Machine Learning

T.H.E. Note

YBiuR

Super Vegetable Me

Preface

```
Loss: NaN

Acc: 0.00
```

Chapter 1

Introduction

"你们 *CS* 比 *CS* 的强吗？那你们的 *EE* 比 *EE* 的强吗？"

1.1 Basics

The basic assumption of machine learning is that **data samples are i.i.d.**.
The goal of training a model is to **minimize the generalization error of the model**. Since we only have limited amount of data, what we can actually do is to minimize the emprical error.

1.2 Overfitting and Underfitting

Overfitting. High variance. The model performs well on training sets but performs poorly on new unseen samples. Using a high-order model to fit low-order distribution of data usually leads to overfitting.

Underfitting. High bias. The model has not fully captured the underlying structure of the data. Conduct more training or change a more complicated model.

1.3 Methods for Splitting data

To train a model, we first need to divide data into training set and test set. Training set and test set should be disjoint.

1.3.1 Hold-Out

Divide dataset $\mathcal{D}$ into traning set $\mathcal{S}$ and test set $\mathcal{T}$ s.t.

$$\mathcal{S} \cup \mathcal{T} = \mathcal{D} \quad \mathcal{S} \cap \mathcal{T} = \emptyset$$

Typical proportion of $\mathcal{S}$ and $\mathcal{T}$ is 30% and 70%.

1.3.2 Cross-Validation

Divide $\mathcal{D}$ into k disjoint sets of similar size.

$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \ldots \mathcal{D}_k \quad \text{s.t.} \quad \mathcal{D}_i \cap \mathcal{D}_j = \emptyset$$

Each time use $k-1$ sets for training and the remaining set for testing. A typical value of k is 10.

1.3.3 Leave-One-Out

A special case of cross-validation, wehre each set $\mathcal{D}_i$ contains only one sample.

1.3.4 Bootstrapping

Suppose $\mathcal{D}$ has m samples. Randomly pick a sample from $\mathcal{D}$, copy it into some $\mathcal{D}'$ and put it back to $\mathcal{D}$. Repeat the process for m times.

$$\lim_{m\to\infty}(1-\frac{1}{m})^m = \frac{1}{e} \approx 0.368$$

About 36.8% samples in $\mathcal{D}$ will not be in $\mathcal{D}'$. So we can use $\mathcal{D}'$ for training and $\mathcal{D}\backslash\mathcal{D}'$ for testing.

1.4 Performance Evaluation

1.4.1 Measure

Regression Common performance measure for a regression model is **Mean Squared Error**.

$$E = \frac{1}{m}\sum_{i=1}^{m}(f(x^{(i)}) - y^{(i)})^2$$

Classification Common measure for a classification model is **Error Rate**

$$E = \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}[f(x^{(i)}) \neq y^{(i)}]$$

1.4.2 TPR and FPR

Definition 1.4.1 (Sensitivity/TPR)**.**

$$TPR = \frac{TP}{TP+FN}$$

Definition 1.4.2 (FPR)**.**

$$FPR = \frac{FP}{TN+FP}$$

1.4.3 Receiver Operating Characteristic

Many classification models output a real value and compare it to a certain threshold.
The **ROC Curve** uses FPR as its x-axis, and TPR as its y-axis. It can be plotted by setting different thresholds for dividing positive and negative samples.
The **Area Under Curve, AUC** is used to evaluate different models. Usually models with a larger AUC is considered to have better performance.

1.4.4 Precision and Recall

Definition 1.4.3 (Precision)**.**

$$P = \frac{TP}{TP + FP}$$

Definition 1.4.4 (Recall)**.**

$$R = \frac{TP}{TP + FN}$$

Similar to the ROC Curve, we can also plot the **P-R Curve**. And the **Break-Even Point, BEP**, defined as the value when $P = R$, is used to evaluate different models.
Another more common measure is the $F1$ rate

Definition 1.4.5 ($F1$ Rate)**.**

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\#Samples + TP - TN}$$

Remark. The $F1$ rate is defined by the harmonic mean of Precision and Recall.

Definition 1.4.6 (F_β Rate)**.**

$$F_\beta = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

Remark. F_β is the weighted harmonic mean. When $\beta > 1$, precision has a higher weight. When $0 < \beta < 1$, recall has a higher weight.

1.5 Error Analysis

Bias. The **bias** is the difference between model prediction and ground truth.

Variance. The **variance** is the variance of outputs of the same model fitted different times.

Noise. Noise.

High variance → Overfitting.

High bias → Underfitting.

1.5.1 Bias-Variance Decomposition

Let

$$bias(x) = f(x) - y$$

$$var(x) = \mathbb{E}_{\mathcal{D}}[(f(x; \mathcal{D} - f(x)))^2]$$

The generalization error of a model f trained on $\mathcal{D}$ can be represented by

$$E(f; \mathcal{D}) = bias^2(x) + var(x) + \varepsilon^2$$

Chapter 2

Regression

Almost all of them have closed-form solutions.

2.1 Linear Regression

2.1.1 Notations

- X: $N \times d$ matrix of data.
- $x^{(i)}$: i-th sample, d dimensional feature vector, suppose $x_0 = 1$.
- y: N dimensional output.
- w: $d \times 1$ parameter.

2.1.2 Model

$$f(X; w) = w^T X$$

Remark. If X is invertible, then we are done.

$$w = X^{-1} y$$

Unfortunately X is usually not invertible, it is usually not even a square matrix. So we need optimizaiton-based approach to solve this.

2.1.3 Loss Function

$$J(w) = \frac{1}{2}\|Xw - y\|_2^2 = \frac{1}{2}\sum_{i=1}^{N}(w^T x^{(i)} - y^{(i)})^2$$

2.1.4 Descent Direction

$J(w)$ is convex, and can be optimized by gradient descent.

$$\nabla J(w) = X^T(Xw - y)$$

$$w_{i+1} = w_i - \alpha \nabla J(w)$$

where α is the step size.

2.1.5 Closed Form Solution

$J(w)$ is convex, and we can calculate the closed-form solution.

Let $\nabla J(w) = 0$.

$$X^T X y = X^T y$$

$$w = (X^T X)^{-1} X^T y$$

Remark. If $X^T X$ is not invertible, we can use its pseudo-inverse, which is defined as

Definition 2.1.1 (Pseudo-Inverse)**.** `raise NotImplementedError('查完凸优化书再写')`