator provides a close approximation to $\boldsymbol{\theta}$.
- In practice, the distribution is usually modeled as a function depending on $\boldsymbol{\theta}$ and random variables with known distribution.
  - E.g. $f(x; \boldsymbol{\theta}) = \sigma x + \mu$ generates any Gaussian distribution from x that is sampled from the standard Gaussian distribution.
  - The point estimator actually solves the function estimation problem in such case.

Bias and Variance

- Since $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ are sampled from a distribution, they should be treated as random variables.
  - So $\hat{\boldsymbol{\theta}}_m$ is a random variable.
- For a random variable ($\hat{\boldsymbol{\theta}}_m$) to approximate a value ($\boldsymbol{\theta}$) well,
  - The expectation $\mathbb{E}(\hat{\boldsymbol{\theta}}_m)$ should be $\boldsymbol{\theta}$, or the bias $\text{bias}(\hat{\boldsymbol{\theta}}_m) = \mathbb{E}(\hat{\boldsymbol{\theta}}_m) - \boldsymbol{\theta}$ should be 0.
  - The variance $\mathrm{Var}(\hat{\boldsymbol{\theta}}_m)$ should approach 0.
- E.g. $\hat{\mu}_m = \frac{1}{m}\sum_{i=1}^m x^{(i)}$ is an estimator of Gaussian mean μ.
  - $\text{bias}(\hat{\mu}_m) = 0$, $\mathrm{Var}(\hat{\mu}_m) = \frac{\sigma^2}{m}$
  - A very good estimator for μ but what about estimators for σ?
- Machine learning algorithms as estimators need to make trade-off between bias and variance.

Bias and Variance

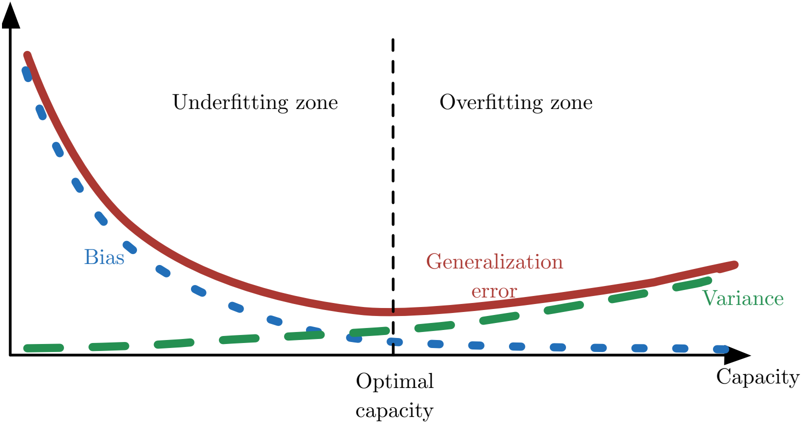

Figure 5.6

(Goodfellow 2016)

Outline

Estimator Design via Maximum Likelihood

- How can we design a good estimator?
  - Use maximum likelihood as a common principle.
- Let $\mathbb{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ be sampled i.i.d. from p_{data}.
  - If we further assume no one could know the actual p_{data}, the best we can do is to choose a known parametric distribution $p_{model}(\mathbf{x}; \boldsymbol{\theta})$ and then estimate $\boldsymbol{\theta}$.
- Maximum likelihood: maximize the probability that you would observe $\mathbb{X}$ assuming they are sampled from $p_{model}(\mathbf{x}; \boldsymbol{\theta})$.

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} p_{model}(\mathbb{X}; \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{model}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

Log-Likelihood and Cross-Entropy

- It is more convenient to work with log-likelihood on computers.

$$
\begin{aligned}
\boldsymbol{\theta}_{ML} &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{model}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}\sim p_{data}} \log p_{model}(\boldsymbol{x}; \boldsymbol{\theta})
\end{aligned}
$$

- As the negative log-likelihood $-\mathbb{E}_{\mathbf{x}\sim p_{data}} \log p_{model}(\boldsymbol{x})$ is known as the cross-entropy of p_{model} with respect to p_{data}, to maximize likelihood is equivalent to minimize cross-entropy.

Conditional Log-Likelihood and Supervised Learning

- So far we assume the examples are $\mathbb{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ when applying maximum likelihood estimators.
  - What about the labels $\{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}\}$ for supervised learning?
- Estimate $\boldsymbol{\theta}$ in a conditional distribution model $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$.
  - With the model, for a given $\boldsymbol{x}$, the output is predicted as $\arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$
- Conditional maximum likelihood: maximize the probability that you would observe the labels given $\mathbb{X}$.

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta})$$

Example: Maximum Likelihood for Gaussion

- Supervised learning: scalar labels $\{y^{(1)}, \ldots, y^{(m)}\}$ for the examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$.
  - Assume $p(\text{y}|\mathbf{x}; \boldsymbol{\theta})$ is of a Guassion distribution of known variance σ^2 and unknown mean μ that depends on x and $\boldsymbol{\theta}$.
  - The prediction is $\hat{y} = \arg\max_y p(y|\boldsymbol{x}; \boldsymbol{\theta}) = \mu$.
  - E.g. if $\mu = \boldsymbol{\theta}^\top \boldsymbol{x}$ then this is linear regression.
- The log-likelihood is

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

- To maximize likelihood is equivalent to minimize MSE.
  - As long as $p(\text{y}|\mathbf{x}; \boldsymbol{\theta})$ is Guassion with known variance but unknown mean.

Outline

Probabilistic Supervised Learning

- Choose a parametric family $p_{model}(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ of conditional distributions.
- Apply maximum likelihood estimation to obtain $\boldsymbol{\theta}$.
- We have seen the example for $\mathbf{y}$ being continuous and p_{model} being Gaussion.
  - But for the more general case, it is difficult to specify the conditional pdf (probability density function).
- For $\mathbf{y}$ being discrete, it is possible to specify the conditional probability for individual cases.
  - E.g. logistic regression when $\mathbf{y}$ can be either 0 or 1:

    $$p_{model}(y=1|\boldsymbol{x};\boldsymbol{\theta}) = \mathsf{sigmoid}(\boldsymbol{\theta}^\top \boldsymbol{x})$$

    where $\mathsf{sigmoid}(a) = \frac{e^a}{e^a+1}$.
  - When $\mathbf{y}$ may take more values, one may use a vector-valued function of $\boldsymbol{\theta}$ and $\boldsymbol{x}$ for such purpose.
    - The challenge is to find the proper form of the function to be able to approximate the unknown p_{data}.

Other Supervised Learning Algorithms

- Support vector machine: mainly for binary classification
- k-nearest neighbor: memorizing training set
- Decision tree: branching on features one by one
- Has their limitations but may work for specific problems and settings.

Outline

Representation and Unsupervised Learning

- Find the "best" representation for the data set $\mathbb{X}$.
  - Simpler or more accessible than $\mathbb{X}$.
  - While preserving as much information about $\mathbb{X}$ as possible.
- Common approaches
  - Lower-dimensional representations where information is compressed.
  - Sparse representations where only non-zero elements need to be stored.
  - Independent representations where a joint pdf is decomposed into products of marginal pdfs.

Principal Components Analysis (PCA)

- Consider $\mathbb{X}$ as the design matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$.
  - Assume $\mathbb{E}[\boldsymbol{x}] = 0$ for $\boldsymbol{x}$ be an example (a column of $\boldsymbol{X}^\top$).
- Unbiased sample covariance: $\mathrm{Var}[\boldsymbol{x}] = \frac{1}{m-1}\boldsymbol{X}^\top \boldsymbol{X}$
- Eigendecomposition: $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^\top$.
  - where $\boldsymbol{W}$ is orthogonal and $\boldsymbol{\Lambda}$ is diagonal.
- Let $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{W}$, then $\mathrm{Var}[\boldsymbol{z}] = \frac{1}{m-1}\boldsymbol{\Lambda}$.
- Now $\boldsymbol{Z}$ works as a representation of $\boldsymbol{X}$ while all $\boldsymbol{z}$ are mutually uncorrelated.

Principal Components Analysis

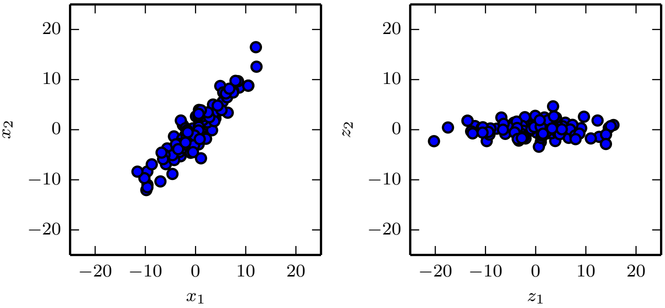

Figure 5.8

(Goodfellow 2016)

k-means Clustering

- Divide $\mathbb{X}$ into k different clusters.
  - Each example can be represented as a length-k one hot code.
- k-means algorithm: each cluster has a centroid and examples in a cluster are closer to its own centroid than to other centroids.
- When k is not given and thus is a hyperparameter, how to decide if it is good?

Summary

- Bias and variance matter for estimators.
- Maximum Likelihood is widely used to design estimators.
- Simple supervised and unsupervised learning algorithms.