Chapter 5. Machine Learning Basics

Overview

Machine learning is a form of **applied statistics** that with increased emphasis on the use of computer to statistically estimate complicated functions and a decreases emphasis on proving confidence interval around these functions.

5.1 Learning Algorithm

Overview

Definition of learning algorithm

A computer program is said to learn from experience E with respect to some classes of T and performance P, if its performance at tasks in T, as measured by P, improves with experience E.

5.1.1 The Task

- Classification
- Classification with missing inputs
- Regression
- Transcription
- Machine Translation
- Structured output
- Anomaly detection
- Synthesis and sampling
- Imputation of missing values
- Denoise
- Density estimation

5.1.2 The Performance Measure

Use **test set** to evaluate the performance

5.1.3 The Experience

- Supervised
- Unsupervised

5.1.4 Example: Linear Regression

Refer to the book

5.2 Capacity, Overfitting and Underfitting

Overview

Train error and test error

- The ability to perform well on previously unseen inputs is called **generalization**
- What separates machine learning from optimization is that we also want the **generalization error** or the **test error** to be low

Problem: How can we affect the algorithm's performance on test set when we only get to observe the train set ?

Basic assumption about data generating process

- If the train and test dataset are generated arbitrarily, the machine learning algorithm is hard to generalize. Under this situation, the datasets affect the performances more than the choice and design of machine learning algorithm
- **Basic Assumption:** The examples in each dataset are independent from each other. The train and test dataset are identically distributed, drawn from the same underlying probability distribution called **data generating distribution** p_{data}
- With this basic assumption, for any untrained machine learning algorithm, the expected train error and test error are the same.
- The two facts that determine the goodness of a machine learning algorithm.
    - Make train error small
    - Make the gap between the train and test error small
- Fail to make train error small => underfitting
- Fail to make the gap between train and test error small => overfitting

Capacity

Capacity is the ability to fit more complex function.

- Low capacity algorithm tends to underfitting
- High capacity tends to overfitting
- Choose a proper capacity is critical

Method to control the capacity of a learning algorithm:

- Properly choose a hypothesis space

Representationnal capacity and effective capacity:

- By choosing the hypothesis space, we choose the representational capacity. But the training process cannot find the optimal function in the hypothesis space.
- The effective capacity may be less than the representational capacity

Qualifying model capacity

- Vapnik-Chervonenkis dimension (VC dimension)
- The most important results in statistical learning theory show that the discrepancy between training error and generalization error is bounded from above by a quantity that grows as the model capacity grows but shrinks as the number of training examples increases
    - If we want complex model, we need more training and testing examples

Bayes error

- If we get a oracle that already know the true probability distribution that generate the data, this oracle will still incur some error.
- The error incurred by a oracle making prediction from the true distribution $p(\mathrm{x}, y)$ is called **bayes error**.
- Bayes error is the error caused by the intrinsic stochastic property of the underlying problem/

5.2.1 The No Free Launch Theorem

No machine learning algorithm is universally better than any other machine learning algorithm.

The Most complex model may perform worse than a random gauss or a linear regression.

Our goal is to **understand what kinds of distributions** are relevant to the "real world" that an AI agent experiences, and **what kinds of machine learning algorithms perform well on data** drawn from the kinds of data generating distributions we care about.

- Understand the data generating probability
- Choose proper machine learning algorithm for that specific data generating probability

5.2.2 Regularization

Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error

5.3 Hyperparameters and validation set

Hyperparameters: settings that we used to control the behavior of machine learning algorithms

- settings that are hard to optimize
- settings that are not appropriate to learn
    - For all settings that control the model capacity, the machine learning algorithm tends to choose the settings that maximize the model capacity

How to determine the hyperparameters:

- The test set cannot be used to choose the hyperparameter
- Split another validate set to choose the hyperparameter from the training set

5.3.1 Cross-validation

- **Used when the training data is not enough**
- K-fold cross-validation(typical method)

5.4 Estimator, Bias and Variance

Question: How can the machine learning algorithm generalize to unseen examples given the fact that the algorithm is only trained with training dataset.

5.4.1 Point Estimation

Point estimation is the attempt to provide the single "best" prediction of some quality of interesting.

- The estimated quality value $\hat{\theta}$
- The true value denotes θ

Function estimation: A type of point estimation that estimate a "best" function in a function space.

> The definition of point estimation here is quite general. I think the author just want to present the basic concept and idea. So just remember these concepts.

5.4.2 Bias

Definition:

The bias of an estimator is defines as:

$$bias(\hat{\theta}_m) = \mathbb{E}[\hat{\theta}_m] - \theta$$

The expectation is over the data, θ is the true underlying value used to define the data generating probability.

- Unbiased => $bias(\hat{\theta}_m) = 0$
- asymptotically unbiased => $\lim_{m\to\infty} \mathbb{E}[\hat{\theta}_m] = \theta$

> 1. What is m ? (m is the number of data examples)
> 2. over that data ?
> 3. what is the meaning of $m \to \infty$ (Have infinite data example)

Example: Estimator of mean of Gaussian distribution

Consider a set of $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ that are independently and identically distributed according to a Guassian distribution $p(x^{(0)}) = \mathcal{N}(x^{(i)}; \mu, \sigma^2)$, where $i \in \{1, 2, \ldots, m\}$.

The gaussian distribution density function is:

$$p(x^{(i)}; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2}\frac{(x^{(i)} - \mu)^2}{\sigma^2})$$

A common estimator for Gaussian mean parameter is **sample mean**, this means we use the sample mean to estimate the true underlying mean

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

The estimation bias of sample mean is

$$\begin{aligned} bias(\hat{\mu}) &= \mathbb{E}[\hat{\mu}] - \mu \\ &= \mathbb{E}[\frac{1}{m}\sum_{i=1}^{m} x^{(i)}] - \mu \\ &= \frac{1}{m}\mathbb{E}[\sum_{i=1}^{m} x^{(i)}] - \mu \\ &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}[x^{(i)}] - \mu \\ &= \mu - \mu = 0 \end{aligned}$$

So the sample mean is a unbiased estimator

Example: Estimator of variance of Gaussian distribution

Under the same data distribution and data set, we estimate the variance this time. The most intuitive variance estimator is.

$$\hat{\sigma}_m = \frac{1}{m}\sum_{i=i}^{m}(x^{(i)} - \bar{x})^2$$

then the expectation of estimator is:

$$
\begin{aligned}
\mathbb{E}[\hat{\sigma}_m] &= \mathbb{E}[\frac{1}{m}\sum_{i=i}^{m}(x^{(i)} - \bar{x})^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=i}^{m}(x^{(i)} - \bar{x})^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=i}^{m}(x^{(i)} - \mu + \mu - \bar{x})^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=i}^{m}((x^{i} - \mu) + (\mu - \bar{x}))^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=i}^{m}((x^{(i)} - \mu)^2 + 2(x^{(i)} - \mu)(\mu - \bar{x}) + (\mu - \bar{x})^2)] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=i}^{m}(x^{(i)} - \mu)^2 + 2(\mu - \bar{x})\sum_{i=i}^{m}(x^{(i)} - \mu) + m(\mu - \bar{x})^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=i}^{m}(x^{(i)} - \mu)^2 + 2m(\mu - \bar{x})(\bar{x} - \mu) + m(\mu - \bar{x})^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=i}^{m}(x^{(i)} - \mu)^2 - 2m(\mu - \bar{x})^2 + m(\mu - \bar{x})^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=i}^{m}(x^{(i)} - \mu)^2 - m(\mu - \bar{x})^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=i}^{m}(x^{(i)} - \mu)^2] - \mathbb{E}[(\bar{x} - \mu)^2] \\
&= \sigma^2 - \mathbb{E}[(\bar{x} - \mu)^2] \\
&= \sigma^2 - \frac{1}{m}\sigma^2
\end{aligned}
$$

The estimation bias is

$$
bias(\hat{\sigma}_m) = -\frac{1}{m}\sigma^2
$$

So, the most intuitive estimator derived from the definition of variance is a biased estimator of the true variance.

The **unbiased estimator** for Gaussian distribution variance is

$$
\hat{\sigma}_m^{unbiased} = \frac{1}{m-1}\sum_{i=i}^{m}(x^{(i)} - \bar{x})^2
$$

Which is usually referred as the **sample variance**

Note:

- While unbiased estimators are clearly desirable, they are not always the “best” estimators.

5.4.3 Variance and Standard Error

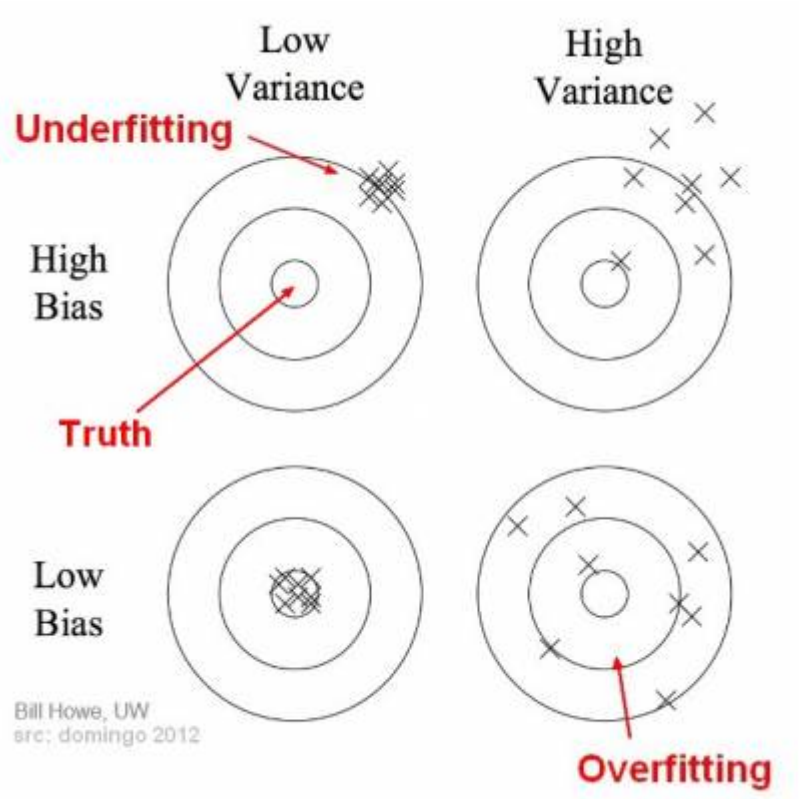
Definition: Variance of a estimator $\hat{\theta}$ is

$$Var(\hat{\theta}_m) = \mathbb{E}[(\hat{\theta}_m - \mathbb{E}[\hat{\theta}_m])^2]$$

Variance provides a measure of how the estimated value change as we resample the dataset with the same data generating distribution.

5.4.4 Trading off Bias and Variance

Bias and variance of the estimator are the two sources of error as show in the following figure.



Mean square error of a estimator

$$MSE = \mathbb{E}[(\hat{\theta}_m - \theta)^2] = Bias(\hat{\theta}_m)^2 + Var(\hat{\theta}_m)$$

How to choose better model

- What happens when we are given a choice between two estimators, one with more bias and one with more variance
- The most common way to negotiate this trade-off is to use **cross-validation**

Relationship between bias, variance, capacity, overfitting and underfitting

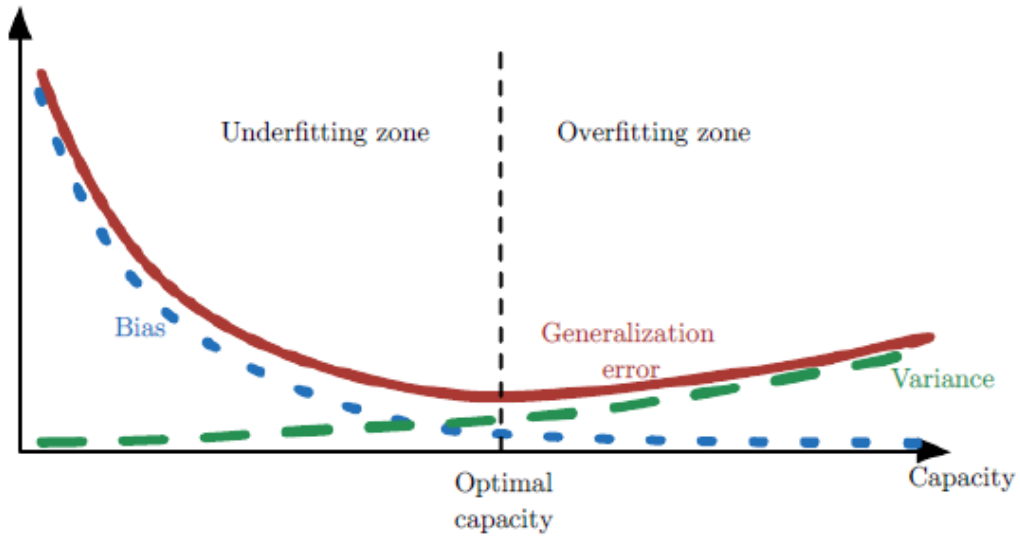


Figure 5.6: As capacity increases (x-axis), bias (dotted) tends to decrease and variance (dashed) tends to increase, yielding another U-shaped curve for generalization error (bold curve). If we vary capacity along one axis, there is an optimal capacity, with underfitting when the capacity is below this optimum and overfitting when it is above. This relationship is similar to the relationship between capacity, underfitting, and overfitting, discussed in section 5.2 and figure 5.3.

5.4.5 Consistency

The discussion about bias and variance fix the size m of the dataset $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$. In other aspect, we want the estimation converge to the true value as the dataset size m increase.

Weak Consistency

$$plim_{m\to\infty}\hat{\theta}_m = \theta$$

$plim$ is the probability limit

Strong Consistency: \ Consider the estimation as the dataset size increase $\{\theta_1, \theta_2, \ldots, \theta_n\}$, θ_i indicates the estimation when the dataset have size i. θ_i is a random variable. The strong consistency is:

$$P(\lim_{m\to\infty}\theta_m = \theta) = 1$$

Consistency and unbiasedness

- a estimator is consistent $\Rightarrow$ a estimator is asymptotically unbiased
- a estimator is asymptotically unbiased $\not\Rightarrow$ a estimator is consistency

5.5 Maximum Likelihood Estimation

Only need to understand that maximum likelihood estimation is to find the best-fit probability distribution given some examples set $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$

Read this post

> Ignore the mathematical derivation here, I think I need to learn a statistic course to better understand these related materials. So for the first reading pass the book, I choose to jump directly to Part II of the book. Complete this part when I reading the book the second time.

5.5.1 Conditional Log-likelihood and Mean Squared Error

This is the basis for most supervised learning algorithms. If X is all the input and Y is all the labels, then the conditional maximum likelihood estimator is:

$$\theta_{ML} = \arg\max_{\theta} P(Y|X;\theta)$$

If the examples are assumed to be i.i.d. Then this can be decomposed into

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

5.5.2 Properties of Maximum Likelihood

Maximum likelihood estimator is the best estimator asymptotically.

Under following conditions, the maximum likelihood estimator has the property of consistency

- The true distribution p_{data} must lie within the model family $p_{model}(\cdot|\theta)$
- The true distribution p_{data} must corresponding to exactly one value of θ

5.6 Bayes Statistics

> 5.5 and 5.6 illustrate the basic ideas and the main difference of frequentist and bayes statistics. Up to the first pass reading, I am still quite comfused about these materials.

Frequentist Statistics

- Use probability to present the times some events will appear given the total trials
- θ is fixed but unknown, use some kinds of estimators to estimate the true underlying value θ. The estimator is a function of the training data.
- The estimator $\hat{\theta}_m$ is the random variable, the data observed is also a random variable x

Bayes Statistics

- Use a *priori* that is subjective.
    - Priori increase the bias but reduce the variance
- Use probability to reflect the degree of certainty of states of knowledge
- The dataset observed is fixed and certain
- The underlying true parameter θ is unknown and is a random variable

A good reading from MIT about the difference between frequentist and Bayes statistics

5.6.1 Maximum A Posteriori(MAP) Estimation

> Confused for the first pass reading ,

5.7 Supervised Learning Algorithms

5.7.1 Probabilistic Supervised Learning

Use maximum likelihood estimation to solve the supervised learning problems.

- Linear regression (Check 5.5.1. Linear regression from the perspective of maximum likelihood estimation)
- Logistic regression

5.7.2 Support Vector Machine

> The illustration in this book is not so good. Just check the corresponding part of the book(统计学习方法，李航)

Support vector machine is a binary classifier

- **Kernel method** or Kernel machine

5.7.3 Other Simple Supervised Learning Algorithms

K-nearest neighbors (KNN)

- K-nearest neighbors is a non-probabilistic supervised learning methods
- KNN have quite high capacity
- KNN cannot learning the importance of difference features

Decision Tree

- Sub-divide space into non-overlapping regions

5.8 Unsupervised Learning Algorithms

> Here this book introduces two traditional unsupervised learning algorithm. 李航机器学习方法 is a better book for further understand.

5.8.1 Principal Components Analysis

5.8.2 K-mean Clustering

5.9 Stochastic Gradient Descent

Stochastic Gradient Descent

- Use a estimated gradient to conduct gradient descent
- Can train non-linear models on large amount of data

Cost function

For a training set $\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$ with size m, the cost function and the gradient is

$$J(\theta) = \mathbb{E}_{\mathbf{x}, y \sim \hat{p}_{data}} L(x, y, \theta) = \frac{1}{m} \sum_{i=1}^{m} L(x^{(i)}, y^{(i)}, \theta)$$

$$\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(x^{(i)}, y^{(i)}, \theta)$$

L is the per-example loss. For negative-log likelihood loss

$$L(x, y, \theta) = -\log P(y|\mathrm{x}; \theta)$$

Calculating the average loss over the whole dataset is computation expensive, so a estimated average over a small subset $\{x^{(1)}, x^{(2)}, \cdots, x^{(m')}\}$ with size m' is used. The estimated gradient is

$$\hat{g} = \frac{1}{m'} \sum_{i=1}^{m'} \nabla_\theta L(x^{(i)}, y^{(i)}, \theta)$$

5.10 Building a Machine Learning Algorithm

Components of a machine learning algorithm

- A specification of a dataset
- A cost function
- An optimization procedure
- A model

Dataset

- Under most situation, the study of machine learning is based on the assumption that the dataset is generated properly

Cost function

- If Cost function allows close-form optimization,
- If cost function don't have closed-form optimization
    - Iterative optimization methods like gradient descent
- If the cost function cannot be evaluate for computational reasons, as long as its gradient can be estimated, it can be optimized

Optimization procedure

- If a machine learning algorithm seems quite unique, it can be understood as using a special-case optimizer

5.11 Challenges Motivating Deep Learning

The problem of high dimensionality

The challenge faced by machine learning or deep learning is actually caused by the high dimensionality

5.11.1 The Curse of Dimensionality

Possible settings grows exponentially with data dimension.

Machine learning problems becomes exceedingly difficult when the number of dimensions in the data is high. This is called the curse of **dimensionality**

5.11.2 Local Constancy and Smoothness Regularization

> Confused about this chapter for first pass reading

In order to generalize well, machine learning algorithms need to be guided by prior beliefs about what kind of function they should learning.(Like choose to use CNN for image classification and then design the architecture of CNN, these are the prior beliefs)

Prior beliefs in machine learning

- Explicitly or implicitly priori beliefs
- **Smoothness prior** or **local constancy prior** is the most widely used implicit prior

5.11.3 Manifold Learning

> A good post about manifold learning

Basic Assumption

- Redundancy in high dimension data
- Data can be regarded as lying in a low dimensional manifold
- A mapping from low dimensional manifold to high original data dimension exists